Using Agile Project Management to Maximize Your and Your Coauthors' Productivity

Dr. Ted Eschenbach P.E., University of Alaska Anchorage

Dr. Ted Eschenbach, P.E. is the principal of TGE Consulting, an emeritus professor of engineering management at the University of Alaska Anchorage, and the founding editor emeritus of the Engineering Management Journal. He is the author or coauthor of over 250 publications and presentations, including 19 books. With his coauthors he has won best paper awards at ASEE, ASEM, ASCE, & IIE conferences, and the 2009 Grant award for the best article in The Engineering Economist. He earned his B.S. from Purdue in 1971, his doctorate in industrial engineering from Stanford University in 1975, and his masters in civil engineering from UAA in 1999.

Dr. Neal Lewis, University of Bridgeport

Neal Lewis received his Ph.D. in engineering management in 2004 and B.S. in chemical engineering in 1974 from the University of Missouri – Rolla (now the Missouri University of Science and Technology), and his MBA in 2000 from the University of New Haven. He is an associate professor in the School of Engineering at the University of Bridgeport. He has over 25 years of industrial experience, having worked at Procter & Gamble and Bayer. Prior to UB, he has taught at UMR, UNH, and Marshall University. Neal is a member of ASEE, ASEM, and IIE.

Dr. Gillian M. Nicholls, Southeast Missouri State University

Dr. Gillian M. Nicholls is an Assistant Professor of Quantitative Methods at Southeast Missouri State University. Her research interests are in applying statistical analysis and optimization to supply chain management, transportation management, and engineering education. She holds the B.S. in Industrial Engineering (Lehigh University), Masters in Business Administration (Penn State University), M.S. in Industrial Engineering (University of Pittsburgh.), and Ph.D. in Industrial Engineering (University of Pittsburgh). Prior to entering academia, Dr. Nicholls was a practicing industrial engineer in the freight transportation industry. Address: Donald L. Harrison College of Business, Southeast Missouri State University, One University Plaza – MS 5815, Cape Girardeau, MO 63701; telephone (+1) 573.651.2016; fax: (+1) 573.651.2992; e-mail: gnicholls@semo.edu.

Dr. William J. Schell IV P.E., Montana State University

Dr. William J. Schell holds a Ph.D. in Industrial and Systems Engineering – Engineering Management from the University of Alabama in Huntsville and M.S. and B.S. degrees in Industrial and Management Engineering (IME) from Montana State University (MSU). He is an Assistant Professor in IME at MSU with research interests in engineering education and the role of leadership and culture in process improvement. Prior to his academic career, he spent 14 years in industry where he held leadership positions focused on process improvement and organizational development.

Using Agile Project Management to Maximize
You and Your Coauthors' Productivity

Abstract

For decades as information technology (IT) projects grew bigger and more complex, project failures seemed to become increasingly common, in spite of intense efforts to apply traditional project planning. Those traditional planning tools focused on balancing the triple constraints of cost, schedule, and scope to create a plan. Then those tools unsuccessfully focused on delivering the planned scope within the planned cost and schedule. In 2001 the "agile project manifesto" pointed the way to better manage projects having a flexible scope in an uncertain environment. Since then agile project management in IT has matured and proven itself for large and small IT projects.

Academic work has features that parallel the reasons agile project management is needed for IT. It often has (1) an undefined scope, (2) unknown and possibly unmeasurable task times, (3) an unidentified assortment of tasks featuring undiscovered task dependencies, and (4) an ever-changing resource availability for each project due to the impacts of other projects on resource needs. These factors acting in combination create a "perfect storm" that destroys the ability to use traditional project planning. A simplified form of agile project management can be built around the flexible management of scope, prioritization of projects and tasks, and creation/management of deadlines that depict much of academic work. This paper details lessons that have been learned, supporting immediate application of this approach.

Introduction

Many practicing engineers and academics from a wide array of engineering disciplines have acted as project managers. In most cases, project management (PM) has generally been learned through taking formal coursework, training classes, and/or on the job experience. Many academics have gained additional experience in PM by teaching the subject. From one of the author's experience as an ABET evaluator, it appears that traditional project management is almost universally taught, often with little if any discussion of agile PM. Thus for most academics not working in IT, the paradigm of traditional project management is what they have learned and practiced.

