Zero Sum Games:
Agile vs. Waterfall Project Management Methods

By Steven F. Wille, PMP, with Michael S. Wille, PMP, CSM

Agile software development is hitting the mainstream and is starting to displace the traditional project management method known as waterfall. It is useful to step back to see what we are gaining and what we are losing with each approach. We will show that picking one or the other approach may be a zero sum game and we will offer an alternative approach that is non-zero sum.

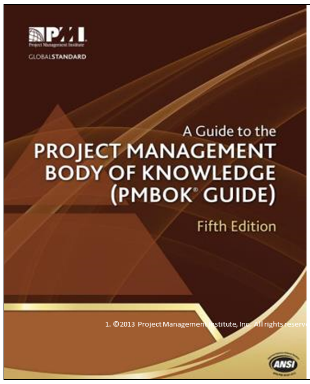
Let's start with some definitions. In a **zero sum game**, each participant's gain or loss is balanced by the losses or gains of the other participants. If we assign a numeric score of 1 for winning and -1 for losing, the sum total will always be zero because there is always a winner and a loser. We can play the game a thousand times and the sum score will still be zero.

Waterfall is a sequential software development processes, in which progress is seen as flowing steadily through the project phases. The water flows in one direction symbolizing that once a phase is completed you don't get to go back to make changes. This is because changes are expensive late in a project time line as compared to getting things right in the design phase. Overall project efficiency is the goal of waterfall development.

Our source for defining traditional waterfall project management is the Project Management Institute's book, *A Guide to the Project Management Body of Knowledge (PMBOK Guide*). This was first published in the 1990s. It documents good practices in project management. The traditional phases of a project are *initiating, planning, executing, controlling,* and *closing*.

PMBOK covers ten knowledge areas with particular attention given to **scope**, **time**, and **cost**. These are known as the triple constraints. It is easier to accomplish two of the triple constraints if you don't care about the third. For example, it is easier to be on time and on budget if you don't care about scope. The hard thing to do is to be on time, on budget, with the full scope of features promised. Effective project management is all about being on time, on budget, and with full scope.

Take note that the for all PMOBK knowledge areas ***management*** is a key word. There are many definitions of management. We will define it as the opposite of unmanaged, chaotic, and out of control.



- Integration Management
- Scope Management
- Time Management
- Cost Management
- Quality Management
- Human Resource Management
- Communications Management
- Risk Management
- Procurement Management
- Stakeholder Management

The dictionary definition of **agile** is, “marked by ready ability to move with quick easy grace, <*an agile dancer*>. ” The second definition is “having a quick resourceful and adaptable character, <*an agile mind*>” (merriam-webster.com) . The theme here is flexibility and speed, with resourcefulness and adaptability.

We will define agile project management as it is expressed in the *Agile Manifesto,* published in 2001. This is a brief document containing four high level strategies and twelve principles. You can find this on line and print it yourself.

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:

1. **Individuals and interactions** over **processes and tools.**
2. **Working software** over **comprehensive documentation.**
3. **Customer collaboration** over **contract negotiation**.
4. **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.

(highlighting added)

Rather than being like a peaceful waterfall, agile projects are like rafting down the rapids. Change can happen any time, even late in the process. Watch where you are going and think fast!

Now that we have defined the terms, let’s explore the strategy for playing a zero sum game. Consider flipping a coin. There are two, and only two outcomes. On each throw you have a 50% chance of winning regardless of which you call, heads or tails. No strategy is required because your choice makes no difference when there is a 50% chance of winning and losing.

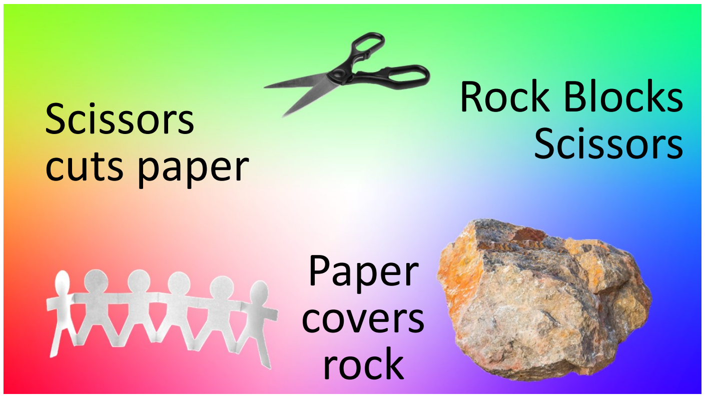
When faced with two choices, it is helpful to ask if there is a third option. This opens things up to new possibilities. Rock paper scissors is a zero sum game with three options, and it requires a little bit of strategy.

