

# Agile Principles as a Leadership Value System: How Agile Memes Survive and Thrive in a Corporate IT Culture

Steven W. Baker  
Software Methodologist, DTE Energy  
bakersw@dteenergy.com

Joseph C. Thomas  
Program Manager, DTE Energy  
thomasjc@dteenergy.com

## Abstract

*What agile approaches are effective to introduce and sustain meaningful change in an IT organization? How can agile approaches be leveraged beyond project work? What processes are involved in introducing agile approaches into an organization, and how might a leader become more agile in adopting agile approaches?*

*DTE Energy's Information Technology Services (ITS) organization continues to embrace and extend the agile mindset throughout its culture. Our organization first embraced agility eight years ago, and we continue to refine our agile methodologies as we deliver and support world-class business solutions. The more we apply agile methods to our project work, the more we find them to be of value to leadership as well as operational work.*

*In this report we discuss our experiences applying agile methods outside of software development projects in a broader context of operational work and organizational leadership. We assert that agile principals and techniques associated with software development projects can be readily applied in other types of organizational work and in creating and sustaining an effective leadership culture.*

## 1. Context and History

DTE Energy, a Fortune 300 company headquartered in Detroit, Michigan, USA, is a diversified energy company involved in the development and management of energy-related businesses and services nationwide with \$9 billion in annual revenue and 11,000 employees. DTE Energy's Information Technology Services (ITS) organization, consisting of over 600 people, provides leadership, oversight, delivery, and support on all aspects of information technology (IT) across the enterprise.

One of the company's strategic IT goals is to produce and support world-class software. As a component of ITS, our software engineering (SE) organization of over 350 people includes development, test, business analysis, project management, and governance groups.

Another strategic IT goal is to grow professional leaders. Our ITS organization includes approximately 60 leaders, most having direct reports and others providing technology thought leadership and implementation

expertise. Additionally, we have approximately 50 project managers leading our various initiatives.

In 1998 a part of the SE organization began using lightweight methods to enable short-cycle, iterative, and incremental software delivery. Our company embarked upon a journey to reengineer its business process, and ITS project teams needed to quickly respond to change along the way. Beginning with a few projects led by willing and able people, we embraced the principles succinctly stated in the Agile Manifesto [1], and we used approaches clarified and codified in eXtreme Programming (XP) [2].

After several years of experimentation, learning, and project success, a cadre of agile activists including experienced project managers and agile thought leaders emerged. Several transitioned into organizational leadership positions. With a fresh perspective, we hypothesized that leveraging agile principles in a wider organizational context could be effective.

In the course of our experiential learning as an agile leadership team, we realized that many approaches and techniques endured while others had limited success or just didn't work at all. Sometimes an approach that seemed to be effective in one situation did not take hold on a wider basis with other leaders and in other areas. Sometimes an approach that did not initially work was later found to be effective in a different time and context.

We applied a core set of agile leadership principles that were generally accepted by the ITS leadership team as a whole; yet in the agile spirit we used an iterative process of trying out different approaches and techniques to find what works for us in particular situations.

Introducing agile approaches into an organization is about managing organizational change. This paper presents our experiences with introducing and applying agile software development techniques in operational work and leadership, and our observations on how people accepted and adapted these techniques over time.

## 2. Survival of Ideas in the Wild

The introduction of new ideas and the adoption of them by people is analogous to the biological concept of "survival of the fittest" or natural selection. This analogy is embodied in the concept of the *meme*, first introduced by Richard Dawkins [3] and subsequently expanded upon

by others (for example, Richard Brodie [4], Hans-Cees Speel [5], and Susan Blackmore [6]).

A meme is an idea that exists in the minds of people; it guides their thoughts and actions, and propagates from one person to another. Over time, memes are replicated from one person to another within a group in a manner comparable to how genes are replicated within a population of a particular species.

Memes are picked up by the minds of individuals and survive and thrive because they “make sense” and provide value in solving problems or achieving goals. As memes are replicated, they establish themselves within a group [7] and become an inherent part of that group’s culture.

## 2.1. Distinctions as “Memetic” Material

In our efforts to expand adoption of agile techniques across software projects as well as into other aspects of our organization, we often found ourselves *evangelizing* the cause with elaborate explanations of agility and its principles. Trying to convince an entire group of the need to understand and apply a bundle of techniques was a difficult challenge. Incrementally teaching useful aspects of a larger, more complex model can be more effective than teaching (or preaching) the entire model at once [8].

