

## CMMI vs. Scrum? No — CMMI + Scrum!

by Jeff Dalton

"Are CMMI and Agile compatible?" It's a great question, and one that I ask myself frequently. For five years, I've dutifully answered the question with a resounding "Yes!" and a full explanation of how it might work. However, instead of exploring that question yet again, I suggest that we are focusing on the wrong question altogether. I would posit that the right question is "CMMI and Agile together: why aren't we having this conversation?" In a community whose most vexing problem is how to make Agile work in a business that is decidedly not Agile, it would seem that characteristics inherent in the architecture of CMMI — robustness, scalability, predictability, and structure — would be useful in scaling Agile to the enterprise. It's not that CMMI and agile don't work well together; of course they do. Given the fact the agile exists to improve the products we build, and CMMI exists to improve how that work is accomplished, why are we not using them together? Why not embrace CMMI to make agile better?



The article in this reprint originally appeared in *Cutter IT Journal*, Vol. 25, No. 11. Below is the Table of Contents from that issue. To purchase the complete issue, shop our bookstore at [www.cutter.com](http://www.cutter.com) or call +1 781 648 8700 or send e-mail to [sales@cutter.com](mailto:sales@cutter.com).

## Agile CMMI: Why Isn't This Conversation Dead Yet?

### Opening Statement

by Hillel Glazer

### The Agile CMMI Conversation Is a Dead End

by Bill Fox

### Blending Agile and CMMI

by Brian Button and Nate McKie

### Agile CMMI: The Real Underlying Obstacles to Effective Integration and What You Can Do About Them

by Paul E. McMahon

### CMMI vs. Scrum? No — CMMI + Scrum!

by Jeff Dalton

### Disciplined Agile Delivery Meets CMMI

by Scott W. Ambler

### What Will It Take to Achieve Agility-at-Scale?

by Douglas Schmidt, Anita Carleton, Erin Harper, Mary Ann Lapham, Ipek Ozkaya, and Linda Parker Gates



# CMMI vs. Scrum: No — CMMI + Scrum!

by Jeff Dalton

When searching for inspiration, I turn to Charlie Parker.

Charlie “Bird” Parker was a brilliant, though seriously flawed, pioneer of the American jazz scene. While his personal challenges are well documented, Parker is known as the undisputed master of the idiom, who understood that being great was not simply about being talented. His legacy, regarded by music historians as one of the most powerful in the history of American music, was not an accident. It was arrived at through years of disciplined study, practice, and experimentation, which resulted in the very definition of the art form. But it was not simply about study, nor was it only about a disciplined adherence to structure. Parker knew what so many in software engineering still struggle to understand today: that great accomplishments are achieved through mastering and synthesizing all three elements — talent, learning, and discipline.

What really set Bird apart was his ability to master these concepts with extraordinary agility. When he performed, he heard something very different from the music of his predecessors. While standing firmly on the shoulders of the giants who came before him, he created a new, agile style that catapulted jazz into an entirely new dimension. With its short bursts of creativity, rapid real-time adaptations, and incremental, iterative improvisational character, this style — which came to be known as “be-bop” — would be better described as real-time composition.

Unlike the music that preceded Parker’s 1939 debut, his was incremental and iterative in three dimensions. The

first was internal to the skills of any accomplished musician, who learns to hear the sound in the split-second it takes for it to escape his or her instrument, and then incrementally *inspects and adapts* the tone, inflection, and pitch — sometimes before the sound wave has even reached the audience. The second dimension was the *real-time collaboration* among and between the members of his group: between saxophone and piano, between drums and bass, between piano and guitar, and a *continuous build* of those collaborations across the ensemble. The magic of be-bop is in the real-time composition created when a group of accomplished players collaborate as a team, *fail fast*, and deliver the *minimum viable product* throughout the course of the composition. Finally, Bird would *collaborate* with his audience — reading their reaction, inspecting and adapting, and recalibrating his compositions to meet the desires of his fans.

If it sounds as though I’m saying that agile methods have been around a lot longer than Scrum, XP, and the spiral model, I am. While I have immense respect for the authors of the Agile Manifesto,<sup>1</sup> they were 60 years behind Bird.