Two people play. Each person makes the sign for a rock, paper, or scissors. If both pick the same option, the

+1 Rock Wins

-1 Scissors Loses

0 Sum



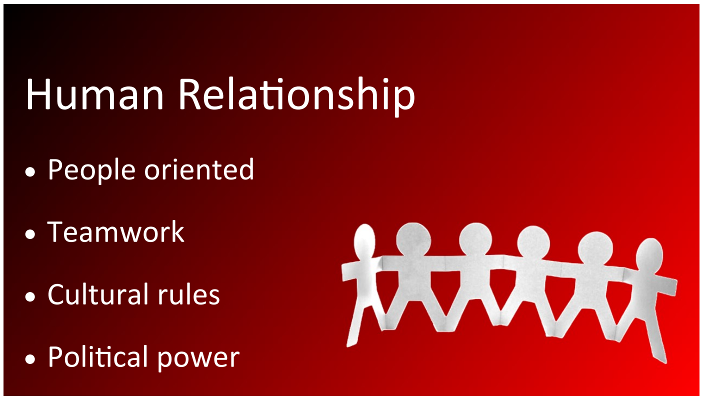
Scissors cuts paper

Rock Blocks Scissors

Paper covers rock

Rock Solid

- Compliance
- Financially sound
- Repeatable processes
- Best Practices



Human Relationship

- People oriented
- Teamwork
- Cultural rules
- Political power

Human Ingenuity

- Cut away from tradition
- Disruptive innovation
- Looks chaotic

score is zero and you replay. If they are different, then the following rules are followed.

- Scissors cuts paper, paper loses.
- Rock blocks scissors, scissors loses.
- Paper covers rock, rock loses.

For each option, there is a way to win and way to lose. The worst strategy is to play the same option every time. Your opponent will soon figure out how to beat your strategy. The best strategy is to be totally unpredictable.

The rock symbolizes rock solid management where you are in compliance with all regulations and expectations. Your decisions are financially sound with the required return on investment balanced by an acceptable level of risk. Processes and procedures are repeatable, and hopefully, show continuous improvement.

Paper symbolizes the people side of project management. Keep in mind that you can manage processes, but people are not processes. You must lead people. Feelings matter. You must pay attention to teamwork when leading a group of people and you must follow cultural and social rules. Confounding all of this is the part that politics play in any organization infested with human beings.

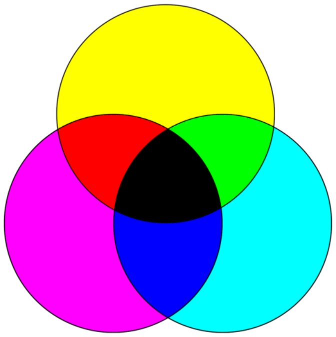
The scissors represents human ingenuity. People are always finding new ways to do things. Sometimes new ways drive out old, disrupting the old order. The scissors lets people cut away from the past to focus on the present and future. It can look chaotic, but order often emerges from chaos. With the scissors you can change the shape of things. With the rapid change in technology we are seeing in the 21st century, flexibility is essential for survival.

Another way to look at the scissors is as a means to get things done, even in an unpleasant way. A surgeon may have to cut you to heal you. A carpenter has to cut the wood to make the structure. The artist has to chip away the stone to liberate the statue that is inside.

Even with three options, let's recognize that we are still playing a zero sum game. Each option can block another option. For example, a rock solid organization can block innovation. Rock solid is good, but not all the time. Sometimes it is a path to going out of business because the world has changed and the market has moved on before the organization can catch up.

Let's look at this from the perspective of filters that block light. Cyan blocks red and lets the green and blue pass. Magenta blocks green and lets the red and blue pass. Yellow blocks blue and lets red and green pass. If you stack all the filters together, no light passes. This is how your **ink jet printer** works. You start with white paper that reflects every color. As you spray on cyan, magenta, and yellow ink, you block light to create many colors. If you spray on all three colors, you get black where no light reflects from the white paper.

The additive color process is different. Imagine a dark room where you shine red, green, and blue lights on the wall. Where they all shine together, you see white. Where they mix, you see other colors. This is how your **television** works. Red, green, and blue lights work together to create a full color, high definition picture.

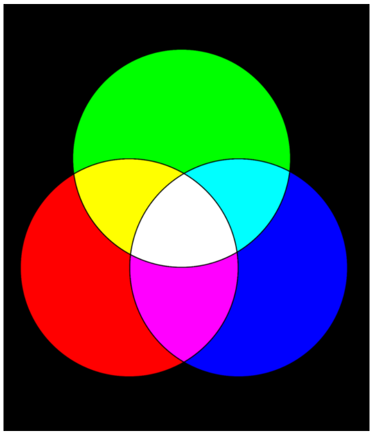


Inkjet Printer

Subtractive process

Start with white paper

- Yellow blocks blue
- Cyan blocks Red
- Magenta blocks Green
- Zero light in center



Television

Additive process

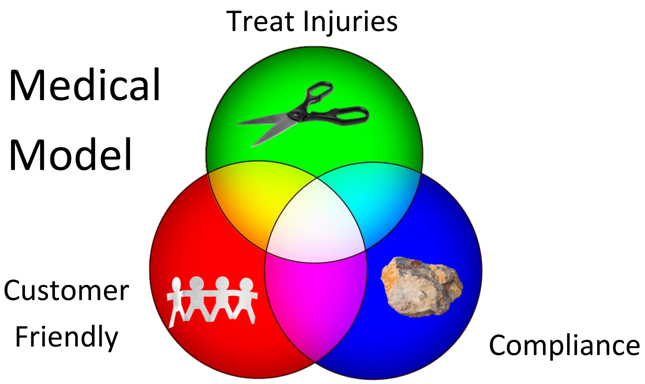
Start with dark screen

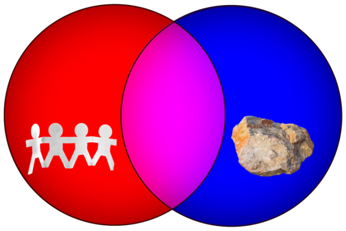
- Green + Blue = Cyan
- Blue + Red = Magenta
- Green + Red = Yellow
- Red + Blue + Green =White

Lets look at this zero sum game from the perspective of the additive light process in order to turn it into a non-zero sum game. Instead of thinking rock, paper, ***or*** scissors, think rock, paper ***and*** scissors. Imagine a team where everyone is required to be a rock. There would be no room for human feelings or human ingenuity. If you have representation from all three, and encourage rather than blocking other perspectives, your opportunity for sustained success increases substantially.

