2013

Hybrid Agile

Bridging Old to New

Name Goes Here

ITSE-1391-001

# Table of Contents

Contents

[Table of Contents 1](#_Toc374565909)

[Introduction 2](#_Toc374565910)

[Waterfall Methodology 3](#_Toc374565911)

[Agile Methodology 4](#_Toc374565912)

[Agile Advantages 5](#_Toc374565913)

[Agile Disadvantages 6](#_Toc374565914)

[Toyota Production System (TPS) 6](#_Toc374565915)

[TPS Key Elements 7](#_Toc374565916)

[Jidoka 7](#_Toc374565917)

[Andon 7](#_Toc374565918)

[Kaizen 8](#_Toc374565919)

[Kanban 8](#_Toc374565920)

[The Common Thread of TPS and Agile 8](#_Toc374565921)

[Hybrid Agile Methodology 9](#_Toc374565922)

[Hybrid – Mixing New and Old and Finding the Happy Medium 10](#_Toc374565923)

[Hybrid Agile – Is It the Future? 11](#_Toc374565924)

[Example: Financing an Agile Project 13](#_Toc374565925)

[Real Options Theory 13](#_Toc374565926)

[Real Options Financing 13](#_Toc374565927)

[Conclusion 15](#_Toc374565928)

[Bibliography 16](#_Toc374565929)

Hybrid Agile

Bridging Old to New

# Introduction

Agile Development sprung from what many developers saw as an answer to the criticisms of Waterfall Development. As we learned in our coursework, Waterfall Development is a long, drawn out process of development that starts with a set of requirements and continues through a process of entrance and exit criteria, which in reality, could take a year or two to hit production release. By time of implementation, users often complain the newly delivered software doesn’t meet all their current needs or is no longer what they wanted. This often results in product cancellations or at best, a software product that users access some features and leave other features untouched. This results in a poor return on investment and eventually a loss of clients.

A group of seventeen people, all experienced in different programming methodologies, met together to discuss their ideas. From those meetings they formed the Agile Manifesto in February 2001. They found a commonality in a core set of ideas and experiences. (The Alliance: What is Agile?) They are the following:

* Close collaboration between the programming team and business experts
* Face to face communication over documentation
* Frequent delivery of deployable code
* Self-organizing teams
* Mitigate crises due to requirements changes

Software development companies quickly adopted these ideas and one simply has to search You Tube to watch Agile Methodology video lectures hosted by Google and other software companies. The move to Agile Development accelerated over the years as competition increased and businesses accrued the benefit of offering new features and enhancements more quickly to customers. This sounds like the perfect solution to Waterfall Development, doesn’t it? However, there is much more than just adopting a new methodology to development software.

Software developers frequently network with other developers or attend lectures. They quickly adapt to Agile Methodology and commonly plant the first Agile seed in their respective companies. Before long, software quality assurance teams join the effort and no longer become a Development Team effort, but an Engineering Team effort. This grabs the attention of company executives, always on the lookout to gain a competitive edge. This report discusses the Toyota Production System model that inspires companies to adapt the Agile Methodology companywide but also the challenges faced in moving to what is known as the Hybrid Agile Methodology. Utopia always seems to be out of reach and when a corporation has invested millions of dollars in tools and carries years of documentation, but how can it forego its past investments into process and jump into the river current that is Agile? Finally, this report will detail an example of using a different model that permits the CFO and his team to forecast the financial requirements for Agile projects and/or production.

# Waterfall Methodology

Waterfall, probably the very first software development methodology ever introduced, is defined as a linear- sequential life cycle model. Adopted by small companies and large enterprise-level companies alike, it had several distinct advantages and disadvantageous. (SDLC Waterfall Model)

