ARTICLE

There has been much said and written about the mythical “conflict” between the values and principles of the Manifesto for Agile Software Development\(^1\) (http://agilemanifesto.org/) and those of Software Process Improvement (SPI) embodied in Models such as CMMI (www.sei.cmu.edu/cmmi/). However, there is increasing experience (and evidence at SEPG and other conferences) that shows that both sets of values and principles can be combined to deliver more value to an enterprise than either one on its own. A less well known descendant of the Agile Manifesto is the Declaration of Interdependence (DOI)\(^2\) (www.pmdoi.org) which seeks to extend the Agile Manifesto to non-software products, project management and management in general.

The DOI is particularly interesting in the debate between process and agile proponents because, as it says in its preamble, it comes from “… a community of project leaders that are highly successful at delivering results.” The Agile community would argue that the perceived need for this community to make such a declaration proves that an agile approach can produce results. The process community would argue that it proves that, for project management, something more than the software development focused Agile Manifesto was needed.

Using examples from a current project, this paper seeks to demonstrate that software process improvement projects, such as CMMI implementations, can positively benefit by adopting an iterative “Agile” approach and adhering to the principles of the DOI. In short, this paper argues that software process improvement projects can and must deliver recognizable value and deliver it often. Adopting an agile philosophy can focus the collective efforts on this imperative.

Typically, most SPI projects are planned as waterfall projects using a sequential approach rather than an iterative approach. Further, it is argued that implementing an SPI project in an organization that is following an agile implementation can be faster than the same SPI project in an organization using the traditional waterfall approach.

To facilitate discussion of how an agile approach to software process improvement can be structured, Scrum\(^3\), the methodology will be used here as the example. There are a number of other agile techniques and models that can be used.

Scrum

A brief overview of Scrum is appropriate. In Scrum, a Product Owner is identified at the start of the project to take ownership of the work requirements. The Product Owner may be an actual customer or the customer’s advocate within the development organization. The use of the term “Product” in Scrum does not imply that the work is aimed at producing a software product. Its use is more closely aligned with the term work product. The requirements are described by the Product Owner at a summary level and then prioritized in numerical order (not just the traditional high, medium and low) into a list called the Product Backlog.

Work for a team of 7-9 people is organized into a series of fixed duration efforts called “Sprints.” At the end of each Sprint, releasable, working software (and/or artifacts such as designs at an equivalent level of readiness) must be delivered. The team is organized, not led, by a Scrum master whose role is to remove impediments to the success of the team. The Scrum Master
works with the customer and management to ensure that the work undertaken by the team is consistent with their priorities and meets or exceed their expectations.

Each Sprint cycle starts with a Sprint Planning Meeting at which the team, the Scrum Manager and the Product owner agree on the subset of the Product Backlog that will be tackled during the upcoming Sprint. Each Sprint cycle ends with a Sprint Review meeting where the team present the deliverables back to the Product owner and, ideally, the customer for acceptance or feedback.

During the Sprint, the team are left as much as possible to organized themselves around the agreed task. The term “Scrum” comes from the concept of a short daily meeting of the development team (7-9 people) to communicate the answers to three questions:

- What have you done since the last Scrum (yesterday)?
- What will you do between now and the next Scrum (tomorrow)?
- What got in your way of doing work?

Comparison of Approaches

Table 1 compares the high level approaches of the Agile Manifesto and the DOI with the high level approaches of CMMI.

From Table 1, the first value statement in the Agile Manifesto leaps off the page at most process-oriented individuals and organizations, “We value individuals and interactions over processes and tools.” Many SPI practitioners stop reading the Agile Manifesto right there! Reconciling this statement with a process-oriented model such as CMMI is a necessary first step.

The answer to the apparent contradiction is that individuals and interactions ARE more important than processes and tools in ANY software process improvement project because software process improvement is primarily about changing a person’s behavior. The use of the singular “person” is intentional because this has to occur on an individual by individual basis. The best processes and tools in the world will not improve software unless and until individuals use them to interact with each other. Does that mean that processes and tools are unimportant? Of course not. No programmer would argue that the Agile Manifesto precludes the use of compilers!

So what might an agile SPI project value set look like?

This paper proposes that SPI Projects should be organized under an agile methodology such as Scrum but take the literal text of the DOI as the basis for its approach rather than its antecedent. Table 2 shows how the Generic Goals and Generic Practices of CMMI can be reorganized under the principles of the DOI.

The Engineering Process Group (EPG) and the Sponsor of the SPI initiative should create an environment for the execution of the SPI project that is based upon the principles of the DOI in Table 2. Further, those teams developing and institutionalizing the processes themselves should be guided to give equal weight to the DOI principles when addressing the generic goals and practices outlined in Table 2.