Not everyone in my family is in a technology professional. Some work in the medical field. I heard my wife and my daughter talking about the pressures of working in the medical profession. That discussion was the inspiration for this paper. In the age of for-profit hospitals you must be customer friendly so that people will come to your hospital to spend their unlimited health care dollars provided by insurance companies or the government. In addition you must be in compliance with corporate policies, insurance policies, and government regulations. In addition, patient care should be your primary concern. In an ideal world, there would be time and energy for all three, but in the less than ideal world, something has to give.

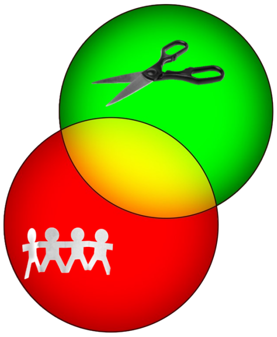
Hospital employees can be measured on compliance and customer friendliness, but it is hard to measure the real job of patient care. Consequently, there is pressure for these two at the expense of the third. In addition, corporations find it more profitable to have less staff, further increasing the pressure. The most profit would come from compliance and customer friendliness with just adequate patient care. This would fill the beds and the compliance part would assure minimally adequate patient care.





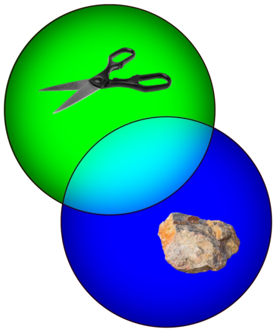
20th Century Model

- Rock solid
- People aligned



Millennial Model

- Innovation
- Collaboration
- Built on sand



Outsource

- Innovation
- Rock solid
- Heartless

Back in the 20th century some large organizations survived and thrived by being rock solid with employees fully aligned with corporate goals. Individual ingenuity was limited to the people who had the appropriate roles. Everyone else had to stay in line. It was a system that worked well in the industrial age.

We are in the 21st century and technology is changing rapidly. If you stand still you fall behind. Organizations need human ingenuity and collaboration to get ahead of the competition. Ingenuity and collaboration are essential for accomplishing that goal. This suggests the scissors and paper may be more appropriate for the 21st century organization.

Dan Lyons, in *Disrupted, My Misadventure in the Start-Up Bubble*, wrote about what it was like to be a 52 year old, seasoned journalist, working in a startup where most employees were in their 20s, many right out of college. The company, *Hubspot*, had a $100 million in venture capital and Lyons was offered stock options for a possible initial public offering (IPO). The book was a humorous read as Lyons described the fraternity like culture. It was anything but the rock solid environment described in this paper. When the company went public in 2014, it was losing money, $34 million according its S-1 document with the U.S. Securities and Exchange Commission. As of the 4th quarter of 2016, it was still losing millions of dollars. Even with the losses, the founders made a lot of money when the company went public. You can see why Lyons references the *start-up bubble* in the book title. You can make your own conclusions about the value of a company that is not rock solid and has never made a profit.

The other way to survive in the 21st century is to outsource everything, leaving little to no work for local employees. This is a picture of an organization without a heart. It may be profitable, rock solid, and innovative even though it has no care for people.

In the traditional waterfall world the goal is to deliver everything promised on time and on budget. The project team is in full alignment with the organization's goals and the project is managed in accordance with the organization's standards. We see this as rock solid with a highly organized team.

In the agile world the goal is to work within the time frame and budget to deliver what can be delivered when it can be delivered. There is high collaboration between the project team and the product owners with a great deal of flexibility in how the project team works to make this happen.

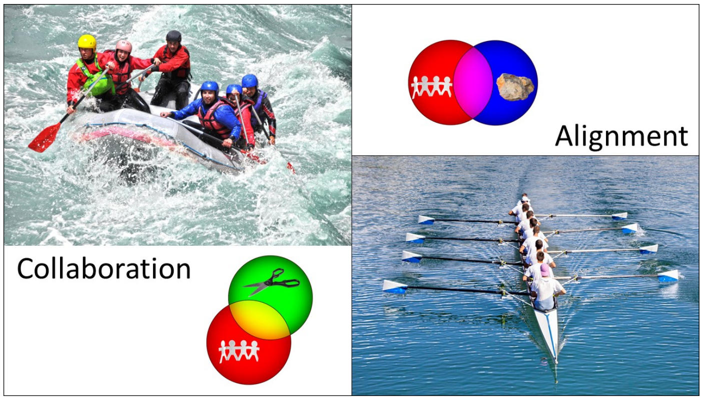
The outsource option is always out there and cannot be forgotten, but we will not be focusing on that option. We will focus on what is being delivered and how the internal project team interacts with the product owners.