Advantages

Very easy to understand

Each phase has deliverables

Each phase has a review

Phases are completed one at a time

Works well when requirements are understand

Disadvantages

After entry into test phase, not easy to go back and change

Software is not deliverable until complete

Not a good model when the project is overly complex

Not a good choice when requirements are likely to change

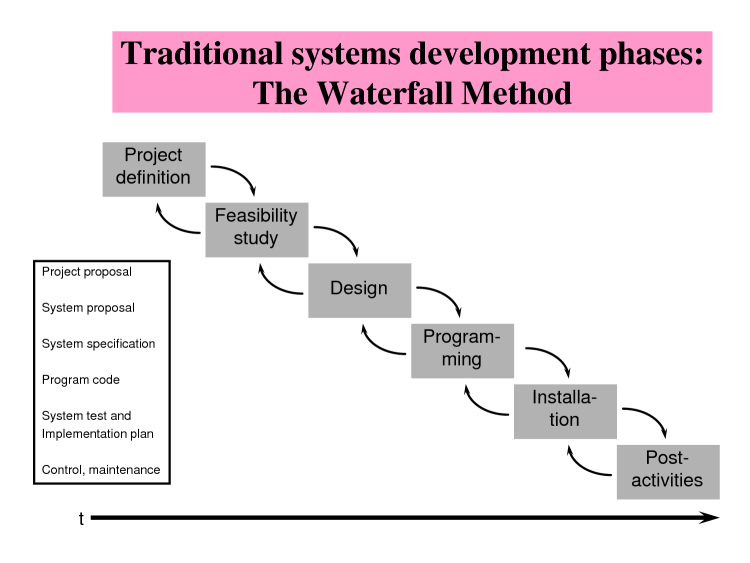


Figure 1 Classic Waterfall Process

One reason that Waterfall has fallen out of a favor is summarized in a table from a study by Dr. David F. Rico where he wanted to determine the value of Agile Project Management. See Figure 2 below:

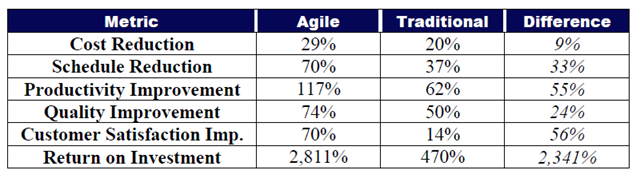


Figure 2 Metric Comparison of Agile and Waterfall

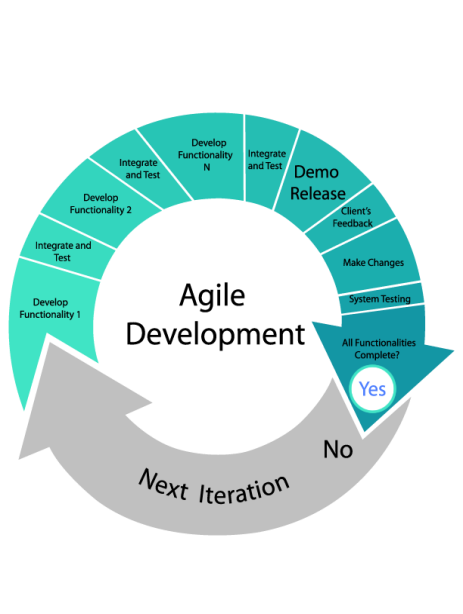
In the table, the ROI is a whopping 2341% bigger than in traditional Waterfall. (Rico, 2008)

From this data, it is clear that even though the CEO and the CFO may not quantify the difference, they have strong opinions on why they prefer Agile over Waterfall. This may explain why Waterfall has fallen out of favor so rapidly.

# Agile Methodology

The illustration in Figure 3 below succinctly shows a classic visual model for the Agile Development process as the Engineering team develops the functionality, integrates and test, develops the next bit of functionality, integrates and test, and so forth through iterations until all functionalities are complete.

Figure 3: Agile Development Cycle



There is always ongoing debate on what methodology is the best for any one company. You have Agile proponents who will claim that any company can adopt this methodology either 100% or close to 100% companywide, others who believe Agile is perfect only for their development and quality assurance teams, while others remain resolved to stay with the Waterfall approach, and others, who now find they can live more happily in a corporate world were a variation of Agile and other methodologies can live in harmony and even benefit from a hybrid approach and even argue this is the wave of the future.

## Agile Advantages

Agile provides many advantages, many which we enumerated in our class, mentioned in our introduction, and what I have personally observed in my place of employment. One cornerstone of Agile is the concept of teamwork. I have personally been a member of a team that has had a 180 degree shift in attitude since the adoption of Agile in early 2012. Prior to Agile, we used traditional Waterfall techniques and the relationship between Alchemy Systems Development and SQA departments was very testy and frictional. Most discussions were a lesson in intimidation and some members did not relish a problem cropping up because it generally involved invoking the blame game. Fast forward a year and a half to the present time and you will find those teams members now have greater respect for each other, understand better how we form our opinions around a problem, and any irritation generally involves a request for more input from a team member and not less.

