

The Impact of Agile and Lean on Process Improvement

Richard Turner, Stevens Institute of Technology

Abstract. Since the early process improvement work in the '70s and '80s, our understanding of process in software and systems engineering has changed significantly. The Agile and Lean movements have made us think differently, and our processes have changed. Have our approaches to process improvement (PI) changed as well? This article discusses how Agile and Lean concepts inform process improvement approaches to address those changes.

Twenty-five years after the first version of the Software Capability Maturity Model and the original SPICE standards, and more than a century since Frederick Taylor introduced the concept of scientific management, there have been significant changes in the way we view process in software and systems engineering. Many of these are responses to a changing engineering environment that includes more uncertainty, continuously evolving systems of systems, and rapidly changing needs and technologies.

The result has been a parade of diverse and often contradictory concepts and approaches to management, development and sustainment. INCOSE's Systems Engineering Capability Model, [1] CMMISM, [2] and numerous ISO/IEC process standards have co-evolved with the introduction of adaptive development concepts such as the Agile Manifesto, [3] Lean principles applied to knowledge work, [4][5] model-based engineering, [6] and value-based system and software engineering, [7] Kanban, [8] Lean Startup, [9] SAFE, [10] DevOps, [11] and the Incremental Commitment Spiral Model. [12] To add additional complexity, many of the parade's participants have mixed up, split off and re-formed into myriad hybrids along the way. There have even been fundamental changes to the revered Project Management Institute's (PMI) Guide to the PM Body of Knowledge (PMBOK). [13]

As definitions and approaches have evolved, how has this parade of ideas — each with one or more corporate, academic or consultative promoters — changed the way we approach improving engineering processes in the real world? This article discusses a number of factors observed along the parade route.

Starting with a Consensus

At the 2015 International Conference on Software and System Process, I was privileged to moderate a panel that discussed whether the way we have developed systems has significantly changed our pursuit of improved processes. Lars-Ola Damm (Ericsson), Philipp Diebold (Fraunhofer IESE), Anton Keks (Codeborne), Rory O'Connor (Lero), Lee Osterweil (University of Massachusetts) and I represented various technical and governance arenas where these ideas have played both with and against each other.

The overall consensus of the panel was that the evolution of the system development environment and the way systems are developed has had a significant impact on the mechanics of process engineering and improvement. At the same time, it has strengthened the influence of a number of fundamental principles. Together, the approaches incorporate planned experimentation and rapid adaptation, increase focus on outcomes and stakeholders, and move away from organizational control and conformance.

We could easily stop there, thank the panel for a job well done, and move to another topic. However, there are a number of interesting ideas that should be addressed, generally supporting but also illuminating the panel's concise summary. For example, what do we see as the important differences between our traditional concepts of PI and these recent approaches from the Lean and Agile community?

Process Improvement Home Grounds

In our 2003 book "Balancing Agility and Discipline," [14] Barry Boehm and I identified a set of "home grounds," or characteristics associated with what the fifth edition of the PMI Guide now refers to as "adaptive" and "predictive" project management environments. At that time, there was general antipathy (to the level of religious fervor) between the proponents of traditional development processes (so-called disciplined) and proponents of newer, lighter and more adaptive (agile) development processes. In the intervening years since that publication, there has thankfully been significant rapprochement between these factions.

I believe recasting those characteristics can serve as a lens through which to consider the evolution of process improvement. Table 1 illustrates the original software development home grounds we identified.

Although this was created as a software development spectrum, it is possible to inspect the application of PI in terms of these characteristics. Table 2 presents a personal interpretation of these home grounds in terms of the process improvement activity.

Now, let's look at each of these modified characteristics more closely.

Application

This characteristic covers the type of environment where the approach is applied as well as the general goals and activities.

Much of traditional PI has been internally focused on meeting specific standards (CMM/CMMI level, ISO Certification) based on a set of goals or recommended practices that have little to do with the end product's applicability or capability. Often these are broadly stated as critical success factors and then broken down to examples of specific practices or activities. If these practices (or other equivalent practices) are included in the process and are performed, then the improvement is assumed to be accomplished. Measurements of the impact are often included in the overall management metrics and are usually evaluated within the organizational context.

