# CMMI the Agile Way in Constrained and Regulated Environments

## Paul E. McMahon, PEM Systems

**Abstract.** Today organizations are questioning if the CMMI is still relevant as they move to more popular agile approaches, such as Scrum. But many of these same organizations are also discovering that agile approaches alone are failing to provide the product quality their customers are demanding especially in constrained and regulated environments. This article employs a case study of an organization that recognized they had gone too far in abandoning their CMMI-based heavy-weight processes in favor of an agile approach. The article describes how the organization rapidly put critical lite-weight practices in place-- complementing their agile approach-- that measurably improved performance in just a few months by using a combination of three frameworks; CMMI, Scrum and Essence.

#### Background

Is the CMMI still relevant given today's new agile approaches? [1, 2, 3] When the agile manifesto [4] was signed close to fifteen years ago in 2001 many believed agile approaches to software development were just a passing fad that would not be around in 2016. However, today agile approaches are not only being employed by small teams, but are now a focus of improvement efforts in many large enterprises [5]. As further evidence of the staying power of agile one needs to look no further than recent changes in the way the US Department of Defense (DoD) is acquiring new weapon systems.

The DoD acquisition process is governed by Directive 5000.01, The Defense Acquisition System [6], and Instruction 5000.02 which was recently released in January, 2015 [7]. The DoD's Defense Acquisition System is not intended to be a rigid, one-size-fits-all process. As stated in Instruction 5000.02:

"The structure of a DOD acquisition program and the procedures used should be **tailored** as much as possible to the characteristics of the product being acquired, and to the totality of circumstances associated with the program including operational urgency and risk factors."

While the word **"tailored"** appears in the instruction, nowhere in this instruction will you see the word **"agile.**" This is because the DoD doesn't want to endorse or dictate any specific method or approach. However, the Defense Agile Acquisition Guidebook [8] states:

"Agile has emerged as the leading industry software development methodology, and has seen growing adoption across the DoD and other federal agencies. Agile practices enable the DoD to achieve reforms directed by Congress and DoD Acquisition Executives. DoD Instruction 5000.02 heavily emphasizes **tailoring** program structures and acquisition processes to the program characteristics. **Agile** development can achieve these objectives through:

- Focus on small, frequent capability releases
- Valuing working software over comprehensive documentation
- Responding rapidly to changes in operations, technology, and budgets
- Actively involving users throughout development to ensure high operational value"

#### Why "Being Agile" is Critical to Future Success, but isn't Easily Achieved

While you don't have to be agile to comply with instruction 5000.02, according the DoD's Agile Acquisition Guidebook being agile can make it easier. Nevertheless, if you google "hate agile" you will find hundreds of stories of failed agile efforts [9, 10, 11]. So while there is great motivation to become agile, it isn't as easy as it might sound, and at least part of the reason relates to the word **"tailoring."** While tailoring is encouraged by instruction 5000.02, there is little guidance in how to conduct tailoring, what acceptable tailoring looks like, and what tailoring pitfalls exist.

#### What Makes Agile Attractive to the DoD?

The world today is changing fast both due to new threats and new technology. The old idea that you could create requirements and hold them constant for years as you developed new weapon systems doesn't hold so well today. This old way of thinking could be referred to as "backward-looking", or always looking back at the fixed requirements as we try to develop new capability that keeps moving further and further away from what we actually need today [12].

What makes agile attractive to the DoD today is that the problems they are facing keep changing and the priorities keep changing. Agile practices are more continually forward-looking which are more conducive to the continual changing world around us. But at the same time while looking forward at continual changes, the challenge we face is how to get the speed of agile while not jeopardizing the assurances of traditional proven engineering practices and frameworks, such as the CMMI. [13] Now let's look at a recent case study that demonstrates how one organization handled this challenge.