Traditional Project Management
Through education and experience, many engineering academics are familiar with concepts from traditional project management. For those who are less familiar, we offer a brief primer here. The Project Management Institute (PMI) defines a project as a temporary endeavor undertaken to create a unique product, service, or result.[1] While projects are as ancient as human endeavors, formal project management has its origins in the production changes needed as the United States entered World War II and the emerging complexities of Department of Defense (DoD) needs as the Cold War began.[2] To successfully meet these complex needs, the government wanted a single point of contact – the project manager – to manage the wide array of tasks required to deliver the desired final product.

Traditional PM involves great effort to delineate the project's scope while scrupulously limiting "scope creep" without approval of documented user change requests that preferably are small in nature. Project tasks are identified and scheduled in a detailed timeline. The project schedule is meticulously executed to achieve a successful completion and delivery of the project's outputs on time and within budget. The premise that undergirds traditional PM is that rigorous planning at the project's start is a crucial element to successful completion. The natural assumption that follows this premise is that better planning of the scope, tasks, and schedule leads to an increased likelihood of project success. This was true for most projects faced at the beginning of the project management era, and it remains true in many cases.

As projects continued to grow in size and complexity, traditional PM continued to mature and now includes a wide range of tools to successfully manage all three aspects of the project management golden triangle shown in Figure 1: scope, schedule, and cost.
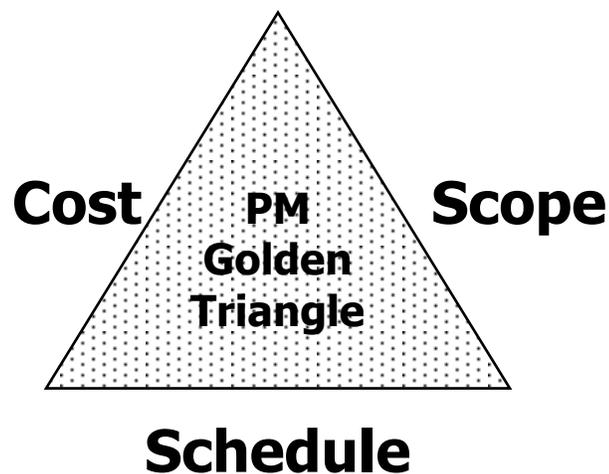
Figure 1. The Project Management Golden Triangle.

The effectiveness of these tools to deliver the desired product within the needed constraints rests largely on how well defined each side of the triangle is at project initiation. When the project is focused on execution, such as a construction project, or launching a new product, improved planning has led to better project execution in a vast number of cases. When great levels of uncertainty exist, traditional project management has proven to be less effective than desired. Notably, traditional approaches led to a large number of failures in the area of software development, causing the information technology industry to question the methods.

A key difference between software development projects and construction projects is the nature of the project scope. In construction (or product launch), the scope of the project is given and fixed, and the project is managed to minimize 'scope creep.' In software development projects, scope is often fuzzy and subject to change as the project progresses. When project scope is undetermined and perhaps indeterminable, traditional project management approaches have significant problems.

Project Management in Academic Work
Many of the traditional PM theories and tools remain relevant for academic work. The tripartite constraints of the scope, schedule, and cost objectives are integral in academia such that a change to any of them affects at least one and possibly both of the other constraints. New engineering educators are especially familiar with these trade-offs from their time as students. Since the schedule constraint on class projects rarely if ever changed, students regularly sacrifice scope (quality) or time (sleep) to meet deadlines. Cost constraints may also be relaxed if purchasing additional reference materials or other support will help in successful course completion.

For faculty, work in academia tends to be a collection of projects. Whether the work is teaching, performing research, writing grants, journal articles, or conference papers, preparing presentations, or performing service for the university, academic work tends to have a flexible scope. Other constraints tend to be more fixed, including schedules and cost (often measured as time available). Considering these factors, academic work tends to have more in common with IT projects, where agile PM is commonly used, than with traditional projects.