To reiterate some key Agile advantages, here is a short list:

* Promotion of teamwork over individual effort
* Faster deployment of features to market
* Improved customer satisfaction
* Self-organizing teams – avoid sitting around waiting around to hold a meeting
* Document only what is necessary and reduce the amount of unused documentation
* Adaptability to changing requirements
* Continuous improvement and increased velocity as teams work more closely

## Agile Disadvantages

The adoption of Agile in the software development community has been amazing and often executives jump on the bandwagon because as they often say, it’s a ‘win-win’ for the company and it’s a ‘can’t lose proposition’. However, it does have some disadvantageous, some of which I’ve listed from my personal work experience. Any veteran of the software industry will understand what I have listed here and may have experienced more than one of these when undergoing a switch from Waterfall to Agile. The need to innovate never stops and some of these listed is what is driving Hybrid Agile methodologies. Here is a short list of the disadvantages:

* Requires close collaboration and may take up much time for contributors to the project
* Weak requirements introduce scope creek and render inaccurate project estimates
* New team members find it difficult to contribute to a project since there is little documentation and little time for training
* Higher cost to introduce testing throughout the project from start to finish
* The pace is relentless and a good and steady pace needs to be reached to avoid burnout
* Conflicting needs with those team members experienced in Waterfall
* Challenging to adapt to companies who provide software and custom hardware
* Agile easily adopted by Development and SQA, but often challenging to migrate Agile methods to other departments, such as Sales, Marketing, and Finance

The next section discusses the Toyota Production System, which is a precursor to Agile and demonstrates that the ideas embraced by Agile are not only suitable for a software development company.

# Toyota Production System (TPS)

Three gentlemen, Sakichi Toyoda, his son Kiichiro Toyoda, and engineer Taiichi Ohno developed the process we now know as the Toyota Production System (often abbreviated ‘TPS’). (wikipedia-unknown)

The core philosophy for the process revolves around continuous improvement, respect for people, long term view, the right process will produce the right results, add value by developing your people and partners, and the solving of root problems drives organizational learning.

Toyota used this process developed in the 20th century to propel their company to the number one automaker in the world and continues to hold this position. (Kubota, 2013)

In perusing the internet for information on TPS, interestingly enough, in Amazon.com customer reviews for books detailing TPS, current and former Toyota employees point to the recent poor quality to an abandonment of some of the principles developed more than 50 years ago.

## TPS Key Elements

Toyota pioneered lean manufacturing with a philosophy of the complete elimination of all waste. (Toyota u. )

### Jidoka

Toyota coined the term to mean “automation with a human touch”. (Toyota, Jidoka - Manufacturing high quality products)This concept is based on providing a quality product through a process where a machine detects a problem, which is seen as a deviation from the norm, work is stopped, a manager or supervisor removes the defect, and after the investigation is complete, a solution is introduced to improve the process.

Clearly, we see that the Toyoda family recognized the importance of delivering a quality product and changing/improving processes as needed. This parallels our introduction to writing an incident report that requires some investigation/review to determine if a defect exists.

### Andon

Andon is one of the main elements of Jidoka and pertains to a system or process where a worker alerts management, maintenance crews, and coworkers of a problem. (wikipedia, Andon)

In understanding Andon, it is interesting to realize the power of the factory worker to stop the assembly line for the purpose of resolving a problem. The historical method of auto manufacturing in the United States did not permit this prior to adoption of Japanese methods of manufacturing. Permitting the worker to stop production indicates the respect for the worker and the dedication to quality in the Toyota manufacturing process.

### Kaizen

Kaizen literally means ‘good change’. Per Toyota, it is the heart of TPS. Each Toyota employee obeys the guidelines but also looks for improvement of their work environment. This is not limited to manufacturing, but it is a concept taught companywide. (Toyota, Toyota Production System).

In many companies of today, it would be astonishing to think that a worker/employee comes in, does their job, clocks out, and goes home. In an Agile environment, we constantly look for ways to improve our product. However, we must also remember that in the 19th and 20th centuries, this was often a rare trait for a company worker to be actively involved in improvement processes.