Waterfall and agile approaches appear to be at odds with each other and could be seen as mutually exclusive. Inder Sidhu, in the book, *Doing Both*: *How Cisco Captures Today's Profit and Drives Tomorrow's Growth*, pointed out that much of Cisco's strategic success came from doing both when faced with an either/or decision. The cover of his book shows the Golden Gate Bridge. Originally, the bridge was going to look like any other bridge that carried cars over an expanse of water. When faced with a decision to focus on the utility of the bridge or to make it a work of art, they did both. The beauty of the bridge makes it a landmark that immediately identifies it with San Francisco. Sidhu says it is easy to settle for one option, but it is better to do both. Many companies focus on existing markets and fail to reach new markets. Other companies focus on the new markets at the expense of existing ones. Cisco did both. The added marginal cost of doing both is often small compared to the initial cost of doing one.

How do you do both? Take out the ***but*** and insert an ***and*** when you are considering one choice over another. For example if you have a great idea and another person has a competing idea, you might say, “I heard what you said, ***but*** my option will take us in a new direction to open up more possibilities.” That puts you in an adversarial position where one of you will win and one will lose. The alternate is to say, “I heard what you said, ***and*** my other option will take us in a new direction to open up more possibilities.” Now both options are up for discussion and there might be some middle ground. Often things that at first appear to be mutually exclusive are not. The problem is we focus too much on one option and fail to accommodate others.

Best practice infers the one best way to do any particular task. This is can be zero sum because it is all or nothing, right or wrong. There are often multiple ways to accomplish a task with equally good results. This simple way to get past zero sum is to focus on good practices rather than the one best practice. Good practices can be good enough to get the job done, allowing for individual creativity in finding a better way. It should be noted that waterfall and agile methodologies both have best practice advocates. Formal training is available for both methods. Many organizations are place great emphasis on doing one method the right way. The problem with this is that people get things done. Methodologies are just tools that people use. **Common practices** or **good practices** would be non-zero terms to describe what we can learn from a methodology, but the best method for a particular project can vary. Rules are a zero sum way to get compliance with best practices. Options are a non-zero sum path to blending human ingenuity with rock solid. The real goal is to get the job done. A blended approach makes room for people to focus on the end goal and find a path to success.

Let’s try blending the rock, paper, and scissors. The rock and scissors are both task oriented while paper is people oriented. How the team is organized often depends on the task. Most projects have scissors and rock tasks. The team will function better if is organized to do both.



Alignment - Collaboration

We have already touched on collaboration and alignment. Consider a rowing team on relatively flat water. The rowing team is most efficient if everyone is aligned, doing the exact same thing at the exact same time. To accomplish this, in the crew there is a coxswain who sits in the back facing the front. The coxswain is responsible for steering the boat, and coordinating the power and rhythm of the rowers who are facing the back and cannot see where they are going.

Next, consider a rubber raft running the rapids in fast moving, dangerous water. People must coordinate their efforts, but due to rapidly changing conditions they must be flexible. Everyone faces the front and pays attention to where the raft is going. There is no coxswain keeping time. Instead, the leader is yelling out what needs to be done. Can one athlete do both? Of course, but not at the same time. When rowing, alignment counts. When paddling, experienced judgment counts. Some aspects of a project require an aligned team and other aspects require experienced judgment with a great deal of flexibility. The mistake is to be a coxswain when a different kind of leadership is needed.

Clear roles - Ambiguous roles

Sometimes we need clear roles and other times we need ambiguity with self-organizing teams. It is great to identify who is responsible for what, and it is also great to give people freedom rather than putting them in a predefined box. If you want innovation, you need ambiguity. If you want predictability and control you need clarity. Can you do both on one project? Of course you can. Some tasks need great clarity and other tasks require creativity. Too much control kills creativity.

The agile principles that address this are:

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

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**.

By contrast, traditional projects usually clarify roles, often through the *RACI Matrix*:

- **R**esponsible (Persons who will be performing the work)
- **A**ccountable (Person who is ultimately held accountable)
- **C**onsulted (Subject matter experts)
- **I**nformed (Communication is one-direction)

Requirements - User stories

Project scope defines what needs to be accomplished. In the waterfall world, requirements define what will be done. ***SMART*** requirements are ***S****pecific*, ***M****easurable* ***A****chievable*, ***R****elevant*, and ***T****ime bound*. There is to be no deviation from these specific requirements without formal approval, even if you could do something better, faster, and at lower cost. You may not deviate without approval.

Smart Requirements

Specific
Measurable
Achievable
Relevant
Time bound

The system shall...

User Stories

- Ambiguous
- Brief
- Written by end users

As a <*role*> I need to <*what*> so that < *why*>

In the agile world there is intentional ambiguity in defining what needs to be done. User stories are commonly used and they are written from the user's perspective. They define what is needed and why it is needed. The agile team is expected to focus on ***<u>why</u>*** when viewing the user story and figure out how to best accomplish this objective. There is room for creativity. Ideas on accomplishing the goal better, faster, and at lower cost are always welcome.

User stories and requirements can both be used for quality assurance testing. Requirements are specific enough to be inserted directly into the test plan. Acceptance criteria are frequently included in the user story and also can be inserted into the test plan. Here are some examples:

Requirements

1. The system shall interact on-line with Mastercard, Visa, American Express, and Discover cards.
2. The system shall transmit name, address, phone, purchase amount, credit card number, and expiration date to the credit card company.
3. The system shall record the authorization from the credit card company.
4. The system shall comply with PCI security requirements for credit cards.