Thus a great similarity exists between projects in IT and academia: scope tends to change to meet a deadline. The largest difference is the driver for scope changes. In IT, the driver is primarily about evolving knowledge about functionalities—how difficult they are to deliver versus how valuable they are. In academia, the driver is primarily shifting resource availability (time) for each project due to the continual changes in the number and priority of projects.

History and Methods of Agile Project Management
Agile project management was born from a desire to find more effective methods in highly uncertain environments. In 2001 a group of IT professionals gathered to discuss problems with IT PM and created the Agile Software Development Alliance (ASDA).[3] This gathering also featured the crafting of an improved method for developing software. Before this gathering, software development was generally directed by the traditional PM approach including highly structured processes, rigid procedures for scope determination, detailed PM schedules, extensive documentation, and high reliance on contractual language to specify software deliverables.

As individuals, some of the members of the ASDA group (a.k.a. the Agile Alliance) had already started using a more flexible approach so they could better accommodate changes in customer requirements and scheduling. The informal gathering of IT professionals discussed the need for greater flexibility in software development and codified it into twelve principles based upon a set of four core values:

- Focus on interactions with individuals rather than following a rigid set of processes and tools.
- Meet customers' requirements by delivering functioning software without the delays caused by providing extensive documentation.
- Cultivate a spirit of collaboration with the customer instead of hard bargaining in contract negotiations.
- Provide flexibility to adapt to customers' requests for changes as opposed to rigidly retaining the pre-defined scope. Be able to embrace scope creep as responding to a customer's need rather than fighting it.

Together the core values and principles are referred to as the "Agile Manifesto."[3]

There are three philosophical beliefs underpinning the agile principles.[4] These are, 1) Visibility, in that project outcomes must be visible to the people controlling the process; 2) Adaptation, such that scope changes must be recognized and responded to quickly in order to minimize disruptions; and 3) Inspection, in which the team members responsible for quality control must do so regularly and be able to detect issues that violate the specifications.

The major approaches to agile PM are Extreme Programming (XP), Lean, and Scrum. All of these assume that collocated team members work with close integration and frequent face-to-face contact.[5] Another hallmark of these approaches is the rule that "team members work on only one project."[6]

Extreme Programming (XP) came into use in 1996 with customer satisfaction as the major objective for software projects.[7] The goal under XP is getting initial software to the customer within weeks so that rapid feedback is obtained and the project team develops a sense of momentum through an "early win."[8] Iterative testing is conducted with each new software product rather than waiting for the conclusion of the product development phase to test each aspect.[9]

Lean originated in manufacturing as a means of minimizing all forms of waste during production. The lean approach in PM centers around designing quality into the product, obtaining wide involvement in the project, resolving root causes for problems, driving out waste, working towards rapid delivery, and striving for continuous improvement.

Of all the agile PM approaches, Scrum is most frequently utilized.[10] Scrum features a series of iterations in activity referred to as "sprints." A sprint starts with the development of a software product module and continues with testing and revision until the module meets acceptance criteria. Each sprint ends with the completion of a product deliverable. The team then moves on to the next sprint until finally the software product is completed and released.

In all of these cases, flexibility is required in order to meet ever changing scope requirements. Since the project scope changes, the traditional project management concept that 'better planning leads to better outcomes' is invalidated.

Key Differences Between Traditional PM and Agile
Agile is not just a different set of PM procedures. Instead it is a shift in how to think about conducting PM. Part of agile is considering projects in small modules to allow the team to prioritize the modules that can be done quickly and are of the highest priority[11]. Rather than mapping out the specifications at the start, agile PM uses a constant series of small steps throughout the project's life cycle.

By comparison, traditional PM builds a project schedule using a defined scope, estimated task times, a network of task dependencies, and estimated availability of resources. Decades of world-wide experience has resulted in developing tools to manage risks, control costs, integrate scope changes, etc. We assert that the underlying assumption of traditional project management is that *better planning of the complete project will lead to better outcomes*. We further assert that agile

approaches to PM are needed when this underlying assumption cannot be satisfied. At least four situations exist where the underlying assumption does not apply, and an agile approach may be superior: [12]

- *Unclear scope.* When clients, customers, owners, or teams are unsure of what features, contents, or capabilities are required or most valuable.
- *Unknown or unknowable task times.* In many research and writing projects, the amount of time required to solve a problem is often unknown and unknowable.
- *Unknown number and set of tasks implies unknown task dependencies.* Research and writing team members almost by definition do not know which possible solution (or experiment) will work. The tasks for one approach may not be identified until the failure of the previous attempt.
- *Unknown or varying availability of resources.* We believe the most common cause of unknown resource availability for a particular academic project is a multi-project environment for all or most team members. Changing time availability due to new or higher priority projects can dramatically reduce the time that was initially allocated to a project.