### Kanban

Kanban literally means signboard or billboard. It is the underpinning of Toyota’s JIT (Just In Time) process. Taiichi Ono developed Kanban as a means to maintain quality and high production levels. Interestingly, the inspiration came from supermarkets and how they stocked shelves with food as needed, or just in time. Since the stock of food is assured, stabilizing the demand, supermarkets and stores could predict when to re-stock shelves. (wikipedia, Kanban)

Software development teams adopted this method in listing tasks, determining who is performing tasks, and when those tasks are complete. Searching the internet for ‘kanban tool’ provides several results, such as the following webpage: <http://kanbantool.com/>. Kanban moved from supermarket, to Toyota, to other production facilities, and also to software development. It has aided companies in streamlining their operations in order to be more efficient and profitable.

## The Common Thread of TPS and Agile

Looking at Figure 4 below, you will see that at the top of the structure, 2nd bullet, it shows that TPS is people-centered. If we tie Jidoka, Andon, Kaizen, and Kaban together, like Agile, the common thread is people and the respect given to people throughout the corporation and the respect given to customers. Also recall that a value of Agile is individuals and interactions over process and tools. (The Alliance: What is Agile?) So begins the journey down the path to Hybrid Agile Methodology.

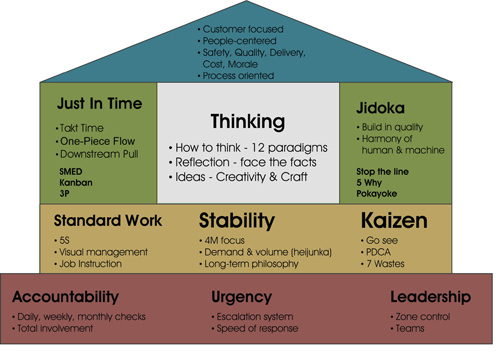


Figure 4 Toyota Production System

# 

# Hybrid Agile Methodology

Sitting in class this semester left me sometimes scratching my head and pondering how is it possible to accommodate test reviews, walk-throughs, formal reviews, informal reviews, requirements documentation and other forms of documentation while also keeping in step with Agile methodologies such as attending long sprint planning meetings, and then starting a sprint immediately and developing new test cases. How does one even start to plan all of this activity with just 3 or 4 team members to test? It seems like a movie titled “Clash of The Titans: Agile vs. Waterfall”.

In our early phase of using Agile, our project manager approached the entire team because he was frustrated because our velocity was too low. Development complained that SQA was documenting way too many incidents for little things like “text missing colon at end of caption” or “column formatted a bit too much to the left on report”. This ended in long exchanges of conversations within the incident report and it soon became obvious to my supervisor that we still followed Waterfall practices and had to understand that we could just walk over with our laptop to the desk of the developer and ask them to take a look at the problem and fix it on the spot. Soon we got the idea that it actually was easier to just walk over, talk about it, and eliminate the long documentation trails for the trivial issues. We went down the Agile path and we were sold on the benefits and our velocity increased. What could be better?

Unfortunately, directors and executives are used to getting reports on project progress. The questions will come in waves. “Have we hit critical checkpoint C?” “Would you say we are 30% done?” “Are we 50% done?” “Can I see the number of failed issues this sprint?” “I don’t understand the Redmine report, can you format it another way?”

Finally, further frustration sets in when the executive team celebrates the increase in productivity but runs into the issues of how we can change our whole organization to Agile. Then Finance requests a forecast on project dollars needed to finance all our sprints in 2014. How do you convert story points into expenses and tie them to a dollar amount? These are the questions that have managers working late into the night seeking answers.

Clearly, a manager and their team cannot simply tell the executives that Agile seeks to reduce documentation and that to take it on good faith that a sprint will be at least 90% complete by the end of the 2 or 3 week period with key functionality working. Agile teaches us people over process and it is necessary to review documentation and work out methods to streamline the process so as not to burden the team.

## Hybrid – Mixing New and Old and Finding the Happy Medium

In researching this paper, I find it intriguing that there is abundant information from CIO websites discussing topics of interest in their professional universe. One if these are the problems they face with maximizing the benefits of Agile. There are those companies that have converted 100% to Agile but find it doesn’t work as well as expected and then there are companies that find they just can’t seem to incorporate Agile into their process and find a division among company departments.

Of course, as companies stumble through this process, Agile experts offer services to train managers and employees and some companies have established training services for companies needing an Agile-like approach. One such company is SpecDD founded by Dr. Tieren Zhou who has created an idea of balancing Agile and integrating requirement and quality management with Agile practices.

