

The Agile Marathon

Bob Schatz
Solstice Software
bschatz@solsticesoftware.com

Ibrahim Abdelshafi
Primavera Systems
iabdelshafi@primavera.com

1. Abstract

Many companies today are facing the challenges of moving their project teams to use agile development methods. Much of the discussion these days centers on dealing with the issues and hurdles that must be cleared for teams to achieve success on their agile projects. This paper is about our experience in what we're calling the Agile Marathon: The long-term view of how to maintain success after your team begins to see that they have achieved the benefits of using agile techniques.

2. Introduction

Our focus and experience is in commercial product software development. We hope to shed some light on the challenge of keeping Agile success going over a long period of time. We have faced the challenge to keep our teams motivated and focused on releases lasting over 18 months; the same product, same people, just new releases. Once the team recognizes its success, a new phase begins where it becomes difficult to hold everything in place knowing you've got a long road to travel.

The benefits of creating a smooth development cycle to maintain a steady pace can sometimes be more of a challenge than other traditional methods. Often, the typical highs and lows of a project can be used to give a project team the motivation to change. Adrenaline levels rise as teams enter short periods of heightened awareness of what they need to do to gain acceptance. Agile projects attempt to create a smooth consistent work cycle and minimize the peaks and valleys that many of us have become accustomed to. This consistency reduces the team's need to adjust their work patterns. This static pace presents a challenge for managers to keep the work interesting without the hero culture.

3. Background

When we began our journey into Agile four years ago, we did so because our organization was

experiencing quite a bit of pain from running our projects using traditional methods. We had times of extreme stress where we worked a lot of overtime for months on end. It was classic firefighting management, heavily weighted to a command-and-control structure. People were exhausted, unhappy, and making more mistakes than they were correcting. Once we finally did complete a project nobody really felt great about what we had done, just relieved that it was over. There were always questions about whether we were really done, or had just run out of gas. Although we had a strong focus on quality, it was more the latter. We had our celebration, handed the "gold" master CD over to our product managers and tried to forget that we would probably repeat this horror many times over in the future.

We finally did reach our limit and after looking at alternatives, we made the leap of faith to use Scrum [1] to produce the next release of our product. We faced the challenges of a complex set of new requirements and nobody really had a good handle on what was actually needed. We did know that is was going to be another long release cycle of about 18 months; and we had to try a new approach to have any chance of delivering. Well, it worked! Scrum was a big success in our organization. We committed ourselves to all of the Agile principles and kept improving sprint after sprint. We created a great agile success story and we all felt great about it. It was a huge win for us all, personally and professionally. Our continual improvement process worked great. We didn't get everything right, but we did recognize when things didn't go as planned and made quick adjustments. In the early times when improvements were made, we saw immediate positive results. That feedback motivated us to make more changes.

Eventually, in a situation like this where the organization remains relatively constant, the product is mature, and the process is consistently working well, there's not much more you can do. From a management perspective, this is a good place to be. The stabilization represented what we had set out to do, although it had now become almost too predictable, a little boring you might say. The new challenge in the successful agile organization is to

continually find ways to keep the team motivated and optimize the capability that has been established.

In this paper we'll provide a picture of the benefits that we gained from establishing an organization powered by Agile methods, the challenges we faced in maintaining our edge, and then provide some recommendations for others looking to establish a long-term commitment to using Agile methods.

4. Benefits

The benefits that remained are very much in line with the promise of Agile. They have helped us build a professional software development organization that is focused on delivering high-value products to our customer community.

3.1. Work-Life Balance

One of the main reasons that we took the plunge into Agile was the concern over burnout of our staff. Our traditional method wore people down to the core. The result was not better products with more features. It was more sick days, weight gain from eating the pizza we kept buying, stress, personal problems due to too much time at work, and the ever-increasing mountain of defects we would deal with at the end of projects. Our success with Agile delivered 40-hour work weeks, an 80% reduction in defects, a 30% increase in delivered functionality, satisfied product owners and customers, and the satisfaction of knowing that this would continue as long as we kept following the process. We have delivered multiple high-quality, feature-rich releases since 2003 with no overtime, and no weekend work. I can't speak for the weight issues, but we all look a little healthier. Sick time is down significantly, people are spending time with their families, getting married, having kids, and bringing their great stories into work. People are proud of their work, and they experienced a positive change in their lives.

3.2. Quality Products

The quality of our products in terms of both pre- and post-release defects, as well as user satisfaction with delivered functionality increased through our use of Agile methods. The short increments, goal-driven development, and collaborative nature of Agile kept everyone focused towards achieving success with each iteration. We maintained a high quality level by setting increasing test automation goals, and imposing limits on defects with which teams could go to a sprint review.

3.3. Repeatable Process

We made a commitment to follow the Scrum process and the principles of the Agile Manifesto [2]. With each sprint we made minor adjustments to accommodate the needs of our organization, but stuck to the core process. This gave us a consistent, repeatable, documented process that was shared by everyone involved in product development. We did not have a requirement or need to be certified by any outside organization. However, in order to check on our own progress, we contracted a software process auditor to examine the artifacts of our process to see how we measured to guidelines such as the FDA 21 CFR Part 11 [3] and SEI CMM [4]. We were very pleased to hear that the team's execution of our process was a model for what these guidelines had intended.

3.4. Strong Development Team

The strength of our development team evolved as we adopted Agile. The leaders spent a lot of their time focused on the team dynamics; helping their people work through the changes, supporting them through the change process. We went from cubicles to team rooms. We worked with every group of developers to help them become a *team*. Some people did not make the journey with us. The weak members surfaced quickly and either by their choosing or ours did not stay in the organization. As people got to know each other on a personal level and discovered how to best leverage individual strengths, the team became stronger through the trust they built. Through coaching and practice, the teams discovered methods to deal with conflict and come to quick resolution of issues for the good of the team. The effort we put into this built a positive high-energy work environment that became a model for others adopting agile techniques. We had many other companies come to our facility to see what a "healthy" environment looked like. The visitors were able to talk with anyone in the organization and hear a consistent success story [5].

