Mashup of Waterfall, Agile and Lean Project Management Techniques

Steve Wille, PMP, CSM, MBA

Why would you want to mashup waterfall, agile and lean? While fundamentally different, each methodology offers value, so why not mix and match?

For many years I managed information technology teams in large corporations. As you might imagine, I have been through many management practices that have come and gone. Agile is the big thing today. I left the corporate world and now make my living consulting and teaching in the agile space. At the same time, I continue to work with companies in the traditional project management space. I see value in multiple approaches.

In the early 21st century, Six Sigma was a big thing, and you often saw "lean" added to it making Lean Six Sigma. At General Electric, if you were not doing Six Sigma there was no place for you, according to a 2007 article in Business Week written by Jack Welch, the GE CEO.[1]

For many years Six Sigma looked good on a resume. Today, according to an article in *Quartz at Work*[2], Six Sigma is falling out of favor and agile is picking up.[2] LinkedIn sees fewer and fewer of its 630 million users showing Six Sigma as a skill on their online résumés. It is now surpassed by agile, a management process that emerged from the world of software development.

> "Six Sigma's decline was also a symptom of a broader change in the corporate world, where innovation became more valued than efficiency, and technical precision was no longer a differentiator. Silicon Valley's culture of 'move fast and break things' meant business leaders were less concerned with reliability and more focused on game-changing discoveries. An obsession with efficiency, researchers have discovered, can come at the expense of invention."

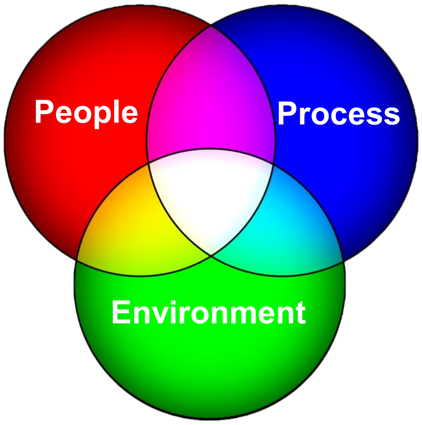
> "'When I get up on an airplane, I'm very glad it went through a Six Sigma process—there's a certain comfort in that,' says Mike Pino, a technology strategist at PwC who spent three years at GE. But at organizations built around Six Sigma, he says, 'disruptive innovation is discouraged.'"

This is just one datapoint, but it is consistent with a megatrend of companies looking to agile for product development.

There is another interesting datapoint showing the agile megatrend. Traditional project management is searched on Google fewer times than agile practices.[3] In the early 21st century, the Project Management Professional (PMP) certification was the gold standard for hiring people to lead projects. Large corporations set up Program Management Offices (PMO) to enforce the common practices documented by the Project Management Institute (PMI). Today, in my teaching and consulting practice, I see agile everywhere I go, and see interest decreasing in traditional project management in technology projects. Traditional project management techniques are still a large part of construction.

Another indication of this trend of moving away from traditional project management to agile is the revised PMP exam. In 2020 it will focus on three domains: ***process, people,*** and ***business environment***.[4] Traditionally, the PMP certifications focused primary on the ***process*** of managing projects. The process focus was apparent in the 2017 version of A *Guide to the Project Management Body of Knowledge*[5] (PMBOK). The word "human" was dropped from the section that covered human resources. People were just resources, nothing more. They were fungible, meaning that they were interchangeable according to skills, just like money is interchangeable. I can loan you money, and you can pay back the money, but it does not have to be the same pieces of paper money.

It is a breath of fresh air to see the PMP exam will place emphasis on people. People are not interchangeable resources. It is a bigger breath of fresh air to see the exam looking at the business environment. These three things come together to form a whole project. It is like red, green, and blue light coming together to make white light. You cannot have white light coming from your television unless you have equal amounts of red, green, and blue pixels light up. A holistic view of project management recognizes that in an environment, people are collaborating, using appropriate processes and tools.