An effective leadership technique is to create *distinctions* through conversation. Distinctions provide for the identification, categorization, and recognition of important ideas, and enable the introduction of those ideas as memes into the group. Mastering the art of leadership conversation is an effective leadership technique [9].

We have learned that it is often more beneficial to introduce agile approaches as individual techniques. We look for situations where an agile approach is helpful, and create distinctions to show how a specific principal or technique is of value. Applying an incremental, iterative approach to using agile techniques in organizational work is just as effective as applying an incremental, iterative approach to deliver software in project work.

To that end, the concept of memes as it relates to natural selection of the fittest ideas provides a useful metaphor to understand how the initiation, adoption, and persistence of specific agile techniques occurs in an organization. A leader who seeks to nurture the agile skills of others may use this to accelerate the deployment and acceptance of the fittest agile memes.

## 2.2. Survival of the Fittest Memes

As a report on our successes and challenges learning how to introduce and adopt agile techniques in the organization, we present a selection of our experiences as a series of memes. We explore each meme as a story of how a particular agile technique was introduced, applied,

and succeeded in the context of actual work and leadership experience in the organization. In our experience, not all memes have survived, but this experience report focuses on those that have thrived as a leadership technique in our organization.

In each section that follows, we first describe a meme as a concept and relate it to an agile principle, practice, or technique. We then describe how we originally introduced the meme in the context of project work. Next we describe why we felt it useful for and how we applied it in the context of an organizational opportunity or problem. Lastly, we describe what ultimately happened, what we learned, and how the meme thrived over time.

## 3. The “Short Cycle Release” Meme

A core software development agile principle is to “satisfy the customer through early and continuous delivery of valuable software” [10]. This encourages project teams to deliver value early in the project lifecycle, to seek and leverage feedback from the customer frequently, and to reach closure on an achievable set of commitments before moving to the next.

### 3.1. How the Meme was Introduced

Our agile project teams and business partners have come to appreciate the rhythm of shorter, more frequent and time-boxed release schedules. We were able to more fully engage with the business because they made the decisions on what features to focus on over others. We delivered working software quickly and enabled decisions about pushing forward (incremental breadth) or cycling back for refinements (iterative depth).

### 3.2. How the Meme Survived

Some of our agile project managers have transitioned from leading projects to leading groups and organizations. We started by leveraging many of the *de facto* leadership techniques like mission statements, goals and objectives, and periodic reviews to drive the organization forward. While these were generally effective, we found that the process felt very different from what had worked for us previously in project work.

Our SE organization had a number of initiatives underway that would enable us to be more effective at delivering better software. These efforts included building communities of practice, capturing and leveraging best practices, and establishing consistent processes. As we considered our organizational objectives and the corresponding work, we realized that we had a backlog feature list we needed to manage and that we could deliver results in an iterative and incremental fashion.

We determined to be more deliberate about applying agile approaches that had worked so well for us in software development project work to our organizational work. We often say that “anything that takes time and costs money is scope.” Recognizing that our effort to build community amongst our software development professionals and to build consistent software development processes was scope, we should manage it like we managed project scope. As a team of leaders, we determined to run the organization like we had run projects. We used our project release cycle as the metaphor for our organizational work.

We decided to release results in short cycle, time-boxed iterations. In our software project culture, three-week iterations worked best so we determined we would also use three weeks for the length of each iteration.

Each leader worked with their group to assemble a list of features and deliverables, develop work estimates, and establish priorities. Each leader then allocated work for items on the feature list to specific three week iterations and commenced delivery work with their groups.

With short time-boxed cycles, we were able to make measurable progress and provide visibility into the work that we were accomplishing. We produced the types of work products that we needed because we had clarity into what we wanted and the “pleasant pressure” of a short timeframe within which to produce them.

Scope varied over time based on the needs of the organization and the point of time in the calendar year. Our individual backlogs included community of practice meetings, presentations, performance evaluations, vendor and contract renewals, and holiday gatherings.

Yet with our separate lists we sometimes lost sight of the big picture and the ways in which each of our groups could help the greater good. We were running multiple Scrum-like efforts in parallel; it dawned on us that we could view this as a Scrum of Scrums.

We pulled our lists together into an artifact that we called the *List of Lists*. This enabled us to collaborate together as a leadership team increasing the visibility of our key deliverables, ensuring our focus was on the most important work, coordinating our efforts, helping each other, and better understanding how our groups contributed to the overall organization.