#### **Background for the NORO Case Study**

NORO is a small organization (less than 100 people) with a DoD contractor heritage that began with an agile-vision in order to respond rapidly to changing customer needs. This vision was in part a reaction of the founders who sought to escape the bureaucratic-slow-moving world of their previous organization's heavyweight CMMI-based processes. However, in 2015 as NORO experienced growth they realized they had gone too far in dropping traditional processes and needed to add back a level of process discipline. I was engaged by NORO to help them in their improvement journey in October 2015.

#### First Step to Improvement at NORO

As I do with all new clients, we started by conducting a discussion with the leadership team to gain a baseline understanding of what was working well and what wasn't working well at NORO. My goal in helping any organization improve is to first understand where they are so we don't break what is already working well. The intent is to figure out where the greatest pain is, and rapidly put small improvements in place that can payback quickly. The next step is to keep working to adjust the process in short iterations making further small improvements a normal part of the client's way of working [14, 15].

In NORO's case I observed a pattern I had seen in other organizations that had attempted to "*become agile*" without fully understand what true agility required [16]. This caused the organization to adopt a reactionary interrupt-driven style of work that led to frequent defect-ridden releases and customer dissatisfaction.

As it turned out we were able to put an incremental improvement plan in place by first teaching the team the basics of Scrum [17], along with a number of "extended-Scrum" best practices to address key pain points. The team chose two-week sprints, and in just the first three sprints were able to measurably improve performance. As part of this effort we employed a combination of three frameworks; CMMI [14], Scrum and Essence [18, 19].

#### Why Use Three Frameworks?

The reason we chose three distinct frameworks is because each has strengths that NORO required. [15]

#### About the CMMI

The CMMI is a process improvement framework that can help process professionals identify gaps in their organizational processes, but it does not provide sufficient help in "how-to" fill the gaps.

#### About Scrum

Scrum is a project management framework. Its strength is its simplicity and wide appeal. It is easy to learn enough to get started quickly, but very difficult to master because it lacks "howto" specifics for many essential practices.

#### About Essence

Essence is a software engineering framework that is intended to be used by software practitioners on an endeavor to help them assess their current status, risks, and gaps, and help them decide what is most important to focus on next. Essence helps teams discover the "**how to**" specifics they need without dictating "how to" specific practices.

How we used each of these frameworks at NORO is described in the remainder of the article.

#### **Improvement Approach at NORO**

In the improvement effort kickoff meeting at NORO we discussed pain points, success criteria, organizational culture, policies, and constraints. During initial discussions it became clear that fundamental work management practices were a priority. After I gave the NORO leadership team a "7-minute Scrum chalk-talk", it was agreed we would start with a two-hour team training session including five basic Scrum practices [17] with some recommended tailoring to address NORO constraints followed by three two-week sprints where the tailored Scrum practices, along with a few Essence-based extended practices would be piloted. The agreed approach was one where the team would learn by using the new practices on a real project. I coached them through the sprints by taking on the role of their Scrum Master while also training an internal Scrum Master and Product Owner. The agreed approach included a planned release at the conclusion of the third sprint.

#### **Tailoring at NORO**

I have never seen two organizations implement Scrum the same way. And even when an organization tries to roll-out a common repeatable organizational "agile/Scrum" process, as soon as the individual teams start to implement retrospective improvements they immediately begin to diverge.

In NORO's case we also recognized the need for specific tailoring to the standard Scrum approach to help address pain points identified. Before we discuss the specific tailoring and extended practices we put in place at NORO, let's talk more about tailoring in general.

#### The Agile Way to Conduct Tailoring

Tailoring is a best practice encouraged by the CMMI framework. However, the way tailoring has been conducted in the past-- especially in many highly regulated/constrained environments-- represents the antithesis of effective agile/ lean practices.

This is because it is often conducted in a "tailoring-down" manner which means you start with a long list of products/practices and you identify the ones you don't need. The problem with this approach is first that it runs the risk of someone inexperienced deleting an essential product/practice.