In Figure 5 below, SpecDD illustrates their concept. They focus on providing companies the means to deliver better working software and also to create awareness and a process that aids a team in capturing process intelligence through improved communication and understanding of requirements. The key takeaway from the illustration is that any change to the software also requires change in the requirements documentation. Dr. Zhou posits that a failure to do so puts companies at risk in the long term because the meaning of the changes is lost. His process takes into account the discussions, the interactions, and the brainstorming that takes place during the product development cycle. (Zhou)

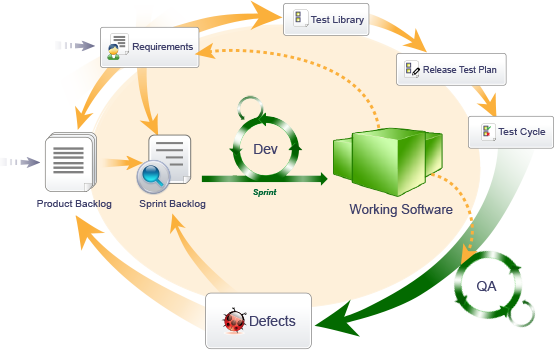


Figure 5 Example of Hybrid Agile

Does Dr. Zhou’s contention that companies lose process intelligence over the long term hold any water? Absolutely! In almost two years of Agile development, I can count numerous times that someone modified a requirement and several months later, no one could recall why it was decided to do so. A few team members will recall part of the conversation and then a feeling sinks in that we lost some vital bits of info so should we stick with the change or do we do our best to recall the original requirement? It’s a difficult choice that sometimes rears its ugly head during our demonstrations and the Product Manager’s eyes widen as he sees something different than she expected.

To resolve our problem of providing real-time information on our progress to our COO, we created a simple spreadsheet with our test cases where we mark “Pass”, “Fail”, or “Skipped” each time we run a particular test until the feature or features reach production. My manager, Irma, has created a dashboard feature on the spreadsheet to track number of cases tested, number of cases to test, number of cases passed, number of cases failed, and number of cases skipped. This is all done automatically via spreadsheet equations and this has satisfied the COO’s requirements and added little to no cost to our current testing operations.

## Hybrid Agile – Is It the Future?

In my continued search to find some bit of information on this topic, I stumbled across a very interesting article in CIO magazine. The author of the article, Michael Hugos, argues that Agile must move beyond development teams and prove that it can scale up on a much larger scale. The author sat down with Jim Highsmith, an expert in Agile and Adaptive Leadership. Jim strongly feels that Agile will become a mix of best practices from Extreme Programming (XP), Kanban, Scrum, and even from Waterfall. He has created a model that expresses his belief in what a company should embrace for the future. See figure 6 below. (Hugos, 2011)

**Layer 1: Technical Skills (XP)**

**Layer 2: Iteration Management (SCRUM)**

**Layer 3: Dynamic System Development Methodology (Products and Projects)**

**Layer 4: Portfolio Governance**

Figure 6

I found Jim’s comments in the article very fascinating because he said that teams start out wanting to be very agile and place themselves in a position to use only one type of Agile method but as they gain experience, they find that they have to adapt to each project and devise different methods. This appears to be a bit of what we are doing as time moves forward and we have now been through at least four different version releases. I work on one of our two teams and Irma has commented that the style and needs of the other team is a bit different from what we do.

Jim also concedes that in Agile workshops that he attends, developers relay to him how adoption of Agile at an enterprise level is sometimes difficult and there is a backlash towards Agile, especially with developers who are used to working a certain way, or other departments who do not want change, or the company has tools more suitable to Waterfall and don’t want to abandon those tools.

The author returns to inform the reader that tools are under development to form a bridge between tools geared towards Waterfall and newer tools geared toward Agile. Such tools are needed to help organizations accept Agile on the enterprise level. He gives a wonderful analogy regarding bridging the new and the old. Architects and civil engineers still use forms such as arches and domes that were devised centuries ago because they work and still give value. By the same token, if a methodology continues to give good value, it can be maintained within the Agile process and bring success to a company. However, he feels that Agile will be at the center of this success. He warns about treating Agile as a dogma and holding on to the requirement of the methodology and to be open to maintaining existing methods while trying some variation of Agile that will give continued value and improvement to the company or organization.

