The Disciplined Agile Framework: A Pragmatic Approach to Agile Maturity

Scott W. Ambler, Scott Ambler + Associates
Mark Lines, Scott Ambler + Associates

Abstract. It is possible to combine Agile and Capability Maturity Model Integrated (CMMI), but few organizations are doing this in practice. There are many challenges to be overcome, not the least of which is the very different mindsets of adherents of each approach. More importantly, we have a better option available to us in the form of the Disciplined Agile (DA) Framework. The DA process decision framework provides light-weight guidance to help organizations streamline their software processes in a context-sensitive manner. It does this by showing how various capabilities such as solution delivery, operations, enterprise architecture, portfolio management, and many others work together in a cohesive whole. The framework also provides a range of options for addressing these capabilities, describing the tradeoffs associated with each option. The DA Framework provides organizations with an easier and better described path to agile capability maturity than the strategy of force-fitting Agile and CMMI together.

The Capability Maturity Model Integrated (CMMI) model purports to provide guidance to help organizations improve their software processes. This has in fact happened in some organizations, albeit in manners that are not as streamlined as many people would like. Agile strategies such as agile development techniques, agile architecture techniques, agile management techniques, agile modeling techniques, and more offer the potential to streamline and improve your software processes. This has occurred in practice in a multitude of environments. Agile has become the standard approach in many organizations and continues to grow in the majority of companies around the world. It is possible to combine Agile strategies and CMMI together: CMMI describes what should be done and Agile describes how it should be done, so combining them should be a good idea. That’s the theory – Practice shows there is a better way.

Rethinking Agile Capability Maturity

Let’s begin with several important observations:

1. Agile and CMMI promote different paths to capability. In CMMI [1], capability is defined as “the inherent ability of a process to produce the planned results. As the capability of a process improves, the process becomes predictable and measurable, the quality of the result product augments and the productivity of the teams too.” The Agile Manifesto [2], on the other hand, is very clear that the path to Agile capability is to build teams of motivated people and provide them with an environment where they can succeed. These teams should reflect on a regular basis so that they can identify, and then adopt, potential improvements to the way that they work. Agile teams own their own process and evolve it over time.

2. Agile seems to help CMMI. Tadros [3] found that there were fewer gaps found by SCAMPI reviews in organizations that had taken an agile approach compared with non-agile. The study also found that quality metrics were noticeably better in the organizations combining agile with CMMI strategies.

3. CMMI is attractive to government organizations and those that serve them. CMMI has “succeeded” in US government agencies where competition is non-existent and in companies focused on working for government organizations where questionable procurement practices virtually assure an onerous, documentation-heavy process. This is a harsh observation, but CMMI tends to thrive when productivity isn’t of paramount importance.

4. CMMI enables offshoring. CMMI is common with Indian service providers (ISPs) who use CMMI as a marketing strategy to lure customers, as well as Chinese service providers hoping to compete against ISPs. CMMI is also adopted by groups within private-sector companies that hope CMMI will help them to manage the work that they offshore to CMMI-compliant service providers.

5. Few agilists are interested in CMMI. Although McMahon [4] argues that CMMI can be applied to enhance agile techniques the reality is that agilists consider CMMI to be an anathema. Without a coach of McMahon’s caliber it is highly unlikely that you’d succeed at inflicting CMMI on an existing agile team – most likely your agile developers would find ways around that mandate or choose to seek employment elsewhere.

6. Leading-edge software engineering organizations rarely consider CMMI. Organizations in highly competitive environments such as Apple, Google, and Etsy to name a few would laugh you out of the room were you to suggest that they adopt CMMI. Time to market and quality are paramount for these organizations, so that forces them to be incredibly effective at software development. If CMMI truly offered the potential for them to be more effective they would adopt it as swiftly as possible, yet that isn’t happening.
7. **CMMI is shrinking in the private sector.** Most large financial institutions, retailers, telecommunications companies, and automotive companies have tried to adopt CMMI at some point. Yet it is rare to find one that applies CMMI-based strategies on a regular basis, and if they're still doing CMMI at all it is on a handful of legacy teams.