The lessons from Charlie Parker are as relevant to agile teams today as they were to musicians in the 1940s — most software organizations still struggle with the synthesis of talent, learning, and discipline. It’s not for lack of trying. The landscape is littered with models, techniques, and tools in search of software’s perfect chord, yet we continue to struggle with the processes required

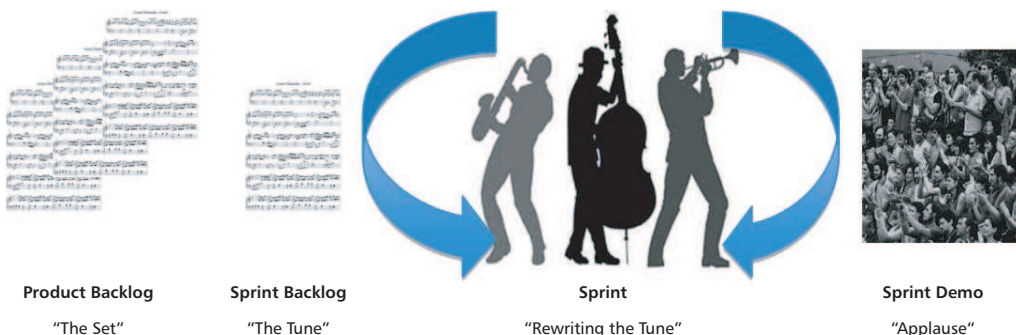


Figure 1 — The first agile teams.

to improve productivity and increase the predictability and stability of software projects.

The theme of this issue is “Agile CMMI: Why Isn’t This Conversation Dead Yet?” It’s a great question, and one that I ask myself frequently when, for the 20th time each week, someone asks on my blog,<sup>2</sup> “Are CMMI and agile compatible?” For five years, I’ve dutifully answered the question with a resounding “Yes!” and a full explanation of how it might work. However, instead of exploring that question yet again, I suggest that we are focusing on the wrong question altogether.

**CMMI was never intended as a process to be followed, but as framework to improve the methods we are following.**

I would posit that the right question is “CMMI and agile together: why aren’t we having *this* conversation?” In a community whose most vexing problem is how to make agile work in a business that is decidedly *not* agile, it would seem that characteristics inherent in the architecture of CMMI — robustness, scalability, predictability, and structure — would be useful in scaling agile to the enterprise.

It’s not that CMMI and agile don’t work well together; of course they do. Given the fact the agile exists to improve the products we build, and CMMI exists to improve how that work is accomplished, why are we *not* using them together? Why not embrace CMMI to make agile better?

## WHY NOT EMBRACE BOTH! A HISTORY

Agile *puristas* scoff at the idea. CMMI process teams shake their heads, struggling to understand how they can simultaneously “get a level” and be agile. In our Software Engineering Institute (SEI) Technical Note “CMMI or Agile: Why Not Embrace Both!”<sup>3</sup> my coauthors and I argue that there is no real conflict between these communities — only a complex misunderstanding based on an unfortunate confluence of events. Indeed, there are not two communities; there is just one, whose only desire is to bring excellence to software engineering. Think about it. Both CMMI and agile methods exist for the same reason: to help us build better products.

In the 1980s, before agile was “agile,” and almost 15 years before CMMI reached maturity, the SEI sprang forth as a federally funded R&D center, forging a partnership between government and academia whose focus was on the research and development of new ways to improve the state of software engineering. While the SEI has done a remarkable job expanding its scope over the years to include other models and technologies — most notably CMMI for Services (CMMI-SVC), People-CMM, and CMMI for Acquisition (CMMI-ACQ) — it is the Capability Maturity Model Integration for Development (CMMI-DEV) that has been its crown jewel and most successful product for many years.

When considering the perspective of the research scientists at the SEI during that period and the methods that were dominating the software industry at the time, one might forgive them for not promoting agile methods, such as Scrum, which had not yet been established. That the CMMI hasn’t done so since is not the result of an oversight or lack of understanding, as its critics often suggest. Rather, CMMI is a method-agnostic model that is more akin to a *behavioral improvement model* than an *engineering process improvement model*. It was never intended as a process to be followed, but as framework to improve the methods we *are* following. And to those who continue to believe that CMMI is about “filling out forms” and “getting a level,” I would ask, how do those things improve the performance of the methods software teams are using?