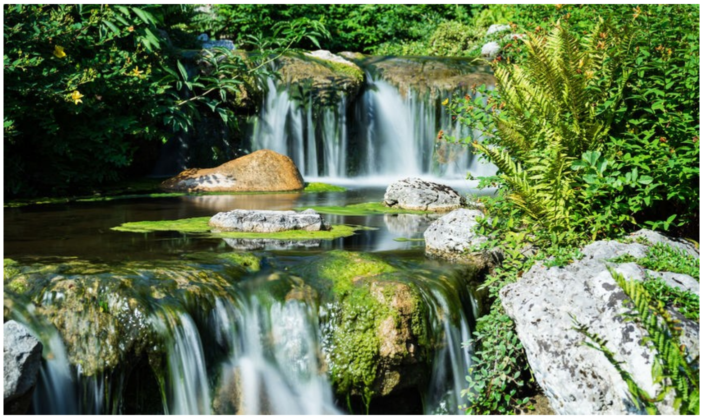


Domain 1—Process

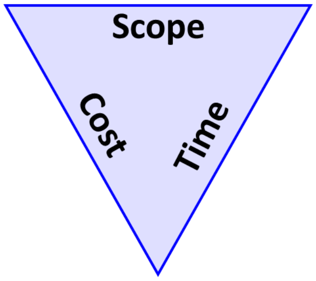
I want to start our mashup by looking at the process side of project management, represented by the blue light in the diagram. There are many processes and tools. One size does not fit all. You can drive in a screw with a hammer, but it works better to use a screwdriver. I want to explore the many processes and recognize where they are appropriate.

On the process side, the PMP exam will be half predictive (waterfall) project management. The other half will be agile or hybrid. This is another datapoint pointing to the agile megatrend.

Let's start with waterfall. Traditional project management is broken into phases such as initiation, requirements, design, planning, development, testing and implementation. You need to complete each phase before moving to the next because it is expensive to make changes late in the project. The reason for the term waterfall is because you don't want to go back up the falls. Once you go down, you continue to the next phase.

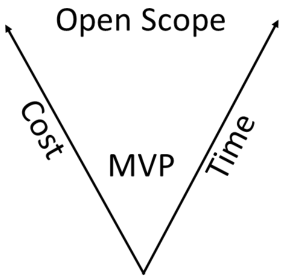


Waterfall project management originated in construction where there were dependencies such as digging a hole before pouring a foundation, and building the foundation before framing the structure. Construction project managers pay attention to scheduling of materials and labor. They find the critical path in the construction process so they can deliver what was promised on time and on budget. Traditional project management focuses on the triple constraints of ***cost, time,*** and ***scope***.



Following the water metaphor, agile is like going down the rapids. You just keep going. You will get wet, and you might die if you don't work together with the people on your raft. In real time, you read the water flowing over the rocks. With agile, you move fast, and you keep moving.

Agile does not work with the triple constraints. Time and money are budgeted against a prioritized backlog of desired features. Agile teams typically start with a ***minimum viable product*** (MVP) that they take to market, and then add more features. The scope is open and the development continues as long as management keeps the funding going, indicating that they want more features.



The water eventually flows into great rivers that move across the continent. The river might look slow, but a great amount of water moves through. This is like the lean method that we learn from manufacturing where great numbers of identical products are built. Along the way, manufacturers look for efficiencies. Lean is based on the Toyota Production System[6] (TPS). It looks for opportunities to reduce waste. Defects are wasteful so an emphasis on quality is essential. In corporations the operations must be lean, because wasteful inefficiencies will drive you out of business as other companies do better. In technology the operations must be lean so they are secure and dependable.

Drivers for waterfall, agile, and lean

The driver for waterfall is to deliver a predictable outcome, on time, on budget, and full scope. The driver for agile is to create opportunity for innovation. The team can experiment and fail fast. The driver for lean is control. When you are making many copies of the same thing you can benefit from statistical process control.

Each method delivers quality in a different way. With waterfall, there is a long testing phase and nothing goes into production without sign-off to complete this phase. With agile, each feature is tested as it is built. Because of incremental delivery, if there is a problem it will be found quickly, and appropriate action taken. With lean, because it is a repetitive process, there can be continuous improvement with statistical process controls.