### 3.3. How the Meme Thrived

We learned that certain aspects of our approach to project work worked well while others did not and that we needed to be flexible in adapting agile approaches to organizational work. Short cycle releases and the feature backlog list worked well.

In our project work, we effectively use a capacity-based planning approach where we establish scope by balancing the available work hours of the project team

against the estimated hours of work for the highest priorities on the feature backlog list.

The capacity-based planning approach was not effective when we applied it to our List of Lists because it was difficult to determine the available work hours of the group (due to the magnitude of wide organizational responsibilities); therefore, estimating feature work effort was moot. We adapted by intuitively assessing the capacity of the people to do the work on priority features and using a level-of-effort approach to deliver results. In the level-of-effort approach, we emphasized the necessity to produce a sufficient version of work product within the time-box even if the work product was incomplete or imperfect. When additional work was necessary, we merely queued it up on the feature backlog list and addressed it in a subsequent iteration.

We measured our ability to successfully deliver on our iteration scope by scoring “hits” and “misses”; in this way we better understood what we were able to deliver in an iteration cycle. Over time, we improved on our ability to deliver on our iteration commitments. We also found that permitting adjustment of iteration commitments in the first third of the iteration cycle in response to environmental factors was effective in managing expectations of what we were actually able to deliver.

The memes of short-cycle delivery and visible work management took hold within our leadership team, most prevalently within those having led agile projects before. The most pressing challenge for us was to hold ourselves responsible for our commitments and to hold one another accountable for our delivery “hits and misses,” because we also played the role of our own business partners.

## 4. The “Incremental Improvement” Meme

A key agile leadership principle is to “expect uncertainty and manage for it through iterations, anticipation, and adaptation” [11]. This meme encourages project teams to deliver a “good enough” product today rather than seek a “perfect” product in the distant future. Teams learn from each release, are better positioned to make adjustments, and in the long run, deliver a product that more closely meets the needs of the customer.

### 4.1. How the Meme was Introduced

The company had a situation where key business processes were likely to change quickly and dramatically. The IT team supporting these processes established a foundation of agile processes and technologies that enabled us to anticipate potential changes, begin development based on what we knew at the time, and rapidly adapt to the final decisions as they became clear.

## 4.2. How the Meme Survived

As many of our agile project leaders became IT organizational leaders, we realized that the agile principles that worked so well in our individual projects were inconsistently applied from one project to the next. As we assembled project teams from our pools of developers, business analysts, testers, and project managers, each team's operating processes took time to stabilize. We felt that more uniformity in our agile solution delivery process would enable greater adoption of agile methods across project teams, and would provide a strong foundation for continuous process improvement.

We considered designing and deploying a new agile methodology to project teams across the organization. This felt like a heavyweight approach where experts on high would dictate to practitioners how to do their work. We decided against that approach as it would tend to inhibit flexibility and adaptability to fit situational needs.

Instead, we opted to work with project teams to capture the current state of their processes and to rally around what was working for us in our project teams' culture. This embodied the "good enough" meme and would provide a baseline for process improvement. Not all project processes were perfect, but we had a set of best practices that we knew were working by virtue of project team success in applying them. By writing down what we were actually doing, we then could assess opportunities and address those with the most potential value.

After capturing existing best practices as a baseline, we began a series of iterative improvement cycles. We partnered with six project teams to pilot the process, adapt it to their needs, and give back practical improvements.

We developed a common process framework that had been field-tested, and we decided to continue improvements using the Software Engineering Institute's Capability Maturity Model Integration (CMMI) [12] as a reference model for software engineering best-practices.

We continued the use of short, release-based improvements to the process in use by our projects, ultimately demonstrating CMMI Maturity Level 3 performance [13, 14].

## 4.3. How the Meme Thrived

Reflecting on our process improvement journey, we recognize that an incremental improvement approach as a means to respond rapidly to change as it emerges is a powerful meme that is thriving in our leadership team. These distinctions have taken hold in the minds of leaders across the organization, even those without hands-on experience in agile software development.

However, the idea of beginning work without a clear, complete specification (let alone delivering an incomplete

or imperfect solution) continues to make some people uncomfortable. Yet nothing succeeds like success; even late adopters are now recognizing the value of quick delivery and improvements based on the learning that takes place along the way.

## 5. The "Daily Team Meeting" Meme

A core agile software development principle is that "business people and developers must work together daily throughout the project" [10]. This meme emphasizes the value of people interacting together to improve communication and to increase the level of shared understanding and ownership.