User Story

As a sales agent, I need to accept credit cards over the phone and verify the information so that I can complete the purchase.

Acceptance Criteria

1. This works with all credit cards accepted by our company.
2. There is a way to verify the information with the credit card organization.
3. Customer data is kept secure.

The *International Institute of Business Analysis (IIBA)* has recognized the evolving face of how requirements can be written. The current version of the BABOK says, "The nature of the representation may be a document (or set of documents), but ***can vary widely*** depending on the circumstances." Clearly you can do both, depending on what is needed for the project. If you know exactly what you want, go for requirements. If you are interested in what you think you want, or something better, then go for user stories. There is no reason not to use both on the same project, depending on the specific needs to be accomplished.

Timing: Random—Sequential

Timing is everything. Ask any musician and you will learn that timing makes the music come alive. The same is true with projects. One of the fundamental differences between agile and waterfall projects is timing. It is also one of the areas driving the greatest division of opinion. What is the right timing for a project task? Waterfall calls for sequential, step-by-step timing with an emphasis on the critical path which is identified during the planning phase. Changes require formal approval with the expectation that the change will affect the project cost and schedule.

Agile promotes more random timing with a focus on the next deliverable, along with creating a minimum viable product. Changes to requirements, even late in development are welcome.

Opinions on the best timing often reflect personal preferences rather than project needs. Anthony Gregorc noticed difference in how we perceive time and saw preferences in how we handle timing. In *An Adult's Guide to Style* he identified predictable patterns. He took no stand on what is best, just that there are common differences in style.

Sequential Time

- Discrete units of past, present, and future.
- Ordering ability: step-by-step linear progression of activities.

Random Time

- Now: total of the past, interactive present, and seed for the future.
- Ordering ability: patterns that are random and three dimensional.

We can separate people into groups. Some people lean toward what Gragorc calls ***random*** and some toward ***sequential***. This is a continuum, with relative few at either extreme, but the extremes are useful for defining the continuum.

People who lean toward sequential timing see time in terms of discrete units, such as minutes, hours, days, and years. Time always moves forward from past to present to future. The clock never stops and never goes backwards. This pattern calls for step-by-step tasks that lead directly to the completion of a goal. People at the extreme of sequential are quite bothered when something is done in the wrong order. To them it destroys the whole process. They think you should take your time on each task to do it right the first time.

On the other side of the continuum there are people who see time as ***now***. Our graphic shows a stopwatch. If something does not work, stop the clock and try again. Doing it right the first time makes no sense. Until they have tried several times, they don't know what is ***right***. Rather than a linear progress of activities, the random style allows work in several dimensions at once. This may look random to a linear thinker, but to the random style person it is simply living in the ***now*** timeframe and working in several dimensions at once.

After many years of managing project teams I have seen more conflict resulting from differences in of timing than any other source. It is one of the fundamental reasons some people have a difficult time in either an agile or waterfall environment. One day I got a call from my son who was in his final semester of college. He said he was struggling with a required class. The class was analysis and design, which was part of his information systems major. I asked what the problem was. He said the professor wanted him to write the design before writing the software code. I stopped him right there and said that half of the experienced software developers I have known have the same problem. I tell them to write the code but don't tell anyone. After that, write the design, which is just a description of the code they wrote. The sequential thinkers cannot handle things done out of order. I suggested to may son that he write the code, but not tell the professor. Give her the design before showing the program. At a reasonable time later, present the program. Then, tell her how wonderful it was to have a design before writing the code.

People who live in ***now*** need to maximize ***now***, and that means jumping right in. They have no problem stopping the clock and starting over. An excellent software developer on one of my teams, Dave, often went through many iterations, starting over with different approaches until something worked. I respected Dave's approach, but when he worked on another development team that was much more structured and sequential, Dave was not respected and not well liked. Dave was my go-to person but he simply could not function in the other team's sequential environment. It was interesting that the other team was using the agile methodology, but they were anything but flexible, making them less than agile.

How do you do both? A better question would how do you accommodate both styles? Chances are, you have a significant number of both styles on your team. I lean toward the random side and I tend to frustrate the sequential style people. They work real hard to get each step exactly right, and then I do a restart which invalidates the work they have done. I learned that when I am working with a sequential worker, I need to do my homework and think through exactly what I want. When I work with the random workers, it is best if I speak of the end goal in general terms and leave them alone. From time to time I check in to see if they are delivering approximately what I want.

If you are a hard core agile project manager, you will probably frustrate the people who have a need to work sequentially. If you are a hard core waterfall project manager you will probably frustrate the people who have a need to be more random. You must do both if you want everyone to do what they do well. If you do one or the other, exclusively, you will be half as effective overall. It is like playing rock, paper, scissors and always being a rock or always being a scissors. This is the subtractive zero sum process rather than the additive non-zero sum process.

It is possible to do both even if your organization has a strong bias for one approach. I once worked for a very large financial company. Because we were managing other people's money, we had to do things right, all the time. The application development culture at the company was waterfall and most of the departments operated that way. The group I was with practiced waterfall project management, but did it in an agile way, long before agile practices became mainstream.