Let's switch from the water metaphor to sports to see

how people interact in these three disciplines. Baseball is highly sequential, meaning you stand and wait for the ball to come to you. When it is your ball, you must do your job with precision, passing the ball to the next person. When your team is at bat, everyone but the batter is sitting and waiting for a turn at bat. The game is over when the full scope of 9 innings are completed according to the rules. There is no time limit on the game. You take your time and do your job right.

Waterfall project management is ***sequential***. Requirements do not start until the project is approved. The design team waits for the requirements to be completed. The development team waits for the design to be completed. According to PMBOK, the bulk of planning is done upfront and the execution is in a single pass. It is a sequential process.

Following the sports metaphor, basketball is the opposite of sequential. It is ***random***. Any player can go anywhere at any time and any player can shoot for the basket at any time. If you stand and wait for the ball to come to you, the coach will tell you to move. Move anywhere and find an opening. Shoot the ball. You might miss, but there will be someone there to go for the rebound. The game is over when the buzzer rings. Your goal is to make as many baskets as possible in the short time allowed.

Agile is random. Typically, small persistent teams work the backlog and do their own planning. Anyone with the right skills can do any task without constraints of position or title. The key is to keep moving. Deliver something now. There is no penalty for doing it wrong the first time. Fail fast and try again. You have limited time so keep moving.

Golf completes the sports metaphor. Every stroke counts. It is wasteful to miss the hole when you putt because that small mistake counts as much as a great long distance drive down the fairway. Golfers keep score and look for ***continuous improvements*** to improve their scores, game after game.

Lean is like golf. Every stroke counts and organizations want consistency with continuous improvement, always looking for efficiencies leading to less waste.

Let's take a moment to ponder the significance of sequential, random and continuous timing. If you want to be more agile you must be more random. It is ok to fail and try again. This is where innovation comes from. If you have a lean coach directing an agile team, innovation will be driven out because innovation is wasteful. To a lean coach, it is better to make incremental improvements to a working process than to take a risk on a new idea that may or may not work. Likewise, an agile coach can mess up a lean process by being too random. A waterfall coach will slow everyone down by telling them to wait until a phase is completely done before moving on to the next phase. Sequential coaches want to do things the right way in the right order, with each step completed and signed off by the project sponsor. Random coaches want customer happiness and will keep trying until they get there.

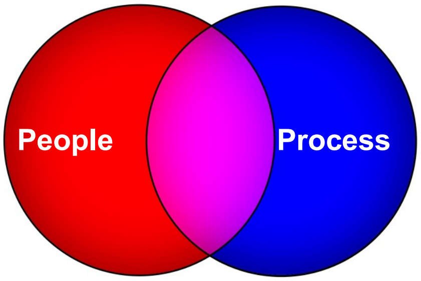
What a mess. If we do everything just one way, it will work when it works, but think of the opportunities missed. Maybe we can take elements from each and apply them across methodologies.

Who is to say you cannot be little bit random in what is otherwise a waterfall project? Within the agile project you can be a little bit sequential. In the lean world you can create agile boxes where people can experiment without penalty. No process must be "all in" where everyone must do everything the same way.

An example of using a common agile practice in a waterfall project is writing user stories rather than requirements. Requirements are everything you must do. User stories are features that people want. Requirements are specific and measurable. User stories contain the essence, not the details. User stories say who wants something, and why they want it. They also contain acceptance criteria giving you a self contained test.

Domain 2—People

Moving from the blue light representing process to the red light representing people, we get the combination of talented people using tools that work for them.



Is there a universal best tool? Of course not. We have individual preferences. Consider an Apple laptop compared to a Windows laptop. They both do about the same things, running some of the same software, but many people have strong preferences. There is no best. What matters is what works at what price. Beyond that, it is personal preference. But don't tell that to Apple advocates because they will insist you are wrong.

The new PMP exam will be 42% on the people side. Within that there 14 items to cover, of which 8 use the word team.