### 5.1. How the Meme was Introduced

This meme had taken hold in the minds of many of our agile project leaders early in our agile project experience. One particular project is a good illustration. We began a project with a customer who had not yet been through an agile project, and we experienced some challenges in gaining their confidence that we could deliver a critical, complex business application in a timely manner.

To heighten the visibility of our project work, and to shorten the idea-to-solution feedback loop on complicated calculations and business rules, we convinced the customer that co-locating the software development team in their office would greatly improve communication and enable us to rapidly deliver and respond to change. Co-location of the project team with the customer and the subject matter experts was effective; the project was a success, and the once-hesitant customer became one of our most ardent supporters.

One of the most effective techniques in that particular situation, as well as in other projects, was the daily huddle or daily team meeting. In less than fifteen minutes, the team articulated what they had recently accomplished, what they planned to work on that day, and what obstacles they most needed to address to sustain forward progress.

### 5.2. How the Meme Survived

From an organizational leadership standpoint, we found that the lack of visibility into the work being performed by our various groups made us uncomfortable. We missed the daily interaction and discussion about the day's work and how it related to our overall objectives.

As a leadership team, we recognized that the List of Lists was a valuable tool but not sufficient by itself to ensure that we were making the progress we sought and to focus on the right things in the midst of day-to-day pressures and organizational challenges.

We started meeting in the cafeteria each morning to discuss the day ahead, our priorities, and the work we were addressing. We focused on what was most important to the group and how we could help one another.

Other leaders noticed our daily presence in the cafeteria, and wondered what we were doing each morning and why we were having so much fun together. Dubbed the “Jedi Council,” we invited them to join us and see for themselves how we made each day both more enjoyable and more productive.

### 5.3. How the Meme Thrived

Our daily meeting in the cafeteria eventually led to the propagation of the meme as we dispersed into other parts of the organization and continued to hold daily meetings in each of our groups. Other leaders picked up on the meme and are successfully applying the approach.

Yet this meme was not without threats in our corporate environment. Well-meaning people might cancel a daily team meeting when a senior leader was unable to attend. Or, some members of a team would decline to attend when the group leader was unavailable.

Ultimately, we are learning the value of the daily team meeting as groups continue to meet even in the absence of their leader or other key group members. We appreciate the value of the group interaction even when the group leader is not available; after all, the benefit of the daily huddle is to interact with the rest of the group.

In general, the meme of the daily team meeting has been adopted throughout the organization; small and large groups alike now interact in this way each day. It is no longer a question of *why* we meet daily but *how* we make that time most effective as leaders and as groups.

## 6. The “Pull the Risk Forward” Meme

Another agile principle is to create a spike solution to “reduce the risk of a technical problem” [2]. A *spike solution* is a preliminary or experimental effort to prove that a specific technology or an approach will actually work in a particular situation. Often, it is natural to succumb to the tendency to complete easy work first to get a quick win, to demonstrate progress, and to gain the trust of the customer. In contrast, this meme encourages teams to address the difficult or unknown aspects of an effort first in order to uncover potential problems sooner. This improves confidence in estimates and increases confidence in ultimate success. The sooner you know about a problem, the more time you have to solve it; the more time you have to solve a problem, the greater the number of options that are available!

## 6.1. How the Meme was Introduced

This meme took hold early in our agile software development journey. On one project we needed to render graphs and charts on a web page based on near-real time data. While components were commercially available for this, we had no experience with this type of functionality and our confidence in our estimates was relatively low. By spiking this feature in an early phase of the project, we demonstrated that the interfaces worked, the component satisfied our needs, and we better understood the amount of effort required to deliver the working software.

## 6.2. How the Meme Survived

As agile leaders, we face a variety of risks and uncertainties each day. From new business initiatives, pressure to do more with less, and mounting budget constraints, we find ourselves needing to mitigate risks regularly. One such challenge is how to effectively leverage our best people across the organization.

Specifically, our senior technical and analytical people were delivering excellent value in their assigned roles as full time project team members, but we were unable to leverage their expertise and lessons-learned to benefit the rest of the in-flight projects in our portfolio.

Our approach to addressing this challenge was to assemble and leverage a pool of technical and analytical experts. This group would be allocated across multiple projects to supplement each dedicated team with additional expertise and organizational support at the right points in the project lifecycle.

We ran the risk of over-promising and under-delivering on the value that a pool of technical experts would deliver. Furthermore, we did not fully understand how to organize the group, how many projects each expert could support, and how to allocate the costs to the projects that the group would support.