Traditional PI has often depended on process expertise and best practice to codify standard processes for all personnel. This is primarily directed toward the establishment of common and certifiable practices. However, the concept of organizational standard processes may be losing its relevance in a development environment with so much uncertainty. When there is so much change

happening, having everyone use the exact same process can often result in the “Bed of Procrustes” effect where the process requires the performers to radically redefine the project, often to the detriment of the customer. [12]

There are circumstances, however, where the stability and commonality of processes are critical to the performance and certifiability of products. Safety and security standards are the poster children for having broadly accepted and conforming processes. These are the cases where PI should be carefully orchestrated and systemically controlled.

Adaptive methods echoing Agile and Lean values apply a more individualized experimental approach, whereas those performing the work are always looking for improvement opportunities. They can apply changes for improvement within every cycle, tracking the results to determine success or failure. This echoes the expressed goal of a level 5 CMMI organization. Figure 1 shows one combination of the two approaches introduced by Vic Basili under the name “Quality Improvement Paradigm.” [15] This approach grew out of 25 years of PI work at a NASA facility, where a team developed models specifically around the process needs of the organization and empirically evaluated the models via experiments.

Governance

This characteristic describes how the approach is managed, resourced, and measured.

Although often initiated from the bottom up as a response to management failure, process improvement was created as a management-driven activity. Traditionally there has been a large infrastructure associated with process improvement with special groups of process experts as well as assessors and QA people that enforce standard processes. The responsibility being placed in an organization (such as an SEPG) that lies outside the performing organization does not necessarily establish ownership of the improvement with the primary performers.

This model has worked relatively well in large organizations where there was a desire for standard practices and the ability to share resources between projects without process-related learning curves. While these large organizations generally have a higher probability of sufficient overhead funding, it also led to larger PI activities with relatively long time scales for improvement evaluation.

The adaptive nature of contemporary work makes this a very inefficient way of approaching PI. Technology has incorporated and often now enforces a certain level of process (as in model-based engineering, repository-based CM tools or required software development kits). Development is seen more as brownfield evolution rather than greenfield creation, so actual development activities have realigned in importance. Concurrent engineering requires continual adjustments among hardware, software and operational concept as the system evolves. Newer approaches to PI tend to fit better in these environments, primarily because they are more highly integrated into the work. They require consistent and supportive communication among management, developers and customers.

A traditional, organizational-centric model can considerably raise the cost of PI in two ways. First, the expense of maintaining an independent or matrixed process group and a set of standard processes is not negligible. Second, whenever there is

Characteristics	Agile	Plan-driven
Application		
Primary Goals	Rapid value; responding to change	Predictability, stability, high assurance
Size	Smaller teams and projects	Larger teams and projects
Environment	Turbulent; high change; project-focused	Stable; low-change; project/org. focused
Management		
Customer Relations	Dedicated on-site customers, where feasible; focused on prioritized increments	As-needed customer interactions; focused on contract provisions; increasingly evolutionary
Planning/Control	Internalized plans; qualitative control	Documented plans, quantitative control
Communications	Tacit interpersonal knowledge	Explicit documented knowledge
Technical		
Requirements	Prioritized informal stories and test cases; undergoing unforeseeable change	Formalized project, capability, interface, quality, foreseeable evolution requirements
Development	Simple design; short increments; refactoring assumed inexpensive	Architect for parallel development; longer increments; refactoring assumed expensive
Test	Executable test cases define reqts.	Documented test plans and procedures
Personnel		
Customers	Dedicated, collocated CRACK* performers	CRACK* performers, not always collocated
Culture	Comfort and empowerment via many degrees of freedom (thriving on chaos)	Comfort and empowerment via framework of policies and procedures (thriving on order)

* Collaborative, Representative, Authorized, Committed, Knowledgeable

Table 1. Home Grounds from Boehm and Turner [14]

Characteristic	Predictive (Traditional)	Adaptive
Application	Certification or appraisal, internally focused; large, multiple team organizations with standard processes; good for highly critical processes	Rapid improvement, customer satisfaction, externally focused; individual teams; neutral to criticality
Governance	Organizational responsibility, process owners are process team (SEPG) and the organization	Individual responsibility, rapid evaluation of impact, process owners are the actors
Values	Repeatability, uniformity, satisfaction of requirements, specific process areas of activities	Customer satisfaction, quality of product, entire lifecycle
Personnel	Organizational team drives solutions, benefits often accrue mostly to marketing or management; standard practice drives knowledge	All personnel are involved with solutions and benefits; cross-fertilization of teams spreads knowledge