We practiced these agile principles even though we had never read the agile manifesto:

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**.

The other departments did annual planning, starting in September with business groups submitting project requests for the following year. The technology people did estimates and the management group set priorities. Senior management drew a line on the priority list and the items at the top became the measurable goals for the year. It was an orderly, step-by-step process and it was perfectly aligned with business priorities. The plan was formally updated quarterly, and a brand new plan created annually. Once a quarter all items that were scheduled for the quarterly release went into integrated testing for 30 days. No changes were allowed during this 30 day period without formal permission.

Our department did continuous planning. Business groups could submit requests at any time. There was a meeting every Friday when they could explain their priorities. Everything that went through the Friday meeting went into the request database. The request backlog was assumed to be multi-year. Some things might never be done, but they were worth keeping in sight. Every Tuesday, the technology management team met to review the request database. Some projects were in process and near completion, some projects were just starting, and most project just stayed on the list with no action. Every Thursday we went live with any projects that had the proper testing and business signoff.

Life changed when our group started reporting to the senior manager of the other groups. It was a painful transition to go to annual planning rather than continuous planning. We learned how to fake it and continue what we had been doing as much as possible within corporate standards. Fortunately we had negotiated with internal audit the minimum standards for project management and continued to meet minimum standards. We made no effort to be in compliance with mythical best practice standards. For us, good enough was good enough, and our real goal was to maintain our flexible, fast pace so we could keep our business customers happy by delivering now what they need now. This is an example of applying agile principles within a waterfall environment.

Sequential team - Random team

When we talk about teams it is useful to relate teamwork to sports teams. The problem with this is there are many types of sports teams and many types of teamwork. Consider a baseball team. At any one time, one person has the ball and the people on the rest of the team are standing in their various positions, watching and waiting. The other team has one person at bat, and the rest are sitting on the bench watching and waiting. Baseball is a slow game, but a very precise game with no room for error.

Consider a basketball team. One person has the ball, but the ball can be thrown to anyone who is available. Anyone can shoot for the basket at any time. Basketball is fast paced with plenty of opportunity for high scores. Speed and agility are the key to success.

A sequential team at work is like a baseball team. It can have workers in cubicles waiting for someone to throw work over the wall. It is essential that each team member do his or her job error free because each person does a specialized part of the total work. The project manager spends a great deal of time sequencing and scheduling work to assure the right resources are available at the right time. People do their assigned work and nothing more. They never step out of their respective roles. It is not important for them to like each other, nor

do they need to interact face-to-face, as long as the work gets done. The sequential team is well suited for waterfall projects.

A more random team at work is like a basketball team. Team members need to interact directly, preferably in the same space at the same time. The open office layout promotes this. Roles and responsibilities may be defined, but opportunities may arise for people to step out of their roles and do what needs to be done. The random team is well suited for agile projects.

Can we do the additive process and take the best of each team structure, regardless of project type? The Gallop organization did a survey to find out the effects of telecommuting on employee engagement, or the opposite, employee disengagement. Engaged employees tend to be more productive than disengaged employees. Gallop found that the maximum engagement and minimum disengagement happened when employees spent some time in the office and some time telecommuting. This provides evidence that we need some of both. Some of our work is sequential and can be done anywhere. Some of our work is more random and is best done face-to-face, interacting with the whole team.

Complete details - Big picture strategy

Agile and waterfall projects both require planning, but they require different types of planning. With waterfall there is detailed planning followed by signoff. The detailed plan is the primary monitoring and controlling mechanism. If it is in the plan it must happen and if it is not in the plan it must not happen. Agile projects have a general big picture plan, and a bunch of user stories. The user stories are on the wall or in a database, but they remain un-prioritized until it is time to prioritize them for the next work effort. At that point the top priority user stories are built out into greater detail and the work gets done.

Agile teams include the product owners who have great influence on the priority. The work team also helps sets the priority because some things are best accomplished before other things. There is a bit of chaos in the process, but from chaos, order can emerge.

Waterfall people might question the value of chaos and wonder if it can work. The big picture plan defines the overall project and the focus remains on accomplishing that objective. The example I gave from my experience with continuous planning demonstrates that it is an effective way of giving product owners what they need now, rather than what senior management decided they needed a year ago. If senior management sets an overall budget and overall time constraint, the team can figure out how to get there effectively.

Lessons learned - Retrospective

One of the final steps of a waterfall project is the lessons learned session. It is supposed to serve as a mechanism for making future projects more efficient. Often, it becomes a paper document that will never again see the light of day. The team is ready to move on and nothing from lessons learned will help the current project because it is too late for that. When I hear the term, “lessons learned,” I feel like I am back in school and the teacher is scolding us, “I hope you learned your lesson.”

In the agile world, there are many deliveries and many opportunities for learning. One of the agile principles is to take time at regular intervals to reflect on how things are going and look for opportunities to enhance

the team's effectiveness. This is commonly called a retrospective and is commonly not documented because it is a chance for the team to do some honest reflection rather than trying to look good for management. By removing the formality, it opens up discussion and problem solving.

Can any team do both? Every team should do both regardless of the type of project. The retrospective is for the benefit of the team and the lessons learned is for the benefit of the larger organization.