Academics are regularly faced with all of the above situations in all areas of the tripartite mission of teaching, research, and service. They encounter high levels of uncertainty when facing a research project, deciding what to include in a given lecture, writing the next journal article, or working on a service project. Our own time availability may be the greatest constraint, which forces changes in project scope. In these cases, agile approaches may offer the best solution.

Extending Agile to Academics

While agile practices began in area of software development, all of the principles are applicable to other types of projects.[13, 14] Gonzalez[15] conducted research in the application of agile PM principles to the creation and management of intellectual property. Another application is agile product development.[14, 16] An incremental approach to tasks keeps the project moving forward, enabling fast, frequent feedback and avoiding wasting effort by 'going down blind alleys'. The rapid iterations of incremental progress and scope adjustment constitute regular "wins" that build momentum and morale. Agile methods have also been applied to student teams.[17] The literature shows agile has proven its value over a wide range of applications. We now examine how this team has adapted key agile practices to promote greater productivity in an academic role.

In most academic appointments, new engineering educators are faced with a complex balance between teaching, research, and service expectations. At a macro level, these needs are analogous to the competing priorities evident in the golden triangle of traditional project management, where an imbalance in the triangle can lead to failure of the overall project (progress toward tenure). While there are components of the role that have a well-defined scope (e.g. one's teaching load this semester) there are many more components where the scope may experience material changes that are difficult to predict. While an agile approach will not resolve these changes, it does promote a greater level of flexibility and ability to respond to changing demands. This improved ability to respond to changing circumstances can materially improve the effectiveness of an educator's work in each of the areas.

Agile in Teaching
The initial preparation of a course for delivery by a new engineering educator with traditional lecture format fits to a degree with traditional PM. There is usually an existing course description that delineates the expected content, the schedule of course meeting times is set, the textbook and other resources may already be set, and the body of knowledge to be taught has a known set of desired learning outcomes. These combine to allow a new engineering educator to prepare the syllabus and course schedule outline in advance much as traditional PM would set out the scope and project schedule. The various lectures are generally not written out that far in advance, and represent some of the tasks to be accomplished in the project.

Agile PM certainly applies when prepping lectures prior to delivery. The scope of how detailed the lecture will be, what example problems are prepared in advance, and whether class activities are used may depend on the time available just before the lecture. The content delivery can be adjusted during a lecture as student questions lead to scope changes, discussion leads to a sudden epiphany of a good teaching example, or if additional time is available to expand upon what is being discussed. If discussion sparks an idea for a different way of communicating the concepts, an extra example problem or exercise may be developed and administered in the next class or posted to a course management site for student self-study.

Agile PM also applies when unexpected problems arise such as when lecture delivery doesn't go well and student confusion needs to be addressed. If illness, weather emergencies, and other unanticipated events lead to a lecture being cancelled or delayed, an agile approach may be applied to recover. Lower priority content may be dropped. Additional online methods may be used to deliver content in place of the lost lecture time. Using a "flipped" or more active learning methodology in teaching generally creates a greater level of scope uncertainty and can lead to more agile planning of smaller modules.

Agile in Research
In the experience of the authors, perhaps the greatest benefit from an agile approach can be found in the area of research, specifically with writing efforts to obtain research funding and publications. While there are many writing projects where a real external deadline exists (e.g. conference paper and grant submissions) there are at least as many without this external pressure. For a new faculty member accustomed to external deadlines from courses, funding agencies, or other university requirements, these projects without an external deadline present a risk to future success since they may not receive the priority they warrant. By embracing an agile approach, similar to the Scrum methodology described earlier, writing becomes a series of mini-projects (sprints) with internal deadlines created by the team. Since the goal of each of these sprints is to develop a usable product (such as an improved or more complete draft, not a perfect one) the deadlines can be short, the temptation to agonize over small details is reduced, and forward progress can be rapid as the project taps into the collective brain of the team.