The language embraced within the CMM for Software (SW-CMM), and later CMMI, was based on the prevailing language of the day, such as “Work Breakdown Structure,” “Configuration Status Accounting,” “Project Audits,” “Technical Data Package,” and others that are still present in the latest version of CMMI-DEV (v1.3). For agile evangelists, this language appears to clash with agile values and has led to the persistent questions about CMMI’s compatibility with agile. These questions about compatibility are rooted in the perception, voiced by many in our industry, that CMMI was intended for use in traditional waterfall-style project environments.

Members of the agile community routinely use the phrases “top-down,” “command and control,” and “low-trust environment” to describe both CMMI and waterfall-style methods, but associations between those phrases and CMMI are incorrectly applied. In order to understand how software professionals came to see CMMI as analogous to waterfall, and seemingly at odds with the values of the Agile Manifesto, we must consider the time frame and the predominant methodologies that existed

when SW-CMM, and subsequently CMMI, were being developed. While researching and compiling their work, the staff at the SEI — who, like agile teams, are pretty good at collaboration themselves — enlisted the expertise and input of a broad cross-section of organizations that were demonstrating success in software engineering. Many of these organizations were running successful, large-scale programs using what we now call “traditional” or “waterfall” lifecycle models, and many of these organizations became early adopters.

As CMMI became more popular, so did the case studies of the early adopters, thus setting a standard for CMMI implementation that mirrored their successes. The US federal government’s mandate (now expired) that its software suppliers achieve a level of CMMI drove later adopters to emulate their predecessors, and thus a pattern was established for everyone to follow.

Today, a growing number of agile software teams are challenging the conventional wisdom that continuous process improvement requires substantial investments in overhead and documentation. Instead, they favor real-time collaboration, information radiators, colocated team rooms, and increased transparency within the team. These agile values are not in conflict with the intent of CMMI. They may be in conflict with the values of the early adopters, but their use of the model was only a single instantiation from an unlimited set of possibilities. Those early users were simply applying CMMI in the way it was intended — to make what they were already doing better!

Agile methods, like more traditional waterfall methods, are a collection of values and techniques that are applied in order to accomplish a task. Instantiations of agile such as Scrum encapsulate those values and techniques into a

system that delivers value to customers in an iterative, collaborative, and transparent way. CMMI is neither a set of values, nor a set of techniques. It’s a set of guidelines intended to make any software development approach — whether it be agile or traditional — more efficient, effective, and predictable.

## CMMI CAN IMPROVE AGILE

As I often say to my students, CMMI is many things to many people.

Some see it as a way to measure organizational performance. Others see a requirement for bidding on federal contracts that only adds overhead. Some others see it as a giant checklist of “must have” process steps, while still others see a set of guidelines for continuous improvement.

CMMI-DEV is organized into a hierarchy of 22 process areas (PAs), each with multiple goals, practices, and subpractices. While these process areas contain clusters of related practices, they are not processes, and it is necessary to trace a path between multiple PAs and practices in order to “follow a thread” as one establishes a localized model that can be applied to an organization’s instantiation of agile methods.

Traditionally, and all too often, companies take a “CMMI-centric” approach to process development that is model-focused rather than value-focused. This results in a process with excessive “process debt” that does not reflect the values of the organization.