The second problem is that it requires an oftentimes lengthy effort to explain why you don't need to do something. Lean/agile approaches are about eliminating waste, and one of the best ways to eliminate waste is by not requiring someone to explain why they don't need to do something that isn't essential.

A better approach is to start with a small list of **essentials** and then "tailor-up" thus eliminating inefficiencies and risk of tailoring out an essential. So where should you look for such an agreed-to-minimal-essential starting point?

#### A Common Ground Starting Point for Lean/Agile Tailoring-Up

An effort began in 2010-- referred to as Software Engineering Method and Theory (SEMAT) [20] -- to define such a minimal essential set, or common ground, for all software engineering endeavors. The result of that effort led to an Object Management Group (OMG) standard in 2014 referred to as Essence [18]. The Essence standard meets three important goals that any common ground in a high regulatory environment should have:

- Goal 1: Widely agreed upon
- Goal 2: Independent of any specific practices
- Goal 3: Extensible

What made the development of Essence challenging was that it had to be independent of any specific practices thereby supporting any approach to software development. The framework



### Software Engineering Kernel



is organized around a kernel containing seven essential things we work with, referred to as alphas, on all software engineering endeavors. Each alpha has a set of states and checklists. See Figure 1. For more information on Essence refer to [19].

What makes Essence particularly attractive to the challenge of high regulatory software endeavors, such as those faced by the DoD, is the fact that it does not limit, or dictate any specific software method.

#### **Rationale for Scrum Tailoring at NORO**

While some "purists" argue that you aren't using Scrum if you aren't following the Scrum rules precisely as prescribed by the official Scrum guide [17], the fact is many organizations must live with constraints beyond their control. As an example, at NORO I heard about contracts that required reported bugs to be fixed within 24 hours. Partly because of such customer constraints, NORO had a culture of being reactionary and interrupt-driven. When a key customer reported a problem, everyone dropped whatever they were doing to solve the problem immediately.

#### Using Essence to Help Team's Discover the "How-To" Specifics They Need

One way Essence can help teams discover the "how-to" specifics they need is by stimulating risk discussions leading to practical mitigation activities. As an example, during a pre-sprint planning session at NORO I conducted an independent risk assessment using an Essence-based risk practice. By "Essence-based risk practice" I mean a practice that has been developed using the Essence kernel. The Essence-based risk practice uses the seven essential alphas, along with their states and checklists to stimulate risk discussions. At NORO this activity led to the identification of the following three high risks and agreed to mitigation activities.

#### 1. Stakeholder Risk

NORO has many customers that use their core product. To address varying customer needs the product is configurable. A common problem NORO faces is changes made to address one customer's reported defects, too often cause unintended negative consequences in the way another customer uses the product.

The Essence Stakeholder alpha state "In Agreement" contains the following checklist:

#### The stakeholder representatives have agreed on their minimal expectations for the next deployment of the new system. See Figure 2.

While it is understood that multiple stakeholders often have competing needs and often reaching complete agreement is not realistic, this checklist highlights the importance of getting your key stakeholders to at least agree on their minimal expectations for the next deployment of the new system.

In NOROs case discussions around this checklist item led to actions accepted by the product owner to conduct meetings with two critical stakeholders most likely to be in disagreement based on past product releases. The goal was to get key representatives from each stakeholder organization to attend an internal sprint review to gain early feedback before the next formal release of the product. This risk discussion uncovered the fact that one critical stakeholder would not be able to attend the sprint review due to a conflict, and therefore a risk mitigation plan was put in place to deliver the early product version on site to this customer to proactively gain their early feedback prior to formal release.

#### 2. Software System Test Risk

When I was contracted to help NORO they were very open in telling me they knew they needed help with their testing approach. They did not have formal written test procedures, although they did have an independent test group, and they did have an independent quality control department that was required to approve each product release before shipment to any customer.