Table 2. Interpretation of Home Grounds for the Process Improvement Spectrum

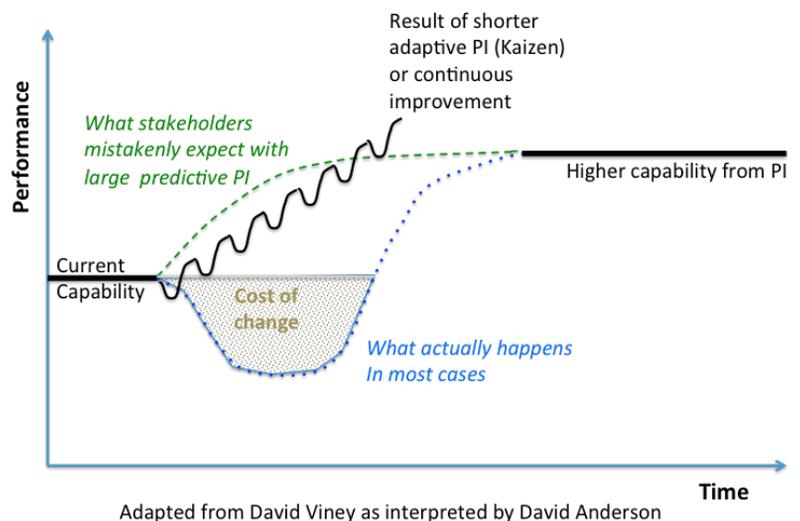


Figure 1. The Quality Improvement Paradigm [illustration from 17]

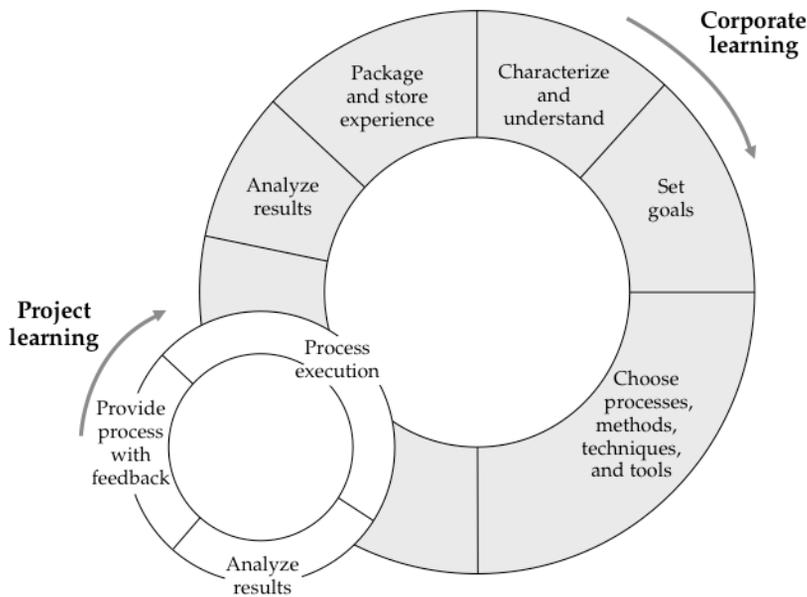


Figure 2. The J Curve (Illustration based on [16])

a “large” process improvement project, there is an unavoidable churn across organizations as the performers change and adopt the processes. This effect, often referred to as the “J curve,” effectively reduces productivity for a period of time relative to the length of the process until the change is complete and the organization experiences the promised benefit. Figure 2 illustrates how the curve impacts improvement, and it shows the difference in effectiveness of shorter, continual improvement approaches.

Adopting the idea of more tailored, individually driven process improvement has been difficult, and it is particularly uncomfortable in a command and control management structure. However, it fits well into the more collaborative management approaches emerging in many companies. Rather than having additional organizational overhead, adaptive approaches are more likely to use coaching as opposed to traditional project management to help individual teams instill improvement as a part of normal work.

