

Process, Dogma and Flexibility

There is a natural tension that occurs in organizations that are striving to be good agile practitioners. This is especially true in organizations with a history of good process management. Organizations that are serious about process management often use the Capability Maturity Model Integration (CMMI) as a means of evaluating their place on the maturity scale. However, agile practices can be somewhat antithetical to the CMMI approach.



Source: cmminstitute.org

Most organizations find it most challenging to get from CMMI level 3 to level 4. Unfortunately, agile and CMMI don't converge until level 5 of CMMI maturity. Prior to that stage, organizations develop rigid processes that require a rigorous set of steps before changes can be ratified.

Agile practices, on the other hand, are best left to the team level. There will always be certain processes that are managed at the organizational level, but the more that agile teams can be autonomous, the better. When agile teams have all the skills and knowledge they need to fully execute their work, that particular group of people can continually experiment to find what works best for them. How skills are distributed, social tendencies, and the type of work being performed will all drive decisions along with many other more subtle variables.

The battle cry of many agilistas is: **Individuals and interactions over processes and tools!** It's right there in the Manifesto. What often gets lost is the word "over." It is carefully chosen. "Over" means don't allow processes and tools to interfere with the open communications between team members. In other words, don't create barriers to people working together to find the best way to get things done right. It is important to distinguish this from another common battle cry which is -- *don't interfere with us doing what we want!* This is a bogus battle cry because of that last word ("want"). It does not mean that there should not be any processes (which I have heard argued by agile team members). This is also what brings us to CMMI level 4.

Immature agile practitioners often believe that it is okay to go by what the team "feels," using intuition instead of evidence to make decisions. This is a mistake. Granted, intuition and feelings can be the source of a hypothesis, but evidence is needed to either validate it or disprove it. Here is an example.

One of the teams I work with was concerned that they were spending too much time doing maintenance work and lifecycle management work. This is what we refer to as "toil" because it is the type of work that generally doesn't add any value to our stakeholders other than keeping the lights on. Our stated goal was to get to 50% project type work and 50% toil. Our sense was that we weren't anywhere close to that.

We break our work into tasks that we call "stories." The idea behind a story is that it is a unit of "customer" value that is testable and in our case deployable so that the intended recipient can actually benefit. I put customer in quotes here because in the case of this particular team, most of the work goes to benefit other software development teams within the organization. As is often the case, the recipient of value from some effort may be generically referred to as a customer, but is not necessarily the organization's primary commercial beneficiaries. In any case, for us to generate evidence to see how we were doing, we needed to first understand the value of what we do. We ended up with this list:

Cost Reduction - reduce AWS costs and other infrastructure

Stability - work that improves our ability to discover and recover from operational problems

Reliability - work that ensures the consistency of the environment

Availability - Uptime of infrastructure components

Maintenance - upgrades, fixes, assistance, patches, etc.

Build Faster - tools that allow developers to deploy more apps more quickly with less complexity and greater assurance of success

Scaling - ability to more effectively support current demand for all of our offerings

Security - ensuring that bad guys stay out while good guys can get in

Consulting - helping others learn about how to use our infrastructure

Monitoring - watching stuff to ensure stability, availability, cost reduction, scaling, etc.

The next step was to start tagging each of our stories with one or more of these tags. These tell us the value of our work so that we can determine if it is toil or actually improving on one of these value categories (maintenance is always toil).

After a month of collecting data, we started to discover a few things. First, we learned that just by identifying what we were working on and what we had planned in this way, we changed our behavior. Team members began picking up the “right” work to balance our efforts. The other thing we noticed is that 50% may be a good goal over time, but what that timeframe is may not always be clear. There are times when we committed to doing lots of work that falls into the toil category and we did it because we agreed that it was necessary under the circumstances.

The important takeaway here is that the team decided how to value their work and the team decided how to best distribute their efforts. The only thing that was encouraged by the organization in this case (and even that is not a prescription) is that we use evidence to drive our decisions.

This noninterference by the organization allows our teams to find their own path to maximizing effectiveness as measured by productivity+customer value. A CMMI level 3 or even 4 organization would most likely have external (outside the team) oversight either via actual review by people or multi-step processes. It does not have to be this way, but most organizations that are serious about continuous improvement have personnel dedicated to quality.

Quality professionals usually have a background in certifications like SOC2, TS-16949, or Six Sigma. All of these impose a fairly rigid framework around the process of change. Done correctly, these methodologies can maintain flexibility, but in my experience as both an executive and a consultant, they usually don't.

Summary

If you have any sacred cows in your organization, ask yourself why they are sacred? What would happen if they were challenged. If our goal is continuous improvement, we will need maximum flexibility. That means any dogma should be questioned, especially if it is standing between people and their interactions.

Moving fast does not mean we can skip over the gathering of evidence to support our hypotheses. If gathering real data slows us down, so be it. Data is the only means by which we can evaluate the effectiveness of our choices and better understand the realities of our situation. Not everything is intuitive. The evidence often delivers surprises. When that happens, it is always a good thing -- even when the surprises reveal unfavorable conditions. You can't fix what you can't see.

Finally, giving operational teams the autonomy to find their own path to greatness will open the whole organization up to new possibilities. Many agile organizations periodically hold something called a “show and tell” (it may go by other names such as sprint review). This is when representatives from each team share what they have accomplished and what they have learned. They could also use this opportunity to share a new way of doing things. This cross-pollinates other teams with ideas that can help them too. It’s the classic case of a rising tide raising all boats.

Author



Mr. Bellinson has been working in information technology positions for 30 years. His diverse background has allowed him to gain intimate working knowledge in technical, marketing, sales and executive roles.

Most recently, Mr. Bellinson finds himself serving as President of a BPM related software start-up company called UnaPage that provides solutions based on Microsoft SharePoint. From 2008 to 2011 Bellinson worked with at risk businesses in Michigan through a State funded program which was administered by the University of Michigan.

Prior to working for the University of Michigan, Mr. Bellinson served as Vice President of an ERP software company, an independent business and IT consultant, as chief information officer of an automotive engineering services company and as founder and President of a systems integration firm that was a pioneer in Internet services marketplace.

Bellinson holds a degree in Communications with a Minor in Management from Oakland University in Rochester, MI and has a variety of technical certifications including APICS CPIM and CSCP.