The Useable state of the Software System alpha defined by Essence contains the following checklist item:

#### Defect levels are acceptable to the stakeholders. See Figure 3

Initially we planned to defer improving test practices at NORO until a later sprint. However, because the reactionary interruptdriven culture was causing serious pain in the organization, I recommended that we raise the priority and start putting some small test improvements in place immediately on the very first sprint.

Our first improvement to address this pain point was to initiate a three-sprint cycle where every third sprint would be a formal release. Previously NORO did not have a well-defined product release process.

This release process didn't require all customers to necessarily install the new release every third sprint, but it did require the NORO team to work to that possibility. With the three-sprint cycle we instituted a plan where in the third sprint of each cycle only 50% capacity of developers would be used for new functionality, and the other 50% would be used for increased testing-- focusing specifically on regression testing. Previous to this recommendation NORO had no regression test suite.



SEMAT	4 / 6	A STANLAND	SEM	3 / 6	And a start	SEM	4 / 6	Chine Star
Figure 2			Figure 3			Figure 4		
Stakeholder Alpha State "In Agreement"			Software System Alpha State "Useable"			Way of Working Alpha state "In Place"		

Stakeholder Alpha State "In Agreement"

Software System Alpha State "Useable"

3. Way of Working Risk During the risk assessment session I asked the leaders at NORO what they were most worried about related to the improvement effort. Another way I often phrase this question is:

"What keeps you up at night?"

The candidate Scrum Master at NORO who I was training replied that his biggest concern was the way we were just training the software team, and that the rest of the organization would continue to operate as they always did. This meant they would be running to the software developers expecting them to drop everything to solve their current problem.

The "In Place" state of the Way of Working alpha defined by Essence contains the following checklist item:

### The practices and tools are being used by the whole team to perform their work. Refer to Figure 4.

When I heard this concern, I turned to the President of NORO and said,

"We need you to handle this risk. You need to let the whole organization know that the product owner is the one and only person who owns the product backlog, and if they want work done by the team it has to go on the backlog, and be prioritized."

He nodded, accepting the action, and in the very first sprint the resulting improved organizational performance was observed when a developer commented that an Operations Vice-President came to him with what normally would have been an emergency interrupt, and said:

"I know you are in the middle of a sprint, but I wanted to get this high priority issue onto your backlog so it can be addressed as soon as possible."

One of the developers mentioned during a daily standup meeting that "bugs often come back". This led the team to agree that they should capture the tests they run to fix bugs and run these tests again in an automated way rather than keep re-inventing them every sprint and running them manually which had become common practice. We started NORO on the path to improve their testing in the first sprint with a small improvement related to a developer's suggestion. This is an example of how NORO began to continuously improve their practices in small steps.

### Agile improvements start small, are continuous and are discovered and implemented by the team

I want to highlight how improvements began to happen at NORO. Most practical and useful improvements start out small and continue incrementally and are discovered and implemented by the team, rather than a separate "process group" as is often found in organizations that implement the CMMI in traditional non-agile ways.

Too often when I go into large organizations that are supposedly "CMMI mature" I find, for example, that they don't have regression test suites. When I challenge them to start improving in this area they say they want to do this, but they don't have the time or budget now-- so nothing happens!

# Continuous improvement isn't about increasing cost

The view in too many organizations is that change is always costly. This is a mistaken belief. This is not to say that change isn't hard. But if you change your perspective to recognize that the best way to change is in small steps every sprint<sup>1</sup> then you can start a new culture that says we can get better every day. When you empower your teams to improve their own processes on a regular basis, process improvement becomes a natural part of their normal way of working.