Rather than put forth a compelling argument for the case for change, we pulled the risk forward and spiked the approach. We experimented by pulling one technical expert from a dedicated assignment and provided consulting support to several teams at the same time.

As a result, we learned how to brand this service, communicate the benefits, and engage each team in a meaningful way. We increased confidence in our cost estimates and the number of projects each person could support, thereby justifying through experience what it would take to get such a group off the ground.

By pulling risk and uncertainty forward, we built the case for change and demonstrated the value to the organization. We proved our ability to succeed and generated demand for this service. We were thus able to

quickly form the organization with low ceremony and high confidence in the value this group would provide.

### 6.3. How the Meme Thrived

The agile technique of developing a spike as a way to mitigate risk and uncertainty may initially seem like it applies only to technology and software project work. However, using a spike as a way to experiment with an organizational change initiative is an example of how it can be effective in other types of situations.

The pull risk and uncertainty forward meme is now well understood and readily applied throughout all types of projects and across the organization. It's easy to explain the rationale and quickly understood by all. This meme encourages the agile leader to learn by doing and to build momentum through demonstrated results.

## 7. Lessons Learned and Recommendations

To those who seek to foster an effective agile leadership community, the following suggestions may provide some degree of value. Our large IT organization embraced agile methods at a project level and subsequently applied those same agile methods to a variety of organizational leadership opportunities.

First, recognize that it is difficult to get everyone to change how they think overnight. Rather than trying to build understanding about agility as a full set, select and introduce principles that best apply in a given situation.

The agile leader creates distinctions to introduce new ideas, and recognizes situations that provide favorable conditions within which those memes can take hold.

Second, actively engage with the community to monitor the memes that have been introduced. Stay connected with leaders, understand their situations, and learn to recognize which memes are surviving.

The agile leader monitors each situation to see which memes survive, and strives to identify opportunities where those surviving memes can be further propagated.

Third, be alert to situations in which people are not able to internalize and apply an agile principle. Even if the principle is quite appropriate, some people may not appreciate the value or utility of a particular meme.

The agile leader recognizes the situation where she/he should let go of a meme that is not surviving, and instead focus on those memes that have the potential to thrive within the group or organization.

We recommend that agile leaders be deliberate about selecting and applying agile principles; remain engaged with their communities; and be agile in sustaining those principles over time. Create distinctions to introduce agile memes, monitor to see which memes survive, and rally around those that survive so they can ultimately thrive.

## 8. References

- [1] Agile Alliance. *Manifesto for Agile Software Development*. <http://agilemanifesto.org/>
- [2] Wells, Don. *Introduction to Extreme Programming*. <http://www.extremeprogramming.org/>
- [3] Dawkins, Richard. *The Selfish Gene*. New York: Oxford University Press, 1989.
- [4] Brodie, Richard. *Virus of the Mind: The New Science of the Meme*. Seattle, WA: Integral Press, 1996.
- [5] Speel, Hans-Cees. *All About Memetics*. <http://www.hanscees.com/mem.htm>
- [6] Blackmore, Susan. *The Meme Machine*. New York: Oxford University Press, 1999.
- [7] Speel, Hans-Cees. *Evolution, Complexity and Philosophy: Memetics*. <http://web.inter.nl.net/users/hanscees/evolu.htm>
- [8] Whetten, Brian. *Teaching vs. Preaching: The Power of Using Key Distinctions in Integral Leadership Development*. *Integral Leadership Review*, Volume VI, No. 3 - August 2006. [http://www.integralleadershipreview.com/archives/2006\\_08/2006\\_08\\_whetten.html](http://www.integralleadershipreview.com/archives/2006_08/2006_08_whetten.html)
- [9] Krisco, Kim. *Leadership and the Art of Conversation: Conversation as a Management Tool*. Rocklin CA: Prima Publishing, 1997.
- [10] Agile Alliance. *Principles Behind the Agile Manifesto*. <http://www.agilemanifesto.org/principles.html>
- [11] Agile Project Leadership Network. *Declaration of Interdependence*. <http://www.pmdoi.org>
- [12] Software Engineering Institute. *Welcome to the CMMI Web Site*. <http://www.sei.cmu.edu/cmmi/>
- [13] Baker, Steven W. *Formalizing Agility: An Agile Organization's Journey toward CMMI Accreditation*. Agile 2005 Proceedings, IEEE Press, 2005.
- [14] Baker, Steven W. *Formalizing Agility, Part 2: How an Agile Organization Embraced the CMMI*. Agile 2006 Proceedings, IEEE Press, 2006.