When we looked at Gragorc's definition of the random and sequential styles, we were focusing on how work gets done. Gregorc called this the *concrete/random* and *concrete/sequential* styles. He also looked at this continuum from an abstract, non-physical standpoint. How do people think, and how do people feel? This he called *abstract/random* and *abstract/sequential*. The *abstract/sequential style* people prefer logic, analysis, knowledge, facts, and documentation. In the *abstract/random* style people think in emotions, relationships, and memories. The highest priority for an agile team is to satisfy the customer. This is an emotional objective. In contrast, the waterfall effort at stakeholder management is a logical and rational objective with a focus on knowledge, facts and documentation.

User stories are preferred by the *abstract/random* style because of the customer focus. User stories allow room for emotion and happiness due to the flexibility and explanation on why something is needed.

Traditional requirements fit the needs of the *abstract/sequential* style. They focus on logic, facts and documentation, taking the emotions out.

Gregorc can be seen as a four-quadrant style profile, drawn here with the ***<u>concrete-abstract continuum</u>***, on the vertical axis and the ***<u>random-sequential continuum</u>*** on the horizontal axis. This should not be confused with other four-quadrant models that look at different behavioral characteristics. The thing to learn from Gregorc's model is that it is predictable that people will have preferences for the project methods that meet the needs of their styles. Some people will never be comfortable with methods that don't meet their needs.

Concrete Random	**Concrete Sequential**
Abstract Random	**Abstract Sequential**

Whether you go waterfall or agile, some people are going to have to compromise.

Concrete/Random

- Instinctive, intuitive, impulsive, independent.
- Now: total of the past, interactive present, and seed for the future.

Concrete/Sequential

- Instinctive, methodical, deliberate, structured.
- Discrete units of past, present, and future.

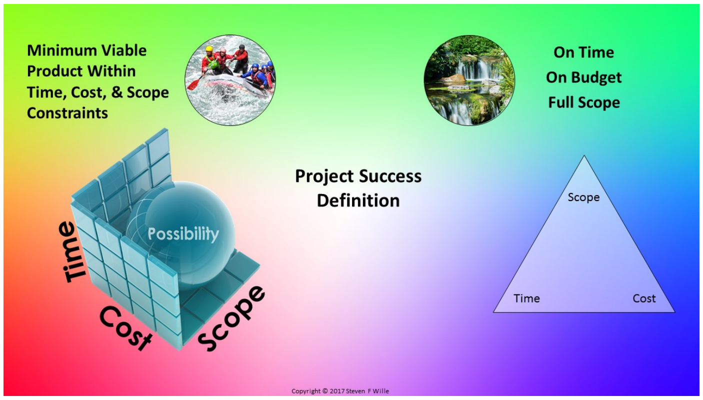
Abstract/Random

- Emotional, perceptive, critical.
- The moment, time is artificial and restrictive.

Abstract/Sequential

- Logical, analytical, rational, intellectual.
- Present, historical past, and projected future.

This takes us to the end. The goal of waterfall is to meet all three triple constraints. Achieving that is difficult. Consequently a large number of projects fail to be on time, on budget, with full scope. Agile takes a more realistic view by prioritizing scope rather than promising it all at once. Typically time and cost are locked in and the team delivers what it can within these constraints. It is important to deliver a minimum viable product early in the time schedule so that the customer has something