For example, the general practice GP2.4 “Assign responsibility and authority” can be all too easily lost at the lower levels of an organization and turned into “just follow the [bureaucratic!] process [stupid!]”. A DOI philosophy would seek to assign responsibility and accountability as far down the organization as possible.
What does an agile SPI project look like?

An agile SPI project based on the Scrum methodology might have a 4 week sprint cycle. The prioritized goals of the SPI project will form the Product Backlog from which candidate activities will be drawn for each SPI project sprint.

In a client using Scrum for development, it is extremely important to synchronize the SPI sprint cycle with the project sprint cycle. It is likely that the SPI project will require input from development project sprint team members whose availability and ability to plan will be based entirely on their commitments to the current development sprint. Also, in pilot and roll-out phase of the projects, it is virtually impossible to introduce a new process in the middle of a development project sprint. New processes must be introduced into the sprint planning meeting. While this initially seems like a constraint, implementing an SPI project in an agile development organization actually proves to be highly advantageous because, to a large degree, each sprint is almost a complete instance of the development organizations SDLC. Hence, pilot opportunities for new processes on new project occur once per sprint cycle. This enables all parts of a process to be tested more quickly and facilitates multiple pilots in parallel or sequentially to refine processes.

One of our focus companies benefited hugely from this when developing and piloting their Configuration Management (CM) processes. Errors in delivered software builds had caused numerous embarrassments in front of customers despite a high level of customer satisfaction in the software reviewed. The company had developed a culture of “fire fighting” teams to deal with this problem. Process Improvement was difficult because nobody had confidence enough to change the engine of the car while it was speeding along the road to the next urgent deliverable. Taking a sprint approach which piloted CM process changes large and small in manageable chunks on a monthly basis has turned this around in four months.

The SPI Project Sprints should be organized as follows: Develop, pilot, refine, roll-out. Note that this approach requires a minimum of four sprints but, in the two organizations where this approach to SPI projects has been implemented, it has proven to be useful to take a flexible approach so that a complex process or one requiring a lot of new process definition might have 2 or 3 “Develop” sprints before the “pilot” sprint. Whenever this occurs, the rigor of the agile approach must be observed and real value must still be delivered at the end of each “Develop” sprint. Similarly, it can be useful to use multiple instances of the “pilot” and “refine” sprint pair—either to evolve a weak process definition, to build and test tailoring schemes, or to ensure correctness and acceptance across a widely diversified development group.

While Process and Product Quality Assurance (PPQA) activities can and should be carried out during sprints, our focus companies for this paper both instituted a “Sprint Retrospective” meeting to follow the Sprint review. This is a team-only meeting which is intended to review the just-completed sprint and identify best practices and opportunities for improvements. Sprint retrospectives in both companies have proved to be ideal for PPQA monitoring purposes. Indeed, one of the companies reports monthly to senior management on number of projects reviewed and number of process “non-compliances” reported.

The “Customer” for most SPI projects is usually the senior management team who is paying for it (or seeking to spend its budget on something else!). An agile-like approach to SPI projects ensures that the senior management will see value being delivered every sprint or they will be right to hold the SPI team accountable and/or change priorities for the next sprint.
Innovation

A company using Scrum or some other agile approach for software development brings another huge opportunity to the SPI team. Effectively, the agile approach implements a very short software development life cycle (SDLC). Often, in putting together an appraisal program for a company’s CMMI initiative, it is difficult to get enough instances of the development team exercising new processes in the desired time for those processes to be considered institutionalized e.g. “We have project durations of two years and the requirements were one month before the new requirements process was introduced. We won’t use it again for 23 months!” That’s a bit extreme but the challenge can be real.

In an agile company, most of the SDLC recurs on a frequent basis. Hence, it is realistic for the SPI team to introduce new processes one month, get feedback and reapply slightly different processes next month. This greatly increases the speed with which processes can be evolved, fine tuned or optimized to the needs of the organization. Fast-paced innovation in process definition!

Conclusion

In conclusion, instead of worrying about the challenges that agile approaches appear to present to traditional process improvement approaches, there is significant value to be gained in adopting some of the thinking of agile and DOI approaches. At the end of the day, process improvement practitioners should keep an open mind and apply techniques and thought processes that can deliver value. The agile manifesto and the DOI certainly offer some new tools.
This information in this article was originally presented, by the author, at the SEPG 2007 Conference in Austin, TX, March 29, 2007.