# Example: Financing an Agile Project

The smooth adaption to Agile takes a great deal of effort and as mentioned before, reaping the benefits of Agile is one thing but going through the heavy lifting is where the real sweat starts. Irma, my manager, asked me to attend a meeting on her behalf. In the end, it was a very painful experience. The new owners want to know how much money to invest into our new projects for 2014. Will we need to hire more employees? Will we need contract employees with certain expertise levels? Will we need to contract programming services when we don’t have the expertise to do it in house? Do we need new tools? Do we need new hardware? Do we need new software? In most cases, the answer is “Yes”, but how do we put dollar figures on that when we are an Agile Engineering team and we only have estimated story points as a basis? Irma explained my specific assignment was to ensure that when I reviewed the upcoming projects that I give input whether or not the estimated story points seemed reasonable or not and to stick to my guns if I felt strongly that we needed to adjust the story points for each future project. Of course, I did just that and fought hard at one point to the consternation of the CTO. I wondered how many of the other companies performed this exercise and how was it possible. By fortune, I discovered that Toyota, Google, and Microsoft all use this method for financing their development projects. Here is a simple explanation to a very complex process for financing projects in an Agile environment.

## Real Options Theory

Real Options Theory is derived from the theory of dealing in Stock Options. Rather than having only two options to buy or sell, there is a third option and that is to do nothing and delay a decision. Likewise in financing a project, a company actually has three decisions it can make. It can decide to finance the project and move forward, it can decide to cancel the project outright, or it can decide to do nothing for the moment. Now the third option doesn’t mean putting off a decision, but is a calculation to wait and then a decision must be made in the future. Microsoft is classic example of waiting to make a decision. They will see how the market is behaving towards a new product and if they feel the project is worthwhile, they will decide to finance the project and they move forward. (Matts & Maassen, 2007)

## Real Options Financing

I found one blog with an opinion written by an IT manager which fits in with the ideas above. His point is: “Break functionality into MoSCoW, must have, should have, could have, and would be nice (but won’t have). One source used the term MVP, minimum viable product. “Must have” represents the minimum viable product.” The author of the blog, Jim Stewart, also states: “Express these in terms of business need, not in terms of what will be delivered. (Traditionally, we’ve focused on being exact about what would be delivered.) With Agile (realize the potential beyond just Agile) you would be clear about the objective, not how it would be delivered.” (Stewart, 2013)

Matts and Maasen also state that financing an Agile project is more a state of psychology that it is of math. They state that Real Options is much more effective if all parties deal with each other honestly and give each other options, rather than hiding intent. Cooperation is the key. (Matts & Maassen, 2007)

There is a common thread here and it is a reliance on cooperation and communication. Does this sound familiar? Again, it comes back to what is referenced in texts on the Toyota Production System and Agile. Both place a high value on respect of the individual, to listen, to understand, and to cooperate.

On a more practical level, it is easy to find guidance on conversation of story points to a dollar amount. One can calculate Cost per Point or CPP with the following equation (Fowler, 2011):

I also encountered a wonderful illustration graphing how a company can realize a benefit from a project before the project is finished and compares both Waterfall and Agile. See Figure 7 below. (Hamilton-Whitaker, 2009)

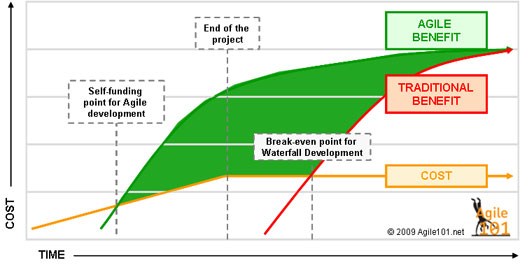


Figure 7 - Waterfall vs. Agile ROI

Quantifying future value, estimation of story points, and cost per point are quite feasible and as time goes on, more and more information becomes available to assist all those who dabble in Agile development. Returning back to my participation in the company exercise, I and the others who participated believe we did the appropriate job in revising our story point estimates. But the key takeaway from this example is that one can crunch the numbers but in the end, it is communicating, conveying, adapting, and eventually gaining insight and understanding on how Engineering and Finance can work in a synchronous state despite the differences in philosophy. Hybrid Agile may indeed be the next step in software development evolution.

# Conclusion

Hybrid Agile is still a new concept and performing a quick search in Amazon.com doesn’t provide any significant results on the topic. But searching the internet provides many results and the amount of information is too much to put into one 20 page report.