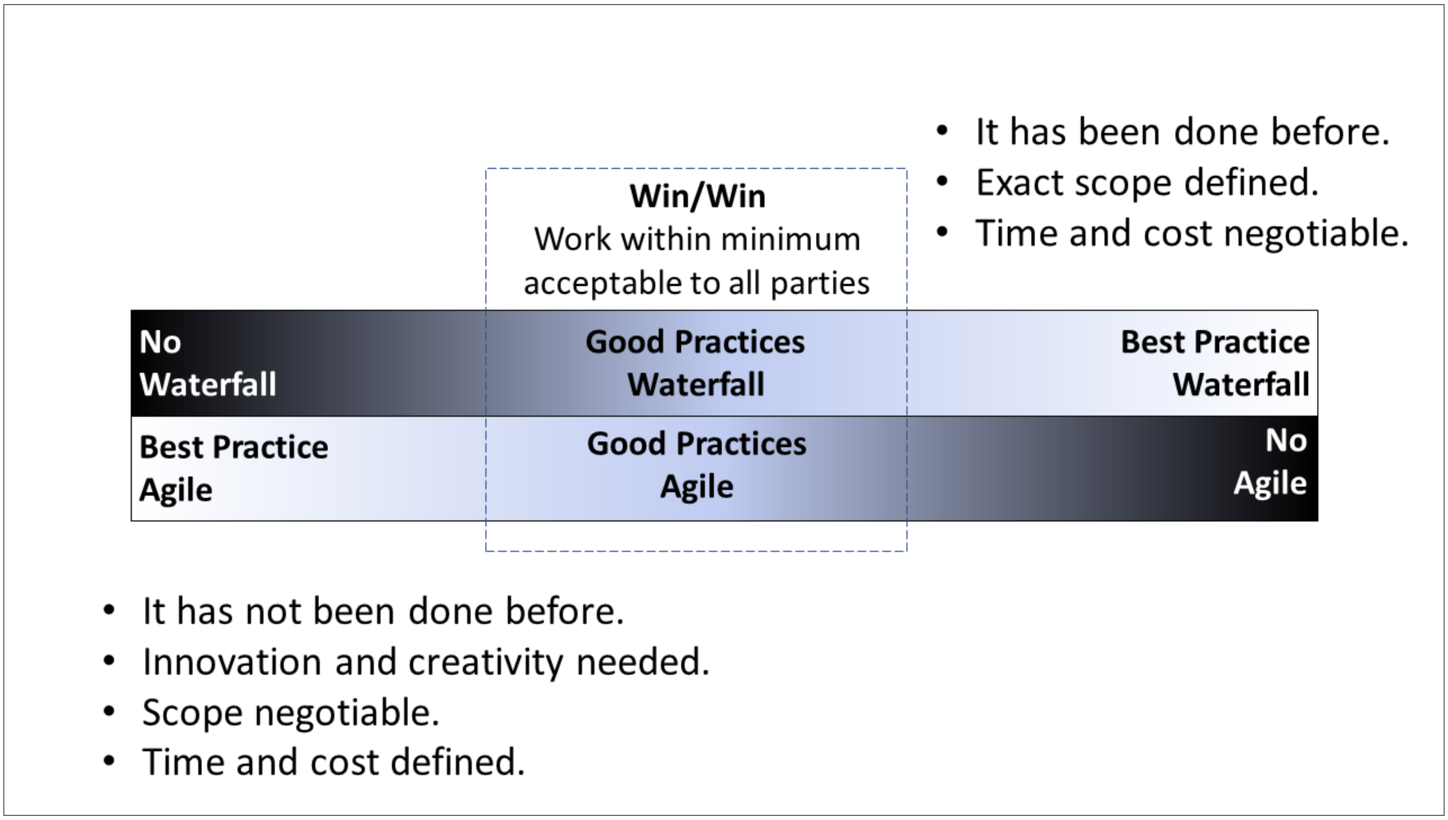


working in production, even if the rest cannot be completed. Once the minimum viable product is in place, the product owners can decide when to end the project. If they are satisfied with progress, they might keep the project going by giving it more time and more budget to get more scope.

It is decision time. What are you going to do? The zero-sum way is to follow best practices for your preferred method. The non-zero sum way is to learn from a negotiation method where you try to figure out what each party really wants, and what each party will minimally accept. This defines the range of acceptability, which is the area up for negotiation. It is also known as the area where there is a win/win solution. If you take a blended approach, you will not be following *best practices* for either approach. But you can follow *good practices* from each.

If you know what you want and it has been done before, then creativity and innovation are not needed or wanted. Best practice waterfall might be the best choice. On the other hand, if it is entirely new, innovation and creativity are essential so best practice agile might be the be the best choice. Most projects are somewhere in the middle.

As a final thought, go back to the first statement in the Agile Manifesto, *Individuals and interactions over processes and tools.* Regardless of your methodology, it is people who do the work. If you don't pay attention to the people, the rest does not matter.



Sources:

Sidhu, **Inder** (2010), ***Doing Both: How Cisco Captures Today's Profit and Drives Tomorrow's Growth***, FT Press

Lyons, **Dan** (2016), ***Disrupted, My Misadventure in the Start-Up Bubble***, Hachette Books

Gregorc, **Anthony** (1982), ***An Adult's Guide to Style***, Columbia, CT: Gregorc Associates, Inc.

Agile definition: https://www.merriam-webster.com/dictionary/agile

Agile Manifesto http://agilemanifesto.org/ © 2001, the above authors this declaration may be freely copied in any form, but only in its entirety through this notice.

Can People Collaborate While Working Remotely? Gallup Business Journal March 13, 2014

Coxswain definition: https://en.wikipedia.org/wiki/Coxswain_(rowing)

Requirements, **user stories, RACI**: International Institute of Business Analysis (IIBA), Business Analysis Body of Knowledge (BABOK) version 3.0 © 2015

Hubspot 2014 losses: http://venturebeat.com/2014/08/25/hubspot-files-to-go-public-with-a-100m-ipo/

Hubspot 2016 losses: http://amigobulls.com/stocks/HUBS/income-statement/quarterly

Waterfall definition: https://quizlet.com/38088105/kpmg-flash-cards/

PMBOK: ©2013 Project Management Institute, Inc. All rights reserved

Zero Sum definition: http://lawprofessors.typepad.com/environmental_law/2016/12/elc-essay-2-the-zen-of-nonzerosumness-in-environmental-policy-easier-to-say-than-to-achieve.html

Steve Wille is an experienced information technology executive having worked for large corporations in many roles including software development, project manger, director of application development, vice president of corporate information systems, and senior vice president of a business division.

His book, *Colorful Leadership*, focuses on three perspectives of project management, people, process, and innovation. He has written and facilitated workshops on project management, business analysis, high performance project teams, and constructive conflict.

Steve is active in the Colorado IT community as a board member of SIM (Society for Information Management, president of RMIMA (Rocky Mountain Information Management Association), and an active member of Mile High PMI (Project Management Institute). He serves on the Regis University IT advisory board. His MBA is from Regis University and his BSBA is from the University of Denver.

Michael Wille is an experienced developer and agile coach. His BS degree is from Colorado State University. After freelancing software development, he entered the corporate world and found his place in the space between traditional project management and agile project management. He is a Certified Scrum Master (CSM) and a Project Management Professional (PMP)