Consider the process flow in Figure 2, which takes a model-centric approach to defining part of a planning

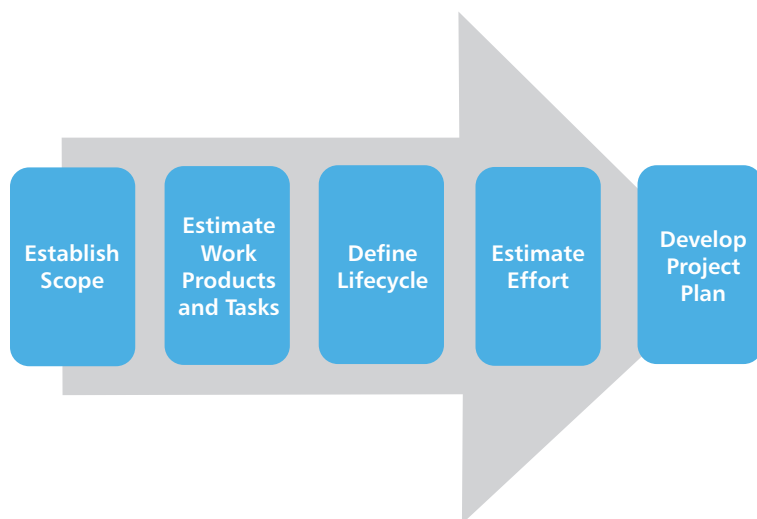


Figure 2 — Linear, model-centric processes don’t reflect agile values.

process. We often see something like this when a process is designed to “comply” with CMMI.

I prefer to interpret CMMI as a set of questions about the environment in which we choose to work. The answers to those questions will determine the experience that practitioners will have, and they will also affect the quality of a software team’s products and the predictability of project delivery.

**How can we strengthen Scrum so that it can scale? The solution lies not in applying CMMI directly to Scrum, but in asking the “CMMI questions” of an organization’s Scrum ceremonies.**

The questions must be asked within the context of team norms and the methods and techniques that the team has agreed to adopt. Interestingly, this is precisely how we should conduct CMMI-based SCAMPI Appraisals for *any* type of software development organization.

A value-focused approach for an agile project might seek to answer the following questions, derived from CMMI, about planning for releases and sprints:

- How are we projecting our ability to deliver value?
- How long will our sprints be? How many sprints are in a release?
- Is our team clear on how we’ll interact with project stakeholders?
- Do we know what we need to produce during the sprint?

- Have we broken down our stories into tasks?
- How will we know how our sprint is progressing?
- What keeps us up at night? Where do these risks originate?

Figure 3 shows the process flow from Figure 2 reimagined as a value-focused process using questions derived from CMMI but translated to the local team’s language.

A value-focused approach is more concerned with the natural flow of the work that is to be accomplished and less concerned with adherence to a linear, step-by-step process, letting a task management framework such as Scrum drive the sequence. In this way, the “process assets” evolve into a set of *process objects* that have encapsulated within them a robust set of characteristics that enable them to be adopted by the enterprise and improved over time. These process objects are instantiated as needed, based on the goals and objectives of the organization and the desire to improve on the highest-priority aspects of agile performance.

The practices that CMMI makes available to us include those that help bring greater clarity and strength to the Scrum ceremonies themselves (the “specific practices,” or SPs), and those that help strengthen the understanding, adoption, and continuous improvement of the agile values and behaviors (the “generic practices,” or GPs). Scrum ceremonies are loosely defined, with implementation left to self-organizing teams. While teams that are performing Scrum with integrity would be loath to accept rigid process oversight, the fact remains that successful implementation across the enterprise has proven elusive to many companies, especially where organizations have attempted to scale agile by applying a