One key takeaway from this report is that for a company venturing out of Waterfall software development, the price may be extremely high to abandon tools and acquire new ones. Product Managers, Project Managers, and Engineering Managers should not be fearful of continuing to use their old tools and methods. CFOs should not be fearful in devising new methods of forecasting investment dollars needed to fund Agile projects. Jim Highsmith coined the term ‘prescriptive Agile’ to describe those organizations that feel they have to go 100% Agile. That is not so and our own teams at Alchemy Systems adapted tools to be more Agile friendly and this has worked without the need to discard time and money invested.

The other key takeaway from covering Agile and TPS is that a company must realize that one of its greatest resources is its employees. With continuous improvement and teamwork, the work environment improves, productivity improves, teams become cross-functional and self-mange, and eventually revenue increases. Before my own eyes I have seen the improvement in teamwork, morale, and respect among team members, knowledge, and productivity. The company expects huge growth in 2014 and one can give a tip of the hat to our switch to Agile.

Alchemy Systems, like many other companies, must embrace Agile not only in the Engineering Department, but also by other departments. It often takes great patience to communicate Agile ideas to others, but we are now far down the Agile path and it is working. The key here is to continue finding the happy medium and working on a hybrid version of Agile that meets our needs. Recall that one key Agile tenet is people over process and as long as a company respects that, will prove successful.

# Bibliography

Fowler, M. (2011, April 7). *Questioning Story Points*. Retrieved December 10, 2013, from Dzone Cloud Zone: http://agile.dzone.com/news/questioning-story-points

Hamilton-Whitaker, T. (2009, July 22). *Self-Funding Projects - A Benefit of Agile Software Development*. Retrieved December 7, 2013, from Financial Planning: http://agile101.net/2009/07/22/self-funding-projects-a-benefit-of-agile-software-development/

Hugos, M. (2011, March 28). *Hybrid Agile – Moving Agile Development to the Next Level*. Retrieved December 2, 2013, from CIO: http://blogs.cio.com/michael\_hugos/15587/hybrid\_agile\_moving\_agile\_development\_to\_the\_next\_level

Kubota, Y. (2013, October 28). *Toyota keeps top spot in auto sales rankings, outselling GM, VW*. Retrieved December 6, 2013, from Reuters: http://www.reuters.com/article/2013/10/28/us-global-autos-toyota-idUSBRE99R04K20131028

Matts, C., & Maassen, O. (2007, June 8). *"Real Options" Underlie Agile Practices*. Retrieved December 4, 2013, from InfoQ: http://www.infoq.com/articles/real-options-enhance-agility

Rico, D. D. (2008). *Rico APM ROI pdf.* Retrieved December 10, 2013, from David F Rico: http://davidfrico.com/rico-apm-roi.pdf

*SDLC Waterfall Model*. (n.d.). Retrieved December 10, 2013, from SDLC: http://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm

Stewart, J. (2013, September 2). *Funding Agile Projects*. Retrieved December 10, 2013, from Just My Opinion: http://www.stewart13.com/

*The Alliance: What is Agile?* (n.d.). Retrieved December 7, 2013, from Agile Alliance: http://www.agilealliance.org/the-alliance/what-is-agile/

Toyota. (n.d.). *Jidoka - Manufacturing high quality products*. Retrieved December 5, 2013, from Toyota: http://www.toyota-global.com/company/vision\_philosophy/toyota\_production\_system/jidoka.html

Toyota. (n.d.). *Toyota Production System*. Retrieved December 8, 2013, from Toyota Australia: http://www.toyota.com.au/toyota/company/operations/toyota-production-system

Toyota, u. (n.d.). *Toyota Production System*. Retrieved December 5, 2013, from Toyota Company website: http://www.toyota-global.com/company/vision\_philosophy/toyota\_production\_system/

wikipedia, u. (n.d.). *Andon*. Retrieved December 4, 2013, from wikipedia: http://en.wikipedia.org/wiki/Andon\_(manufacturing)

wikipedia, u. (n.d.). *Kanban*. Retrieved December 6, 2016, from wikipedia: http://en.wikipedia.org/wiki/Kanban

wikipedia-unknown. (n.d.). *Toyota Production System*. Retrieved December 6, 2013, from wikipedia: http://en.wikipedia.org/wiki/Toyota\_Production\_System

Zhou, D. T. (n.d.). *SpecDD Introduction*. Retrieved November 23, 2013, from SpecDD: http://specdd.com/view/1ywxz1x120x/SpecDD-Introduction-