The real question isn't whether we can apply CMMI and Agile together, but instead does CMMI realistically bring anything of value to the table for Agile teams? Although McMahon [4] makes very good arguments for how Agile can help CMMI, his arguments for the reverse seem a bit stretched at times. Yes, CMMI does provide guidance for what software organizations need to do, but our experience is that CMMI guidance is both incomplete and often inflexible in practice. Organizations that are serious about improving their agile capability can do a whole lot better than trying to force fit Agile and CMMI together.

**Disciplined Agile to the Rescue**

The Disciplined Agile (DA) process decision framework is a people-first, learning-oriented hybrid agile approach to IT solution delivery [5]. It has a risk-value delivery lifecycle, is goal-driven, is enterprise aware, and is scalable. DA is a hybrid approach which extends Scrum with proven strategies from Agile Modeling (AM), Extreme Programming (XP), Unified Process (UP), Kanban, Lean Software Development, Scaled Agile Framework (SAFe), and many other methods. The framework provides contextual advice showing how agile strategies work together in a light-weight, streamlined manner from beginning to end. It describes the tradeoffs associated with hundreds of agile strategies, enabling you to tailor your software process to effectively address the situation in which you find yourself. By describing what works, what doesn't work, and more importantly why, DA helps agile teams increase their chance of adopting strategies that will work for them in the situation that they face. In short, the DA framework does the heavy lifting when it comes to showing how Agile works in practice.

Earlier we described the challenges associated with combining Agile and CMMI in practice. Now let’s explore strategies for how the DA framework can be applied to help organizations move towards greater maturity in their Agile capability.

**Strategy #1: Adopt a flexible, context sensitive approach.**

The DA framework originally came about from empirical observations of dozens of teams apply agile and lean strategies in dozens of organizations working in different domains around the world. Although there were similarities between how these teams worked every team worked in a unique manner, and every team had spent considerable time and effort determining how to do so. And every team still had improvements to make, and many were struggling with doing so because they didn’t have the process background to identify candidate options. It was clear to us that agile/lean teams could benefit from light-
weight process guidance that described the range of options available to them.

To do this DA adopted a goal-driven, or capability-based, approach. In many ways these process goals are analogous to CMMI Process Areas (PAs), albeit with a time-oriented focus rather than a subject-oriented focus. The delivery goals are summarized in Figure 1. The diagram indicates that when a team is in the Inception phase (sometimes called Sprint 0 or Initiation) that they must address goals such as forming the initial team, aligning with the enterprise direction (e.g., follow your corporate roadmaps and guidelines), and explore the initial scope amongst other things. There are also process goals for the Construction and Transition phases and ongoing goals that apply throughout the entire lifecycle. Every team addresses these goals in some way, but does so in a different manner and will evolve their approach over time as they learn from experience.

The DA framework makes your process choices explicit and provides guidance for selecting the options most appropriate for your team. This guidance helps teams to get going in the right direction and provides options when they realize

![Diagram](image.png)

Figure. 2. The process goal diagram for Explore Initial Scope.
that they need to improve. Because we observed that teams find themselves in a range of situations the framework supports multiple lifecycles [6]—a Scrum-based lifecycle, a lean lifecycle, a continuous delivery lifecycle, and an exploratory (“Lean Startup”) lifecycle.

**Strategy #2: Marry the what with the how**

To provide teams detailed process guidance the DA framework introduced the concept of process goal diagrams [7]. A process goal diagram depicts the process factors that should be considered when addressing the goal and then a representative list of options for those goals. Because people around the world are constantly improving upon and identifying new practices and strategies the list of options presented in the goal diagrams cannot possibly be definitive. Instead they represent the range of options available, letting people know that choices exist (regardless of what prescriptive methodologies like Scrum imply) and that sometimes some choices are distinctly better than others (again, regardless of what prescriptive methodologies imply).