Our team has found this Scrum approach to writing and research projects to be very beneficial in terms of both productivity and output quality. Typically, this approach uses the following steps:
1. The team develops a schedule of regular meetings. These meetings serve as sprint review meetings and set the standard duration for each sprint.

2. The next need in the project is identified; this defines the scope for the next sprint. Sometimes this need is larger than will fit into the time available before the next meeting, in which case the sprint may be extended to multiple meetings.
3. Based on availability and interest, a member of the team volunteers to lead the sprint.
4. The team comes together at the next sprint meeting to review the new product, discuss open questions and define the needs of the next sprint.
5. The process repeats until the overarching requirements are successfully completed.

For example, this paper began as an extension of an article written by three of the authors when one of them was with the fourth author at a conference and asked how the topic might be extended for the benefit of the ASEE audience. A short while later all authors were together at dinner, and this project was on its way. Over the course of the next 36 hours, the team generated six versions of the abstract (while catching a variety of flights back to their homes) and agreed to a finalized version for submission.

One gap with this approach is the necessity of a writing team. How does the new engineering educator who has not yet built a network of collaborators and co-authors gain similar benefits without the team? The use of writing groups is an approach utilized by the junior member of our team that provides many of the same benefits for projects that are less team based. Specifically, this author became a member of a weekly writing group at Montana State University based on the model developed by Tara Gray.[18] In this model, a small group of interdisciplinary scholars meets weekly to edit the work of the team. While this model does not gain the advantage of having another member of the team move the project forward rapidly, it does provide regular deadlines and exceedingly helpful input regarding the clarity of the writing already produced.

Agile in Service
The service expectations of new engineering educators may be very specific or very unclear. How these expectations may be met is likely to be highly variable. The agile principles of simplicity and welcoming changing requirements call for an approach that sets the priority on aspects of service that will enhance other goals. For example, it's helpful to prioritize service opportunities with respect to national service for research visibility vs. university service for building a local reputation with new colleagues. Finding a service opportunity that fits available time and advances academic goals is an application of agile principles. The scope of the service project may be unclear and thus allow for struturing it to fit the available team resources and time. Each service commitment may be treated as a flexible task within the larger project of providing academic service. Larger service commitments may instead resemble separate projects to be juggled in an agile, multi-project environment.

Once a service opportunity is selected, agility can be applied in how it is conducted. One example of applying agility to a service opportunity arose from participation in a university commencement volunteer committee. A suggestion was made to consider offering bundles of flowers available for sale to the students or their family members. Committee members brainstormed pros and cons of the idea via email over the course of two days. Potential problems were weighed such as complicating the process of leading students up to the stage and back to their seats while carrying flowers or leaving them behind. The risk of triggering allergy issues among attendees was discussed. The disposition of potential funds raised from the exercise was

debated. A plan was put together and a subset of the committee was assigned to contact vendors and develop a proposal for further consideration. Overall, rather than a formal process to analyze the suggestion, develop a plan, and execute it, the informal asynchronous communication and assignment of tasks to a subset of the committee worked much faster.

Overcoming Structural Limitations to Agile Work
As discussed in the overview of agile, a key tenet of most agile approaches is that team members must spend a great deal of time working together on a problem in a face-to-face environment. For many academics their best potential collaborators are not located at their university. Our team works through the same issue with four people in four different time zones. To date, the team has used a combination of asynchonous communication (via email), face-to-face meetings (at project inception), and occasional phone calls between members of the team to discuss specific issues. In more complex projects or times of greater scope uncertainty, synchonous communications become more more important. At these times, additional technology (e.g. WebEx, Skype) are deployed to enable real time viewing and editing of documents at regularly scheduled intervals.