Key CMMI Process Areas and Generic Practices	Scrum Feature	Essence Feature	
Project Planning	Product Backlog, Sprint Planning	Early Alpha state checklists for all 7 Kernel Alphas	
Project Monitor and Control	Daily Stand-Up	Work Alpha	
Risk Management	Daily Stand-up	Common ground, and Essence-based Risk Practice	
Requirements Management, Requirements Development	Product Backlog	Requirements Alpha, Stakeholder Alpha	
Verification and Validations	Sprint Review	Software System Alpha checklists	
Supplier Agreement Management	Common Scrum base practices	Agreed to common ground	
Organizational Process Focus and Definition Generic Practice 3.1 Establish a Defined Process	Common Scrum base practices	Common ground supporting tailoring up approach	
Generic Practice 2.7 Ident & Involve Stakeholders	Product Owner role	Stakeholder Alpha	
Generic Practice 3.2, Continuous Improvement	Scrum Retrospective	Essence state checklists	

Table 1 Scrum and Essence Features Helping the CMMI

#### Subcontract Management at NORO

NORO, at times, uses a subcontractor that has specialized knowledge about a specific area of their core product. We had decided not to involve any subcontractors in the initial improvement effort so as not to add risk by taking on too much change at once. However, the day after we kicked off the first pilot sprint two subcontractor developers showed up on site to work a number of high priority customer reported defects.

NORO management was trying to figure out how they would monitor the subcontractors work when we realized how easy it would be pull them into the pilot effort. This was because the two developers had been trained in a similar process previously by myself on an improvement effort for a different client.

Often managing subcontractors can present large challenges due to variances in processes across organizations. But because we had an *"agreed-to set of essentials"*, the subcontract's work was easily integrated into NORO's pilot improvement effort.

#### Scrum and Essence Features Helping the CMMI

The CMMI is a process improvement framework that is still relevant today, but it doesn't provide everything an organization needs to continuously improve their way of working. Table 1 shows Scrum and Essence features that can help organizations effectively implement key CMMI Process Areas and Generic Practices [21].

#### Measurable Improvement at NORO

Survey's conducted with NORO leadership, development team members and NORO stakeholders confirmed measurable improved performance through reduced interrupts, improved reuse of tests, improved tracking of work tasks and increased team velocity and morale in the very first sprint. By the end of the second sprint we were able to validate the survey results with quantifiable data indicating a doubling of the workload being completed, or, in other words, a 100% improvement in team productivity.

#### Conclusion

The CMMI is a process improvement framework that can help process professionals identify gaps in their organizational processes, but it does not provide sufficient help in "how-to" fill the gaps. The NORO case study demonstrated that Scrum and Essence used together can help organizations implement key CMMI process areas and generic practices in an agile and effective way. Agile approaches, such as Scrum, are not just a passing fad, but they are insufficient to ensure software intensive products exhibit the quality customers are demanding today-- especially in constrained and regulated environments. Essence is a software engineering framework that teams can rapidly start using along with whatever practices or improvement frameworks they are currently using to assess current status, risks, root causes of problems, practice gaps, and put timely actions in place to continuously improve.<sup>2</sup> Essence helps teams discover the "how to" specifics they need without dictating "how to" specific practices. It can help teams transition from a static way of working to a more dynamic way that embraces continual improvement through continual small changes to the way they are working today.<sup>3</sup>

# <u>ABOUT THE AUTHOR</u>



Paul E. McMahon, Principal, PEM Systems (www.pemsystems.com) has been an independent consultant since 1997. He has published more than 45 articles and multiple books including "15 Fundamentals for Higher Performance in Software Development." Paul is a Certified Scrum Master and a Certified Lean Six Sigma Black Belt. His insights reflect 24 years of industry experience, and 17 years of consulting/coaching experience. Paul has been a leader in the SEMAT initiative since 2010. E-mail: pemcmahon@acm.org

NOTES

- 1. The focus of improvement when using Scrum is on each individual team. A strength of the CMMI is bringing attention to the need to propagate improvements across the organization.
- Essence state checklists help teams with continuous improvement by stimulating conversation related to where weaknesses exist leading to team-agreed improvements.
- I thank Bob Epps, Winifred Menezes, and Barry Myburgh for reviewing and providing improvement suggestions for this paper. The Essence figures are courtesy of SEMAT, Inc.