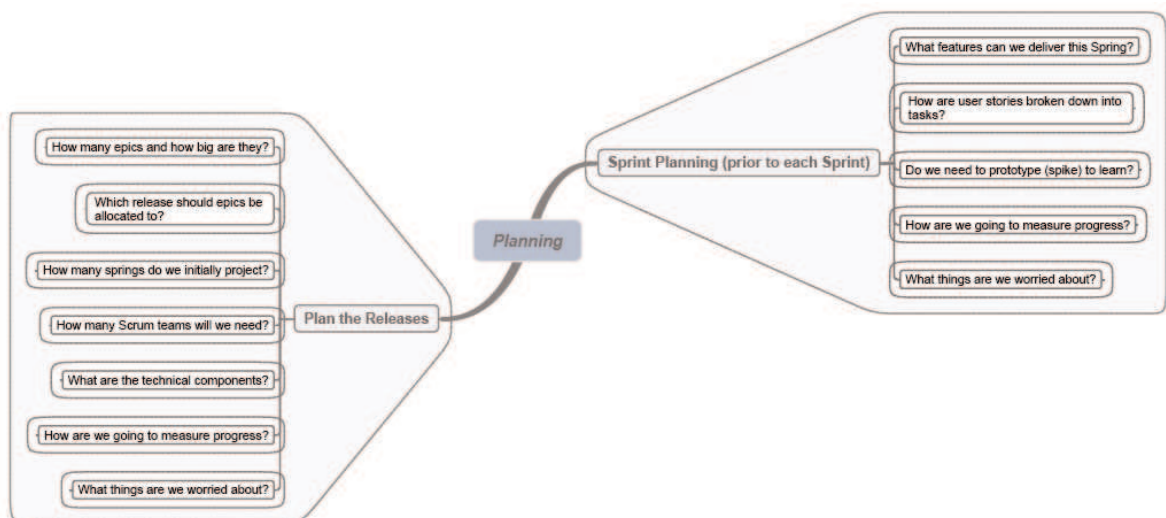


Figure 3 — Agile planning is iterative and incremental.

“Scrum of Scrums” framework. How can we strengthen Scrum so that it can scale?

The solution lies not in applying CMMI directly to Scrum, but in asking the “CMMI questions” of an organization’s Scrum ceremonies. For example, the following questions are derived from CMMI:

## Release Planning

- How many epics should be in a release?
- How are epics allocated to each release?
- How often are releases scheduled?
- How are releases organized into a definable lifecycle that we can communicate to our business customer?
- How many sprints are contained within a release?
- How many story points are projected to be in each release?

## Sprint Planning

- How long are our sprints?
- What criteria determine the length of sprints?

- How are stories allocated to each sprint?
- What criteria are used to determine allocation?

## Sprints

- What design artifacts are useful to our teams?
- How are effective code reviews conducted?
- What criteria are included in our definition of “done”?

## Retrospectives

- Which stakeholders should participate?
- How does the information gathered at retrospectives help the rest of the organization?
- What categories do we use to brainstorm improvements?

The mind map in Figure 4 ties these ceremonies, and others, to CMMI process areas, and the questions are derived from CMMI’s SPs.

For a more detailed example, let’s examine the retrospective. In Scrum, the retrospective is intended as a method for the team to learn lessons, in near real time,