Agile practitioners also commonly indicate the importance of team dynamics and the need for team trust to successfully deploy agile.[19] This trust is critical because in order to move swiftly, members must not be hesitant to critique the work of others. And all members of the team must recognize that such critiques are designed to improve the collective work, not as a personal attack. Building such a team is a complex undertaking, however certain steps can be taken, such as developing a team norms agreement, to improve the odds of successful team development. For best practices in this area, we recommend reviewing the findings of Katzenbach and Smith[20] or the team parable from Lencioni.[21] In the case of our team, we have expanded the membership of this team over time by adding members (friends) as interests and writing needs have expanded. Each of us is a member of various other writing teams, with overlapping memberships depending on the nature of the work being done.

Lessons Learned
Since our joint academic projects often focus on producing a document (journal article or conference paper) or a presentation, these observations are offered from that perspective. These observations are from our personal experiences and are not the results of a formal research study.
- Individual tasks should be small, and in line with agile methodology. Small tasks mean delivering small project increments in a short period of time. Quick feedback from collaborators creates momentum and generates 'wins' that improve output quality and team morale.
- Team members work in series so that we avoid working on the same material unless preparing for a meeting (either in person or by phone).
  - If two or more team members are working on the same document, each is assigned non-overlapping portions.
  - Normally only one team member is working on the document at a time. Reviews and edits are then done and circulated using a 'round-robin' approach.
  - It is critical to employ unique, descriptive file names that provide clear version control. For example, a draft document may be labeled as "ASEE2015 Agile v12."

- Early in any writing project, it is best to aim for completing roughly 80% of a document section.
  - There is an old adage that "The first 80% of a project takes 80% of the time. The other 20% of the project takes the other 80% of the time." This applies in that the closer a team member gets to a finished product, the more challenging it is to get it just right.
  - This leaves room for improvement without wasted work. Each person's 80% is different than a teammate's 80%. Leaving room for and relying on a writing partner to build upon and improve the initial work produces a better quality product with less rework. Your writing partners will have their own ideas, so trying to achieve a finished product will simply result in changes and frustration. The first, second, or third round will never be a finished product, so don't spend the time and energy trying to make it one. When working alone, obsessive wordsmithing may be needed. In teams, trying for perfection during each iteration becomes a barrier to achieving a timely high quality product.
- Regular communication is essential to keep everyone involved and the project moving forward. Clear incremental objectives, agreed-upon deadlines for these small steps (sprints), and progress reports maintain the productive engagement of all team members. It also helps the team member with the next task on a project to schedule the time to work on it. This in turn supports progress even in the face of competing time pressures. Silence from a team member doesn't mean that person is necessarily deeply engaged in their portion and deserves follow-up communication.
- Teaching is enhanced by using agile to remain flexible in how courses are designed, delivered, and assessed. Rather than sticking to a rigid plan for delivering X amount of content in Y period of time, deciding upon the critical content areas and structuring delivery to suit learning by the students can be more efficient and effective.
- The selection of a set of service opportunities to meet career, departmental, college, and/or university requirements is an essential part of success in academia. A specific service commitment that resembles a project may also benefit from an agile approach when the scope is unclear or adjustable as customer requirements become more apparent.
- Reading just a few reference sources on agile PM will help an academic interested in applying this technique to get a sense of how it can be adopted. Good places to start include the books by Sims & Johnson[22] and Layton.[23] The Sims & Johnson book in particular is a very short, condensed discussion of one particular agile PM technique (Scrum). While the authors focus upon the IT world, their very practical discussion can readily be expanded to other applications needed by the reader.

What Agile Has Done for Us

We have achieved good levels of productivity during our careers; however, our shift to managing our projects using agile methods has had a profound impact. Some things we are doing differently, and some things we are doing the same, but with a better understanding of *why* we are doing it. For example, two of the co-authors have collaborated very successfully on writing projects since 2006. Productivity in terms of journal articles and book chapters written since an *agile epiphany* in September 2012 has been even higher with several awards garnered for the work completed during that period. Agile project management has:

- Forced us to continually ask, 'how valuable will this project or task be?'
- Allowed us to deliver more and at a higher level of quality.
- Freed us from the nagging guilt that we were not managing our projects well (since we could not plan them).
- Allowed us to celebrate the scope and quality of a completed task, even if it took much longer than expected.