Figure 2 depicts the goal diagram for **Explore Initial Scope**, a goal that you should address at the beginning of a project during the Inception phase. Where some agile methods will simply advise you to populate your product backlog with some initial user stories, the goal diagram makes it clear that you might want to be a bit more sophisticated in your approach. What level of detail should you capture, if any (a light specification approach of writing up some index cards and a few whiteboard sketches is just one option you should consider)? What view types should you consider (user stories are one approach to usage modeling, but shouldn’t you consider other views to explore the data or the UI)? Suggested starting points, are shown in bold italics. The framework makes it clear that agile teams do more than just implement new requirements, hence our recommendation to default to a work item list over Scrum’s simplistic Requirements Backlog strategy. Work items may include new requirements to be implemented, defects to be fixed, training workshops, reviews of other teams’ work, and so on. Finally, the goal diagram makes it clear that when you’re exploring the initial scope of your effort that you should capture non-functional requirements—such as reliability, availability, and security requirements (among many)—in some manner.

There are several fundamental advantages to taking a goal-driven approach to agile solution delivery:

1. **Process decisions and choices are explicit**. This approach makes your process options very clear and thereby makes it easier to tailor your process for the situation you find yourself in.

2. **Tactical agility at scale is enabled**. A goal-driven approach enables effective scaling by guiding you through process tailoring to reflect the realities of the scaling factors that you face. These scaling factors include team size, geographic distribution, regulatory compliance, tech-
To date there have been three major release tiers of the Disciplined Agile framework:

1. **Disciplined Agile Delivery 0.x.**

   The framework was originally developed at IBM Rational from early 2009 to June 2012. The IBM team worked closely with business partners, including Mark Lines, and was led by Scott Ambler. IBM Rational Method Composer (RMC) currently supports an early, 0.5 version of the DA framework.

2. **Disciplined Agile Delivery 1.x.**

   The DA 1.0 release occurred in June 2012 with publication of the first DA book, Disciplined Agile Delivery [2]. Evolution and publication of the DA framework continued at the Disciplined Agile site starting in August 2012. Ownership of the DA framework intellectual property effectively passed over to the Disciplined Agile Consortium [3] in October 2012, a fact that was legally recognized by IBM in June 2014. The focus was on the software delivery process.

3. **Disciplined Agile 2.x.**

   This is the current version of the framework, initially released in August 2015. The focus is on describing a flexible, context-sensitive approach to the entire IT process.
If your agile teams are to be capable then they need to know what process factors they need to address, what their options are to do so, and what the tradeoffs are of these options.

**Strategy #3: Show how to apply proven strategies together**

Many teams new to agile will adopt a method like Scrum or SAFe as if it’s a recipe, ignoring advice from other sources and thereby getting into trouble. This is exacerbated when you weave in CMMI’s process area approach into mainstream agile strategies. Many SEPGs will organize their improvement efforts around the process areas, which has the effect of local optimization of each process area at the expense of the overall flow and efficiency within your process. The fact that CMMI-Level 5, which few organizations achieve in practice, focuses on process optimization belies this point.

The DA framework adopts practices and strategies from a range of existing sources and provides advice for when and how to apply them together. The framework effectively weaves the CMMI process areas together. For example, the *Produce a Potentially Consumable Solution* process goal includes process factors such as *Needs Exploration*, *Solution Exploration*, and *Planning* which map to aspects of *Requirements Development (RD)*, *Technical Solution (TS)*, and *Integrated Project Management (IPM)* CMMI process areas respectively. Via the combination of lifecycles and process goals the DA framework is able to present time-oriented advice, e.g. at this point in time here’s what you need to do and here’s how it fits together, that is actionable by agile teams. In short, the framework takes the mystery out of how all this agile stuff actually works in practice.