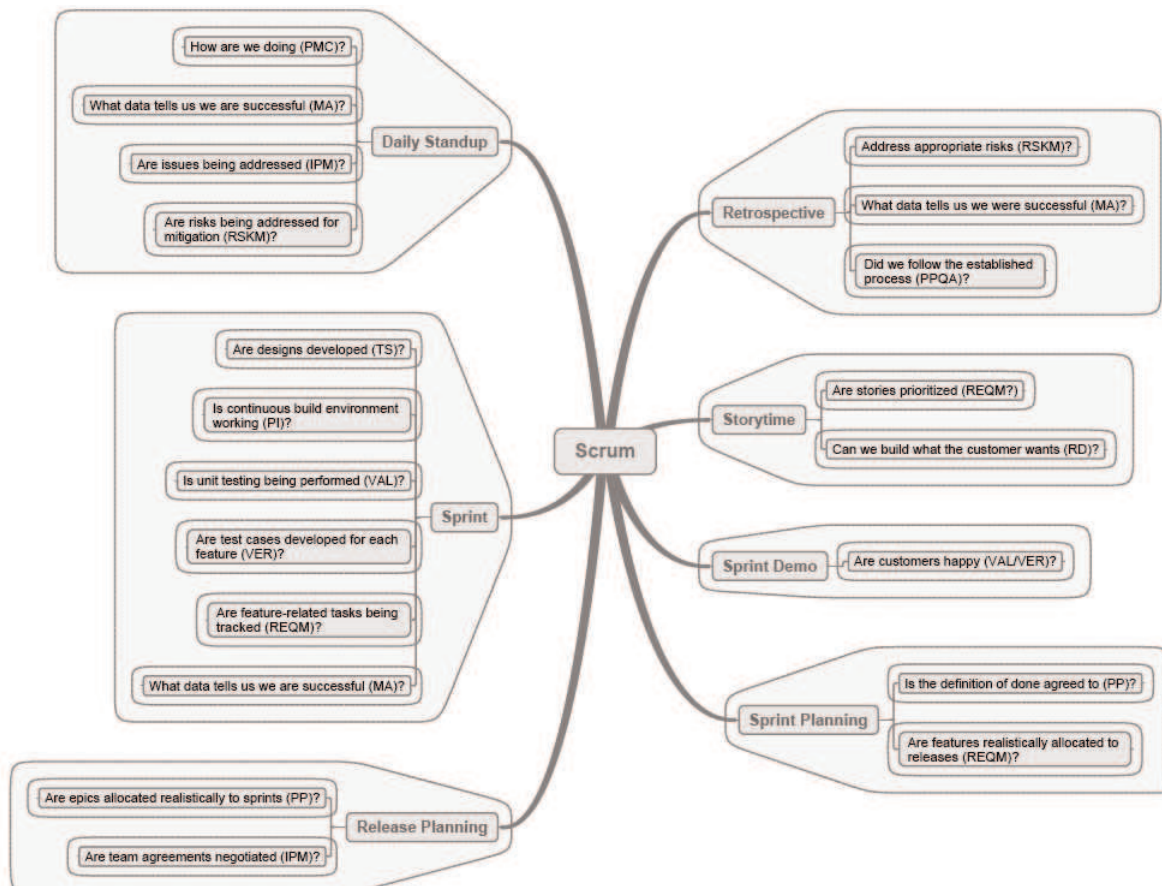


Figure 4 — Scrum ceremonies and events can be improved through questions derived from CMMI.

from the most recently executed sprint. Scrum itself gives little guidance, but many teams discuss the success or failure of the sprint, how closely they met their projected velocity, how well they collaborated together, feedback from the sprint demo, and other sprint-related experiences. With a self-organized team, the retrospective can be very effective in improving performance. But the rest of the organization learns little from this exercise, and changes based on the experience of the Scrum team often do not translate into benefits for other teams.

CMMI provides guidelines for the conduct of retrospectives in GP 3.2: “Collect Process Related Experiences.” The guidelines exist in every process area and can be applied to all Scrum ceremonies and activities being performed by Scrum teams.

By using CMMI to drive structured brainstorming, retrospectives become more powerful and provide increased value to the rest of the teams in the organization. Using the mind map in Figure 4, or better yet, a Scrum team’s own interpretation, a team can categorize retrospectives into logical buckets and use questions from the applicable PA to enrich their learning. For example:

- **Technical lessons** — practices from Technical Solutions and Product Integration to drive discussions
- **Sprint lessons** — practices from Project Planning, Project Monitoring and Controlling, and Requirements Management
- **Testing lessons** — practices from Validation and Verification
- **Scrum performance lessons** — CMMI’s generic practices

Teams may want to take care not to overload their retrospectives. They might consider rotating topics from one sprint to the next, or from process area to process area, and then share the information outside of the team so that others can learn and benefit from the experience. Collaboration exists beyond the Scrum team, and it’s the key to enterprise-wide performance improvement.

## MASTERY AND SYNTHESIS OF TALENT, LEARNING, AND DISCIPLINE

For more than a quarter of a century, software organizations have launched one attempt after another to establish a useful framework for effective continuous improvement. Both CMMI and agile methods have had a positive effect on the software industry, but neither

approach has yet to succeed in driving the industry-wide levels of adoption we need in order to claim victory. Why not combine the best of each?

The entire enterprise can leverage CMMI to make agile methods like Scrum better and more powerful. Instead of trying to achieve “a level of CMMI,” embrace CMMI to improve what is already being done within the Scrum team, to scale Scrum to the enterprise, and to expand the scope and influence of agile methods from the team room to the boardroom.

Seventy years ago, Bird proved that the integration of agility with talent, learning, and discipline can break down the barriers to greatness, and in doing so he created a revolution in music that lives on to this day. We can do the same with agile.

## ENDNOTES

<sup>1</sup>Agile Manifesto (<http://www.agilemanifesto.org>).