# REFERENCES

- 1. Is CMMI Still Relevant? http://vip-vatsa.blogspot.com/2012/08/is-cmmi-still-relevant.html
- CMMI Still Relevant? SEPG 2012. http://www.davidconsultinggroup.com/insights/ blog/posts/2012/march/05/cmmi-still-relevant-sepg-2012/
- 3. Is the CMMI Dead? http://www.cmmifaq.info/#200
- 4. Agile Manifesto, http://agilemanifesto.org/
- Disciplined Agile Delivery; Agility at Scale http://www.disciplinedagiledelivery. com/agility-at-scale/.
- Department of Defense Directive 5000.01, May 12, 2003, http://www.dtic.mil/whs/directives/corres/pdf/500001p.pdf
- Department of Defense Instruction 5000.02, http://www.acq.osd.mil/fo/docs/500002p.pdf
- Defense Agile Acquisition Guide, March, 2014, http://www.mitre.org/publications/technical-papers/defense-agile-acquisitionguide-tailoring-dod-it-acquisition-program
- 9. Why Do Managers Hate Agile?, http://www.forbes.com/sites/stevedenning/2015/01/26/why-do-managers-hate-agile/
- Seven Things I hate about Agile, http://blog.assembla.com/AssemblaBlog/tabid/12618/ bid/87899/Seven-Things-I-Hate-About-Agile.aspx
- 11. I can't take this agile crap any longer, https://news.ycombinator.com/item?id=5406384

- Miller, Suzy, Latham, Mary Ann et al, Parallel Worlds: Agile and Waterfall Differences and Similarities (CMU/SEI-2013-TN-021), October, 2013, http://resources. sei.cmu.edu/library/asset-view.cfm?AssetID=62901
- 13. Myburgh, Barry, Effective Governance Enables Success, Chapter 11, Software Engineering in the Systems Context, College Publications, 2015
- 14. McMahon, Paul E., Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement, Addison-Wesley, 2011
- McMahon, Paul E., 15 Fundamentals for Higher Performance in Software Development, Appendix D, PEM Systems, 2014
- McMahon, Paul E., Hybrid-Agile Software Development: Anti-Patterns, Risks, and Recommendations, Crosstalk, The Journal of Defense Software Engineering, July/August 2015
- Scrum Guide, The Definitive Rules of Scrum, http://www.scrumguides.org/docs/ scrumguide/v1/scrum-guide-us.pdf
- 18. OMG Essence Specification, http://www.omg.org/spec/Essence/Current
- McMahon, Paul E., "A Thinking Framework to Power Software Development Team Performance, Crosstalk, The Journal of Defense Software Engineering, Jan/Feb 2015
- 20. www.semat.org
- 21. CMMI for Development, V1.3, http://www.sei.cmu.edu/reports/10tr033.pdf

# **WE ARE HIRING**

# **ELECTRICAL ENGINEERS AND COMPUTER SCIENTISTS** As the largest engineering organization on Tinker Air Force Base, the 76<sup>th</sup> Software Maintenance Group provides software, hardware, and engineering support solutions on a variety of Air Force platforms and weapon systems. Join our growing team of engineers and scientists!

# **BENEFITS INCLUDE:**

- Job security
- Potential for career growth
- Paid leave including federal holidays
- Competitive health care plans
- Matching retirement fund (401K)
- Life insurance plans
- Tuition assistance
- Paid time for fitness activities

Tinker AFB is only 15 minutes away from downtown OKC, home of the OKC Thunder, and a wide array of dining, shopping, historical, and cultural attractions.



Send resumes to: 76SMXG.Tinker.Careers@us.af.mil

US citizenship required

Oklahoma City SkyDance Bridge, Photo © Will Hider