Values

One of the drawbacks of earlier process improvement approaches was the concept and distribution of value. The overall value of the process improvement initiative was often situational at best and nebulous at worst. Where it was seen as a necessity for competitive credibility, the value was in passing the audit rather than in any value to the organization and the customer. In other cases, the value was essentially associated with the success of one or two champions and disappeared if they failed, changed positions or left the company. On those occasions where PI was primarily instituted for the actual improvement of the organization, the internal focus on practices was often valued as a way of cutting costs, standardizing work, or deploying better predictive management capabilities rather than improving the product or raising customer satisfaction.

With the emergence of Agile and Lean, the concept of value became more aligned with outcomes. The focus on value stream and value-based decision making and scheduling brought ad-

ditional considerations to what were once considered best practices. In Lean Startup and its laser focus on the market, value is intensely associated with buyers and their desires, known or unknown. Process improvement that does not improve the ability to adapt has little value.

The values of PI in adaptive environments are both holistic and individualistic. Agile and Lean focus on the ability to satisfy the customer through value delivered for cost and product suitability, and the ability to provide the individual with resources to own and benefit from improvement. Traditional PI values standardization and specifications and organizes according to key technical areas rather than the overall value chain.

Personnel

This characteristic covers a good deal of ground but essentially looks at the preferences of the people involved in the process improvement.

Traditional PI focused more on the process than on the people performing the process in a highly sterile and rigid atmosphere. Often awards went to the team leading the PI project rather than to those who suffered through the transition. The concept of having a process expert that did not really understand you and your work, but was trying to squeeze you into a predefined role or task, caused more stress than was probably necessary.

Much of the fervor behind the Agile Manifesto came from the recognition that creative people who are doing knowledge-based work are not plug-and-play resources. Even with standard processes, they operate according to soul more than to programmed instructions. People vary along the same types of spectrums as their projects, environments, and PI approaches. Thus, understanding the people following the process is key.

Lean has shown that we have outgrown Taylor’s view that workers are too stupid to understand the “science” surrounding their tasks. In an age of automation and technical service industries, this point of view is rarely applicable. And, given the number of books on empowerment, coaching, collaboration and project-less workflow, it is clear that management theory is catching up.

In adaptive organizations, the team performing the work is responsible for their own processes. These team members are individually involved in both the PI activities and the derived benefits. These organizations rely on cross-fertilization of personnel across multiple projects to organically improve the organization as a whole.

A Way Forward from Looking Backward

So what is the purpose of process improvement? Returning to the consensus of the panel, process improvement techniques have certainly changed to adapt to the new realities of system development. Both approaches to process improvement follow some form of the “plan—do—check—act” or “observe—orient—decide—act” cycles for identifying barriers and enablers to improvement.

Differences in process improvement approaches seem more common in the governance and value characteristics. They echo the general changes in the development process from the greenfield, highly defined projects in the ‘70s and ‘80s to the brownfield, uncertain, rapidly evolving projects of today. Similarly, there have been more hybrid development life cycles and models to fit specific developmental and environmental needs, so

combinations of predictive and adaptive process improvement implementations have emerged.

Fundamentally, all processes and the approaches implemented to improve them should be engineered to be as amenable to change as the environment requires. Hybrid approaches are a principal means of assuring this, and their structure and content fall naturally out of a review of risks associated with the holistic environment. Ways to balance approaches using risk is described in detail in [14] and in the Meta-principle of Risk Balancing in [12].

Finally, process improvement fundamentals, generally derived from change management fundamentals, remain valid. Implementing them to PI approaches is a more difficult challenge. Here is a list of critical success factors, drawn from experience with matching PI approaches to needs in software, systems, and systems of systems evolution:

- Improve for the benefit of the business, organization, and personnel, not some externally mandated target.
- Clearly identify and exemplify the desired PI values, both internal and external, and use them to determine your approach.
- Fit the approach to the environment.
- Follow the pain in prioritizing what is to be improved.
- Understand the current capability; set achievable, measurable and meaningful goals; track progress.
- Experiment and deploy incrementally; fail fast and safely; reduce the improvement cycle time.
- Utilize reflection techniques to provide “double-loop learning”: find an error, correct it, and then try to understand how it happened to prevent it in the future.
- Involve and empower the people who do the work to adapt and improve their own processes.