<sup>2</sup>*Ask the CMMI Appraiser* (<http://asktheCMMIAppraiser.com>)

<sup>3</sup>Glazer, Hillel, Jeff Dalton, David Anderson, Michael Konrad, and Sandra Shrum. “CMMI or Agile: Why Not Embrace Both!” SEI/Carnegie Mellon University, November 2008.

## ADDITIONAL READING

Sims, Chris, and Hillary Louise Johnson. *The Elements of Scrum*. Dymaxicon, 2011.

Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. *CMMI for Development: Guidelines for Process Integration and Product Improvement*. 3rd edition. Addison-Wesley Professional, 2011. *Agile CMMI* ([www.agileCMMI.com](http://www.agileCMMI.com)).

Jeff Dalton is President of Broadsword Solutions Corporation, a process innovation firm based in southeastern Michigan. Mr. Dalton is an SEI Certified SCAMPI Lead Appraiser, CMMI Instructor, ScrumMaster, and author of *agileCMMI*, Broadsword’s methodology for incremental and iterative process improvement designed to eliminate the process debt that demoralizes engineers and destroys any potential productivity gains. He is Chairman of the SEI’s Partner Advisory Board and President of the Great Lakes Software Process Improvement Network. Mr. Dalton is author of the popular blog *Ask the CMMI Appraiser* and builds experimental aircraft in his spare time. He can be reached at [appraiser@broadswordolutions.com](mailto:appraiser@broadswordolutions.com).

# Agile CMMI: Why Isn't This Conversation Dead Yet?

## FOR MORE INFORMATION/ TO ORDER:

Go to our secure bookstore at [bookstore.cutter.com](http://bookstore.cutter.com) for more information and/or to order this report. Your report will be delivered immediately in PDF.

You can also order by:  
Email: [service@cutter.com](mailto:service@cutter.com);  
Tel: +1 781 648 8700

Published: November 2012,  
40 pages, PDF format

## SUMMARY

It's not agile or CMMI that fosters your culture, and it's not agile or CMMI that shuns or requires documentation. In fact, as we'll see in this issue of *Cutter IT Journal* with Guest Editor Hillel Glazer and his host of six expert authors, it's really not about agile or CMMI. And perhaps "To agile or not to agile?" is the wrong question to ask and the wrong perspective on matters.

This issue addresses everything from the right questions to ask to how to deal in an agile manner with very large systems. It seems a fairly universal conclusion that it's not agile or CMMI that are incompatible but the way they're applied in a given situation that makes them incompatible.

## Table of Contents:

- **The Agile CMMI Conversation Is a Dead End** by Bill Fox. Get key recommendations on where to refocus your efforts to speed delivery and improve quality.
- **Blending Agile and CMMI** by Brian Button and Nate McKie. Explore how to apply the agile principle of learning to the adoption of CMMI.
- **Agile CMMI: The Real Underlying Obstacles to Effective Integration and What You Can Do About Them** by Paul E. McMahon. Understand "the real underlying obstacles" to agile CMMI and how to overcome

them and get innovative ways to interpret and apply CMMI so that its practices actually make sense in agile settings.

- **CMMI vs. Scrum? No — CMMI + Scrum!** by Jeff Dalton. Discover a simple and robust architecture that can be used to build processes incrementally and iteratively.
- **Disciplined Agile Delivery Meets CMMI** by Scott W. Ambler. Learn how Disciplined Agile Delivery (DAD) supports both agile and CMMI by providing a decision framework for determining the best processes to use in your organization.
- **What Will It Take to Achieve Agility-at-Scale?** by Douglas Schmidt, Anita Carleton, Erin Harper, Mary Ann Lapham, Ipek Ozkaya, and Linda Parker Gates. Learn how risk, measures, technical debt, and strategy come into play when dealing with Agile/CMMI issues.

Order your copy of *Agile CMMI: Why Isn't This Conversation Dead Yet?* today and **SAVE 50%!** Just use Coupon Code **AGILECMMI50** when you visit [bookstore.cutter.com](http://bookstore.cutter.com). You'll receive your copy immediately in PDF.