1. Conflict
2. Lead a ***team***
3. ***Team*** performance
4. Empower ***team*** members and stakeholders
5. ***Team*** members/stakeholders are adequately trained
6. Build a ***team***
7. ***Team*** impediments
8. Negotiate
9. Collaborate with stakeholders
10. Build shared understanding
11. Virtual ***teams***
12. ***Team*** ground rules
13. Mentor stakeholders
14. Emotional intelligence

The exam writers are not bound by the PMBOK sixth edition. It lacks much of the material that will be covered in the exam. You will have to go to other sources for good practices for leading teams.

If not PMBOK, then what is a good source? I did a Google search on "leading a team" and got 1.6 billion hits. That is a little more than I want to read so I went to Amazon to find a book on the subject and found over a thousand books. Where should one start?

A good place to start on leading technology teams is with the ***Manifesto for Software Development***[7]. It was written by some of the leading thinkers who had developed what would come to be called "Agile." In 2001 they met in Park City, Utah, and invented the term agile. Together they wrote the key values of agile, the first of which is "***Individuals and Interactions over processes and tools.***"

Jim Highsmith wrote the history of the event that you can still read on the Internet.

> "While the Manifesto provides some specific ideas, there is a deeper theme that drives many, but not all, to be sure, members of the alliance. At the close of the two-day meeting, Bob Martin joked that he was about to make a 'mushy' statement. But while tinged with humor, few disagreed with Bob's sentiments—that we all felt privileged to work with a group of people who held a set of compatible values, a set of values based on trust and respect for each other and promoting organizational models based on people, collaboration, and building the types of organizational communities in which we would want to work. At the core, I believe Agile Methodologists are really about 'mushy' stuff—about delivering good products to customers by operating in an environment that does more than talk about 'people as our most important asset' but actually 'acts' as if people were the most important, and lose the word 'asset'. So in the final analysis, the meteoric rise of interest in—and sometimes tremendous criticism of—Agile Methodologies is about the mushy stuff of values and culture."

The group wrote 12 principles, 5 of which focus more on people than on process or techniques. These principles apply to any situation where you want knowledge workers to be engaged in what they are doing. This is probably the best place to start when considering how to develop teamwork. I have highlighted the key words in each of these principles.

4. Business people and developers must ***work together daily*** throughout the project.

5. Build projects around ***motivated individuals***. ***Give them the environment and support*** they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is ***face-to-face conversation***.

11. The best architectures, requirements, and designs emerge from ***self-organizing teams***.

12. ***At regular intervals, the team reflects*** on how to become more effective, then tunes and adjusts its behavior accordingly.

Many of the common agile practices are built around these five principles. These practices can be applied to management of knowledge workers all the time, regardless of project methodology.

An example of principle 12, team reflection, is the agile practice of a retrospection after every sprint. This is a closed door meeting were team members can openly discuss what is going well, what is not going well, and what should be changed. By contrast, the waterfall "lessons learned" meeting takes place at the end of the project when it can add no value to the project. Waterfall and lean teams can meet at regular intervals to reflect on how the team is performing, and it can be a closed door meeting with no written documentation so people can be open and honest without penalty.

An example of principle 11, self-organization teams, is the scrum team that has no roles or titles. Anyone can do anything, just like a basketball team. Waterfall teams are often staffed by specialized "resources" that are allowed to do just their specialty. Why not have a waterfall team that stays together over the life of the project? People can expand their skills, getting help from the more experienced members of the team.

Consider principle 6, face-to-face communication. Waterfall projects tend to be heavy in written documentation and written status reports. Agile teams tend to interact in real time and minimize written reports. Waterfall teams could work in real time, too.

Principle 5, motivated individuals, calls for supporting the team with the environment they need. Agile team scrum masters are expected to be servant leaders, doing what they can to support the team. Waterfall project managers could be servant leaders, too.

At the core of agile is principle, 4, daily interaction. Many waterfall teams meet weekly with a long status report. They could skip the weekly meeting and do a quick daily standup. The point is, these agile practices can be applied to any team of knowledge workers where you want engaged people.

Domain 3– Business Environment

The third domain on the PMP exam is the business environment. It is only 8% of the exam, but it forms the basis for everything else. There must be compliance with the organization's policies, whatever they are. There must be a return on the investment. These things apply to all projects. Driving the move to agile and lean is the external environment. Becoming agile may require some change in the organization's culture.