Summary and Conclusions

IT projects have demonstrated many times that success does not always come through better planning. IT projects were also the test bed for developing agile PM which is a clearly superior project management method for this environment. Academia has aspects in common with IT that make agile PM a viable choice for managing activities. Because academics are always juggling multiple projects over an academic year (or day), agile management of project scope is very likely to work better than better planning to execute the current scope.

This paper is a case study of applying agile PM in our personal career endeavors working together as a small team. It has presented a usable summary of conditions in academic projects where an agile PM approach may be superior to traditional PM tools. The following scenarios individually or in combination can complicate reliable project schedule planning.
- The project scope is unclear or poorly defined.
- Required task times are unknown or unknowable.
- The tasks and task dependencies are unknown.
- The availability of resources (money, time, people, etc.) is unknown or continuously changing.

This paper has described a small team multi-project environment that is subject to all four of the above conditions. This environment is common in academia, but it is not restricted to academia. A shift to an agile PM approach has improved our productivity, the quality of our output, and the quality of our working life. We contend that agile tools can work well for many academics. We hope that our work and the work of others on agile encourages more academics and practitioners to *adopt* and *adapt* these techniques so that more formal research on techniques, success rates, project size, etc. can be done.

Bibliography

1. Project Management Institute. (2015). What is Project Management? http://www.pmi.org/About-Us/ About-Us-What-is-Project-Management.aspx. Accessed January 9, 2015.

2. Kwak, Y.H. (2005). Brief History of Managing Projects. *The story of managing projects: An interdisciplinary approach*. Carayannis, E.G., Kwak, Y.-H., and Anbari, F. T., Editors. Praeger Publishers.

3. Beck, K., Beedle, M, van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development.* Retrieved from http://agilemanifesto.org/ .

4. Schwaber, K. (2004). *Agile project management with scrum.* Microsoft Press.

5. Cockburn, A. & Highsmith, J. (2001, November). Agile software development, the people factor. *IEEE Computer*, 34(11), 131-133. doi: 10.1109/2.963450 .

6. Fewell, J. (2012, July). All aboard. (The Agile Project Manager column) *PM Network*, 27(7), 26.

7. Beck, K. (1999). *Extreme programming explained: Embrace change*. Addison-Wesley Professional.

8. Highsmith, J. & Cockburn, A. (2001, September). Agile software development: the business of innovation. *IEEE Computer*, 34(9), 120-127.

9. Pyhäjärvi, M. & Rautiainen, K. (2004). Integrated testing and implementation into development. *Engineering Management Journal*, 16(1), 1-7.

10. Williams, L. & Cockburn, A. (2003, June). Agile software development: It's about feedback and change. *IEEE Computer*, 36(6), 39-43.

11. Gale, S.F. (2012, January). The evolution of agile. *PM Network*, 26(1), 28-33.

12. Redacted for blind review

13. Highsmith, J. (2003). Agile project management: Principles and tools. *Cutter Consortium Agile Project Management*, 4(2), 1-37.

14. Highsmith, J. (2010). *Agile project management, 2ⁿᵈ edition*. Addison-Wesley.

15. Gonzalez, W. (2013). Agile project management and the creation of intellectual property. *Dissertations Abstracts International Section A: Humanities and Social Sciences*, 73(7-A). ProQuest Information and Learning.

16. Blank, S. & Dorf, B. (2012). *The startup owner's manual.* K&S Ranch, Inc.

17. Kussmaul, C. (2004). Agile product development: Lessons from industry. *Education that Works: The NCIIA 8ᵗʰ Annual Meeting*, 125-129.

18. Gray, T. (2005). *Publish & Flourish: Become a prolific scholar.* Teaching Academy, New Mexico State University.

19. Cobb, C. (2011). *Making sense of agile project management; Balancing control and agility*. Wiley.

20. Katzenbach, J.R., & Smith, D.K. (2006). *The wisdom of teams: Creating the high-performance organization*. Collins Business Essentials.

21. Lencioni, P. (2002). *The five dysfunctions of a team: A leadership fable.* Jossey – Bass.

22. Sims, C. Johnson, H.L. (2012). *Scrum: A breathtakingly brief and agile introductions.* Sausalito, CA: Dymaxicon.

23. Layton, M.C. (2012). *Agile project management for dummies.* Hoboken, NJ: John Wiley & Sons, Inc.