## To Order/For More Info:

Go to our secure bookstore at [bookstore.cutter.com](http://bookstore.cutter.com).

Go to [bookstore.cutter.com](http://bookstore.cutter.com)

### Get global perspectives and solutions to some of the most critical business technology issues facing organizations today!

#### Your Front-Row Seat to IT Management Debate at the Highest Level!

Every day, your organization is confronted with the stark reality of having to achieve more aggressive goals with a shrinking budget, ever-changing requirements, and impossible deadlines.

Few of you have the time to develop well-supported arguments on how to get your organization to improve its IT operations. It's a tough trap: you know solutions are out there, but you're too busy to identify them and convince your organization to implement them.

#### Advice, Solutions, and Experience That You Can Rely On

A **Cutter IT Journal** subscription helps you break out of the trap. Every month, **Cutter IT Journal** features a select Guest Editor who articulates the controversial issues, offers his or her opinion on them, invites others to introduce opposing viewpoints, and sparks a lively debate.

**Cutter IT Journal** provides you the opportunity to experience a variety of perspectives: viewpoints that will be instrumental in advancing the cause of better software development. No matter where you stand on these issues, the thoughtful discourse delivered in **Cutter IT Journal** will certainly help you clarify your position.

In addition to your monthly journal issue, you will also receive its companion **Cutter IT Advisor**. Each **Advisor** brings you practical advice and thoughtful analysis from well-known and respected experts in the IT field. Learn from the experiences of others, including what you should avoid and what you should consider implementing.

As a subscriber to **Cutter IT Journal**, you'll stay up to date on important IT issues such as project management, security, risk management, business intelligence, sourcing, enterprise architecture, requirements, trends in technology, and more. Whatever the topic, you can be sure you'll receive frank, unbiased opinions, in the no-holds-barred manner **Cutter IT Journal** is known for.

#### Don't miss upcoming issues on:

- Disciplined Agile Delivery Part II
- Value of Social Media Data Part II
- Agile Data Warehousing and Analytics

**Special Offer!** Begin your subscription to **Cutter IT Journal** today and save \$100 off the regular subscription rate! Order online in our secure bookstore at [bookstore.cutter.com](http://bookstore.cutter.com), using Priority Code CITJSAVE100 to receive your discount. Or complete and return the form below by fax +1 781 648 8707 or phone +1 781 648 8700.

For more information on **Cutter IT Journal**, please visit [www.cutter.com/itjournal.html](http://www.cutter.com/itjournal.html).

### Special Offer: Save \$100 on a New Subscription!

- ☐ **YES!** Please start my new, one-year subscription to **Cutter IT Journal** for just \$385 (US \$485 outside North America) — I save \$100 off the regular price of \$485/US \$585!

**CUTTER**  
CONSORTIUM

Name	Title
Company	Dept.
Address/P.O. Box	Mailstop/Suite
City	State/Province
ZIP/Postal Code	Country
Phone	
Fax	
Email	

Order online at [bookstore.cutter.com](http://bookstore.cutter.com) (enter Priority Code CITJSAVE100 to save \$100). Or fax form to +1 781 648 8707, call +1 781 648 8700, or send email to [service@cutter.com](mailto:service@cutter.com). Mail to Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA.

## About Cutter Consortium

Cutter Consortium is a unique IT advisory firm, comprising a group of more than 150 internationally recognized experts who have come together to offer content, consulting and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, business intelligence, metrics, and sourcing.

Cutter delivers what no other IT research firm can: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts; experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats including print and online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or [sales@cutter.com](mailto:sales@cutter.com).

## The Cutter Business Technology Council

The Cutter Business Technology Council was established by Cutter Consortium to help spot emerging trends in IT, digital technology, and the marketplace. Its members are IT specialists whose ideas have become important building blocks of today's wide-band, digitally connected, global economy. This brain trust includes:

- Rob Austin
- Ron Blitstein
- Tom DeMarco
- Lynne Ellyn
- Israel Gat
- Vince Kellen
- Tim Lister
- Lou Mazzucchelli
- Ken Orr
- Robert D. Scott