No list of one-liners can replace understanding, and process improvement is critically dependent on understanding the environment, the organization's values and needs, and most critically, the people. Improvement is necessary, so identifying the collection of practices that holistically best suits the improvement target worthwhile.

In the *CMMI Survival Guide*, [16] a book much more about process improvement than about CMMI, Suzanne Garcia Miller and I used the metaphor of a journey to describe our philosophy for process improvement. I think it remains both accurate and deep. My favorite epigraph included in the book sums it up nicely:

*You must travel a long and difficult road –
a road fraught with peril, uh-huh,
and pregnant with adventure.
You shall see things wonderful to tell. . . .
I cannot say how long this road shall be.
But fear not the obstacles in your path,
for Fate has vouchsafed your reward.
And though the road may wind, a
nd yea, your hearts grow weary,
still shall ye follow the way,
even unto your salvation.*

– An old blind man on a flatcar in
“O Brother, Where Art Thou?”,
a film by Ethan and Joel Coen.

REFERENCES

1. INCOSE, “A Systems Engineering Capability Model.” Vol 1. (June 1996.)
2. Chrissis, Mary Beth; Conrad, Mike & Sandy Shrum. (2011.) “CMMI for Development v1.3” Addison-Wesley, Boston.
3. Beck, K., et al. “Manifesto for Agile Software Development.” <http://agilemanifesto.org>.
4. Reinertsen, Donald G. (1997.) “Managing the Design Factory: A Product Developer’s Toolkit” The Free Press, New York.
5. Shalloway, Alan; Beaver, Guy & Trott, James R. (2009.) “Lean-Agile Software Development: Achieving Enterprise Agility.” (1st ed.). Addison-Wesley Professional, Boston.
6. Friedenthal, S., Moore, A. & Steiner, R. (2011.) “A Practical Guide to SysML: The Systems Modeling Language.” Elsevier.
7. Biffl, Stefan et al. (2006.) Ed., “Value-based Software Engineering.” Springer, Berlin.
8. Anderson, David. (2010.) “Kanban: Successful Evolutionary Change for Your Technology Business.” Blue Hole Press, Sequim, Wash.
9. Ries, Eric. (2011.) “The Lean Startup.” Crown Publishers, Danvers, Mass.
10. Leffingwell, D. et al. “Scaled Agile Framework (SAFe) 4.0.” <http://www.scaledagileframework.com>.
11. Kim, Gene et al. (2016.) “The DevOps Handbook.” IT Revolution Press, Portland, Ore.
12. Boehm, B.; Koolmanojwong, S.; Lane, J. & Turner, R. (2014.) “The Incremental Commitment Model: Principles and Practices for Successful Systems and Software.” Addison-Wesley, Boston.
13. Project Management Institute and the IEEE Computer Society. (2013.) “Software Extension to the Program Management Body of Knowledge Guide Fifth Edition.” Project Management Institute Incorporated. Newton Square, Penn.
14. Boehm, Barry & Turner, Richard. (2003.) “Balancing Agility and Discipline: A Guide For The Perplexed.” Addison-Wesley, Boston, Mass.
15. Basili, V. (Sept. 1993.) “The Experience Factory and its Relationship to Other Improvement Paradigms.” In “Proceedings of the Fourth European Software Engineering Conference (ESEC) in Garmish-Partenkirchen, Germany.” The Proceedings appeared as lecture notes in “Computer Science,” Sept. 1993.
16. Anderson, David. (March 29, 2016.) “Organizational maturity & the J-Curve Effect.” Blog post, <http://anderson.leankanban.com/organizational-maturity-the-j-curve-effect>.
17. Miller, S. G. & Turner, R. (2007.) “CMMI Survival Guide: Just Enough Process Improvement.” Addison-Wesley, Boston, Mass.

ABOUT THE AUTHOR



Dr. Richard Turner has 40 years of experience in systems, software and acquisition engineering in both the private and public sectors. Currently a Distinguished Service Professor at the Stevens Institute of Technology, Turner is active in the Agile and Lean engineering communities and was a core author of the IEEE-CS/PMI Software Extension for the Guide to the PMBOK. His current research uses simulation to investigate the effectiveness of Agile, Lean, service and complexity concepts in managing SoS evolution. He is a Golden Core awardee of the IEEE-CS, a senior member of the IEEE, and co-author of four books.