Domain III Business Environment—8%
1. Project compliance
2. Project benefits and value
3. External business environment changes for impact on scope
4. Organizational change

Amazon and Google keep trying new ideas and some of them work. Who would have thought in the late 20th century that these two new companies would set the competitive standard for the early 21st century? This drives me to think of the butterfly effect which suggests that small things can make big differences, but we cannot know what small things will affect what big things. We will never be able to predict the future, regardless of how smart we become with algorithms and artificial intelligence.

The butterfly effect is driving the movement to agile. Try something. Some things will work out, others will fail. In the agile world, you fail fast, allowing room for other new ideas. This is how innovation happens and how new ideas get to market quickly. At the same time, Amazon sets the standard for efficiency. Their warehouse and delivery systems are as lean as you can imagine. It is hard to compete with that. If your company is not agile with development and lean in operations, you probably will not survive.

Going back to the *Manifesto for Agile Software Development*, if you want your organization to ever become agile, you must start by valuing individuals and interactions over processes and tools.

If you are to value people over process, you must accept that ***it is OK to do the right things the wrong way***, rather than insisting that everyone follow the one best practice.

A number of years ago I worked for a large insurance company in Philadelphia where I tried some unconventional ideas in data processing. It was discovered by management that I was accomplishing things that had business value, but no one else had been able to do. I was promoted to officer level of that corporation and given more staff and budget to do more good stuff. This went on for several years until a management change and suddenly I was not so appreciated. The new president told me, "You are doing the right things the wrong way." He wanted me to report to someone who could teach me the right way to do things. I asked him, "What's the other choice?" Needless to say, it was time for me to find a new job with a company that was interested in what I could accomplish for them.

Process over People

If you value **process over people**, it is probably because you value predictability and control rather than innovation. If everyone does the same things the same way, most people will meet your expectations. When

	Right Way Process over people	**Wrong Way** People over process
Exceed Expectations	Disengagement ☹	Engagement ☺
Meet Expectations	Compliance ☺	Engagement ☺
Below Expectations	Get training ☹	Get training ☹

some people perform below expectations, you can send them to training to learn the right way to do things and most of them will learn. It is ok if some people perform above expectations, but they will gradually become discouraged and disengaged because the system is rigged for meeting expectations.

People over Process

If you value people over process you make room for people to try new ideas and perform above expectations.

The challenge within the organization is when you allow individuals freedom to do things their own way, the people who work on doing things the right way can become irritated. The human tendency is to get even with the people who do things the wrong way and still succeed. If you want to make room for agile teams, you must accept the idea that there is no one best way to do every task. Give teams freedom and support them. Read the agile principles again and take them to heart.