**Strategy #4: Optimize the whole**

A Disciplined Agile IT department is a flexible learning organization that is responsive to the needs of the organization(s) that it supports in a financially effective manner. DA 2.x extends disciplined agile delivery strategies to the entire IT workflow [9]. We began the development of DA 2.x in the Autumn of 2014 and the work continues to this day. DA 2.x is based on several important observations. First, every organization is unique, and every IT department within each organization is also unique. Second, IT departments are dynamic complex adaptive systems that evolve over time. Third, the components of IT departments – teams and sub-departments – also evolve over time. Fourth, these components, when left to their own devices, are often not well aligned with each other. Worse yet, these groups work under their own locally optimized “improvement strategies.” This misalignment is caused by competing leadership visions (or less delicately, by “politics”) and exacerbated by disparate bodies of knowledge (BoKs) within our industry: The Agile Manifesto; The Project Management Institute’s BoK; The Data Management BoK; The Business Analysis BoK; The Open Group Architecture Framework (TOGAF); The Information Technology Infrastructure Library (ITIL); and many more. Although all of these bodies of work provide valuable insight, they each provide their own locally optimized view of how things should work. These views overlap, provide inconsistent advice, and are often focused on a single specialty. For example, the BABoK provides a business analyst-centric view, TOGAF provides an architecture centric view, the DMBoK provides a data management centric view, and so on. All great views, but when combined with one another, which is a common approach in most organizations today looking for “best practices”, they prove to be an ineffective mishmash. DA 2.x provides a coherent, integrated, high-level view of how an IT department may address all of these key areas in a consistent, flexible, and evolutionary manner. Wherever possible DA 2.x references the effective ideas in these BoKs and supplements them with strategies that are more consistent with modern agile approaches. Figure 3 overviews the workflow for Disciplined Agile IT.

DA 2.x has been arranged into components called process blades – such as *Enterprise Architecture, Data Management*, or *Reuse Engineering* – that in effect are process areas focused on major IT activities. Just like each delivery process goal encompasses a collection of potential strategies or practices, similarly so does each process blade. The point is that for true software process improvement organizations must look at the entire IT process, not just their software development processes. Focusing on software development alone is like focusing on building a race car engine. It’s interesting and you can create a very fast and efficient engine, but if you take that engine and put it into the body of a tractor you’re not going to win any races. What you really need is to create an IT race car and a racing team to race it, then you’ve got a shot at winning the race.

**We Can Do Better Than Agile CMMI**

The Disciplined Agile framework can be instantiated in a CMMI-compliant manner if you need to [10]. We’ve even mapped CMMI to Disciplined Agile, showing that DA fully supports the process areas of CMMI-Dev [11]. We definitely see the value of applying agile strategies to improve existing CMMI implementations, but we can’t honestly recommend applying CMMI to existing agile teams. There are many challenges with combining agile and CMMI – although it’s possible to overcome these challenges, in most situations it likely isn’t worth the effort.

Most organizations that have embraced agile ways of working have moved beyond the false promise of repeatable processes and instead focus on being able to produce repeatable results. To do this they adopt a flexible, context-sensitive process; they marry the what with the how by making process-oriented decisions, and the options associated with them, explicit to their teams; and they focus on evolutionary learning and improvement over the false consistency of process conformance.
Scott W. Ambler is a Senior Consulting Partner of Scott Ambler + Associates, working with organizations around the world to help them to improve their software processes. He provides training, coaching, and mentoring in disciplined agile and lean strategies at both the project and organizational level. Scott is the (co)-creator of the Agile Modeling (AM), Agile Data (AD), Disciplined Agile Delivery (DAD), and Enterprise Unified Process (EUP) methodologies. He is the (co)-author of several books, including *Disciplined Agile Delivery*, *Refactoring Databases, Agile Modeling, Agile Database Techniques, The Object Primer 3rd Edition*, and *The Enterprise Unified Process*. Scott blogs regularly at DisciplinedAgileDelivery.com and he can be reached at scott [at] scottambler.com and tweets @scottwambler.

Mark Lines is Enterprise Coach and Managing Partner at Scott Ambler + Associates. He is co-creator of the Disciplined Agile Delivery (DAD) framework and co-author with Scott Ambler of *Disciplined Agile Delivery: A Practitioner’s Guide to Agile Software Delivery in the Enterprise*. Mark helps organizations all over the world transform from traditional to agile methods. He writes for many publications including the Cutter Consortium and is a frequent speaker at industry conferences. Mark blogs about DAD at DisciplinedAgileDelivery.com. He is also a Founding Member of the Disciplined Agile Consortium (DAC), the certification body for disciplined agile. He can be reached at mark [at] scottambler.com and tweets @mark_lines.

REFERENCES