For further information on this topic or to talk to a DCG expert, contact us at 610.644.2856 or send inquiry e-mail to info@davidconsultingroup.com
Table 1: Comparison of alternative approaches to software project excellence

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: Individuals and interactions over processes and tools Working software over comprehensive documentation Customer collaboration over contract negotiation Responding to change over following a plan That is, while there is value in the items on the right, we value the items on the left more&quot;.</td>
<td>&quot;We follow these principles: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. Business people and developers must work together daily throughout the project. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Continuous attention to technical excellence and good design enhances agility. Simplicity—the art of maximizing the amount of work not done—is essential. “The best architectures, requirements, and designs emerge from self-organizing teams. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;We are a community of project leaders that are highly successful at delivering results. To achieve these results: We increase return on investment by making continuous flow of value our focus. We deliver reliable results by engaging customers in frequent interactions and shared ownership. We expect uncertainty and manage for it through iterations, anticipation, and adaptation. We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference. We boost performance through group accountability for results and shared responsibility for team effectiveness. We improve effectiveness and reliability through situationally specific strategies, processes and practices.&quot;</td>
<td>&quot;The quality of a system or product is highly influenced by the quality of the process used to develop and maintain it.&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GG 1 – Achieve the Specific Goals
GP 1.1 – Perform base practices
GG 2 – Institutionalize a Managed Process
GP 2.1 - Establish an organizational policy
GP 2.2 – Plan the process
GP 2.3 – Provide adequate resources
GP 2.4 – Assign responsibility and authority
GP 2.5 – Train people
GP 2.6 – Manage configurations
GP 2.7 - Identify and involve the relevant stakeholders
GP 2.8 - Monitor and control the process
GP 2.9 - Objectively evaluate adherence
GP 2.10 - Review status with higher level management
GG 3 – Institutionalize a Defined Process
GP 3.1 – Establish and maintain the description of a defined process
GP 3.2 – Collect Improvement Information
GG 4 – Institutionalize a Quantitatively Managed Process
GP 4.1 – Establish quantitative objectives for the process
GP 4.2 - Stabilize subprocess performance
GG 5 – Institutionalize an Optimizing Process
GP 5.1 – Ensure continuous process improvement
GP 5.2 - Correct Root causes of problems
Table 2 – Alternative approaches sorted by Declaration of Interdependence

|--------------------------------------|-------------------------------------------------|----------------------------------------------------------|
| We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference. | Individuals and interactions over processes and tools | GP 2.4 - Assign responsibility and authority  
GP 2.5 – Train people |
| We boost performance through group accountability for results and shared responsibility for team effectiveness. | Individuals and interactions over processes and tools | GG 2 – Institutionalize a Managed Process  
GP 2.1 - Establish an organizational policy  
GP 2.3 – Provide adequate resources  
GP 2.4 - Assign responsibility and authority  
GP 2.5 – Train people  
GP 2.10 - Review status with higher level management |
| We increase return on investment by making continuous flow of value our focus. | Working software over comprehensive documentation | GP 2.2 – Plan the process  
GP 2.6 – Manage configurations  
GP 2.9 - Objectively evaluate adherence |
| We deliver reliable results by engaging customers in frequent interactions and shared ownership. | Customer collaboration over contract negotiation | GG 1 – Achieve the Specific Goals  
GP 1.1 – Perform base practices  
GP 2.7 - Identify and involve the relevant stakeholders |
| We expect uncertainty and manage for it through iterations, anticipation, and adaptation | Responding to change over following a plan | GP 2.8 - Monitor and control the process  
GP 3.2 – Collect Improvement Information |
| We improve effectiveness and reliability through situationally specific strategies, processes and practices." | Responding to change over following a plan | GG 3 – Institutionalize a Defined Process  
GP 3.1 – Establish and maintain the description of a defined process |
Agile & the Declaration of Interdependence: A new approach to Process Improvement
www.davidconsultinggroup.com

1 Manifesto for Agile Software Development, ©2001, Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

2 Declaration of Interdependence, ©2005 David Anderson, Sanjiv Augustine, Christopher Avery, Alistair Cockburn, Mike Cohn, Doug DeCarlo, Donna Fitzgerald, Jim Highsmith, Ole Jepsen, Lowell Lindstrom, Todd Little, Kent McDonald, Pollyanna Pixton, Preston Smith and Robert Wysocki.

3 Schwaber - Agile Project Management with Scrum, Microsoft Press, 2004

4 Schwaber, Beedle - Agile Software Development with Scrum, Prentice Hall, 2002

5 Chrissis, Konrad, Shrum – CMMI Guidelines for Process Integration and Product Improvement – CMU/SEI – Addison-Wesley, 2004