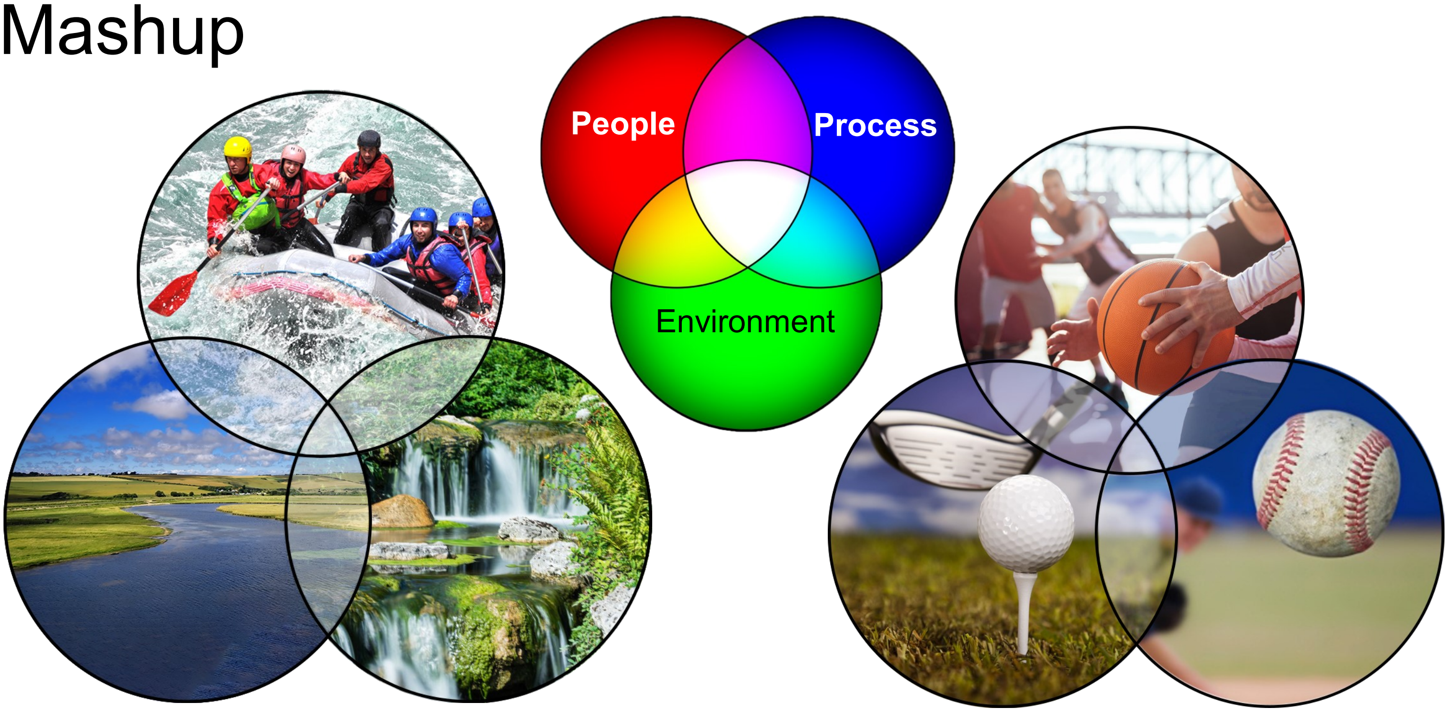
This takes us back to the three methodologies we have reviewed. Process is at the core of lean and waterfall. People are fungible, interchangeable resources. The best you can expect is to meet expectations. At the core of agile is teamwork that leads to innovation and the possibility of exceeding expectations. Agile is messy and wasteful, but can lead to great things.

Consider moving waterfall and lean projects into the teamwork realm, allowing freedom to try new ideas that might work. To do this you need to get away from measuring individual performance and look at team performance, letting people on the team figure out how to accomplish what needs to be done. The same team doing development might be doing support. Let them decide when to use waterfall, agile, and lean techniques.

This takes us to an organization culture where it is ok to do different things in different ways, rather than following a single common practice. Some projects are sequential in nature. Innovation requires randomness. Some things are best accomplished through continuous improvement. This same team can play three different games depending on what needs to be accomplished.

As we mash up methodologies and timing, let's not forget that things get done by people working within an environment, using processes and tools that work for them.

Mashup



Manifesto for Agile Software Development

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.

We follow these principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must ***work together daily*** throughout the project.
5. Build projects around ***motivated individuals. Give them the environment and support*** they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is ***face-to-face conversation***.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from ***self-organizing teams***.
12. At ***regular intervals, the team reflects*** on how to become more effective, then tunes and adjusts its behavior accordingly.

1. Business Week, May 21, 2007 page 110, ideas the Welch Way, Jack and Suzy Welch
2. https://qz.com/work/1635960/whatever-happened-to-six-sigma/ Whatever happened to six sigma
3. https://onlinepmcourses.com/cant-ignore-agile-methodologies/ Why You Can No Longer Ignore Agile Methodologies
4. PMI PMP Examination Content Outline, June 2019, Project Management Institute, Inc.
5. A Guide to the Project Management Body Of Knowledge (PMBOK® Guide) Sixth Edition 2017 Project Management Institute
6. https://global.toyota/en/company/vision-and-philosophy/production-system/
7. https://agilemanifesto.org/history.html
8. Stock photographs from https://123rf.com and https://www.dreamstime.com/
9. Graphics design by Steve Wille