3.5. Measurable Progress

We started our agile journey with Scrum and later began incorporating Extreme Programming (XP) practices. We primarily focused on Test-Driven Development including automated unit testing and continuous integration and testing. With Scrum and XP we were finally able to gather data on our development efforts and improve our ability to estimate and set achievable goals. Working in short iterations, setting team goals, and gathering performance data that helped with future sprints

provided us with the basis for improved prediction for what could be delivered by the team. We were diligent in posting daily burndown data for each team and maintained a public area where anyone could come in anytime and see what each team was doing. Daily visibility gave the leaders an indication of the health of each team in terms of their progress and adherence to the concept of self-management. We could quickly see which teams needed some coaching. We used this data in a positive way, not as a way to measure teams against each other or an indicator of success or failure.

3.6. Early Problem Resolution

From the very first days of using Agile methods we were challenged with resolving issues quickly. In the beginning this was extremely stressful. So many issues surfaced, we sometimes wanted to go back to the way we were doing things so we could bury our heads in the sand at least for a little while. We stuck to the process however and just did our best to resolve issues as they came up. Issues surfaced on the people, process, and product fronts. You never knew where the next one was coming from, but you knew that they would be there, and they needed to be dealt with quickly. Over time, we built up our ability to tackle problems with relative ease. It felt good to clear the roadblocks and get everyone moving. When we began to focus on finding solutions rather than someone to blame, resolving the issues became much easier. We worked with the leaders in the organization, teaching them how to deal with conflict, and how to coach teams to deal with conflict. This skill helped us move faster as an organization.

3.7. Flexibility

One of the most important aspects of Agile is the ability to build flexibility into the development cycle. The history of software development project failures over the years has resulted in mistrust between developers, managers, customers, support and other organizations involved in the creation and delivery of a software product. Since software development is a very creative endeavor, it is difficult to manage in a predictable manner. With projects that have a lot of ambiguity, using a process which gets everyone involved to accept that there will be significant changes improves the chances of success. In the process a great deal of collaboration, trade-off decisions, and experimentation takes place. This flexibility ultimately helps shape and mold the product into something that is focused on solving the problem it was intended for in the first place. This flexibility is not easy to establish. There are many organizational and political hurdles that must be cleared to get this right. Nothing will test the fabric of the organization

more than a process that forces decisions to be made based on business value, technological feasibility, and customer needs, downplaying individual interests, appearances and agendas. When the whole organization adopts the principles of Agile the flexibility becomes a very powerful tool that can be used to gain a competitive advantage.

4. Challenges

Along with the enduring benefits of Agility, we also faced many challenges during our journey. These challenges have involved all roles within the agile process; the development team, the product owners, and the stakeholders. The following section highlights some of these challenges.

4.1. Development Team

The challenges that a development team can experience change over time as it adopts Agility. During our first year of moving to Agile, we experienced a lot of growing pains. We faced the challenges as a team and worked hard to make continuous improvements. After the 2nd year of experience the team settled into the process and became comfortable. We were very proud of our accomplishments and started to share our success story with others in the software community. After that initial celebration we started asking ourselves: This was really exciting, now what do we do? What is the next big thing? When the goal of establishing a balanced and consistent workload is met, the team can become “bored” with the process. Everyone feels a need to continue a pace of rapid changes. The team can begin to slip back to old practices. Without it being everyone’s main focus, it becomes easy to lose sight of the Agile principles that brought success in the first place. In essence, you can become a victim of your own success.

There is a sense of discipline that must be continually reinforced in the organization. There are very few rules within Agile, but the ones that are there have to be followed to the fullest. Rules such as, “You should always be at most 30 days away from being able to release” or “Test first before you write any code” are there for a reason. Following these rules “most” of the time does not work, and can get you in trouble quickly. The challenge is to maintain this sense of discipline and the commitment to follow it.

One of the strengths of Agile lies in flattening the organizational hierarchy and eliminating unnecessary layers of management. The cross-functional, whole team environment encourages a team to work together, as one unit, to complete pieces of functionality. While this helps us be more productive, it creates little

“perceived” career growth for the team members. They start asking themselves: “What is my career path in the agile environment? How can I grow professionally? Am I a tester, a programmer, maybe a business analyst or a little of everything? How does my new diverse skill-set fit in to my functional career path?” Managers and leaders have a similar set of challenges in trying to redefine their roles and find ways to support the team.

Continuous improvement lies at the heart of Agility. When we started our journey, we created a culture that embraced change and encouraged developers to initiate and push forward process improvements. The ones that worked we made part of our process, while the ones that did not, we changed once again. The impact of these process improvements was clearly evident at the beginning, and that fueled the energy and excitement of the teams. Over time, that impact became less evident and the benefits were less quantifiable. With the big issues resolved and positive progress made, after a few years the retrospectives produced suggestions with a diminishing rate of return. Continuous improvements are helpful, as long as they add value. Otherwise you can find yourself making changes just for changes’ sake, and you can quickly end up creating unnecessary layers of bureaucracy in your process.

4.2. Product Owners

One of the aspects of Agile is its recognition that software development is not deterministic in nature, making it very difficult to predict a detailed schedule into the future. Agile also recognizes that change is an inherent part of building software and thus promotes adaptation in order to be both flexible and responsive to change. This flexibility provides a big benefit to product owners, as it enables them to revisit and adjust priorities on a regular basis. They gain the advantage of seeing the system grow with each iteration, and can make course corrections based on feedback.

We’ve seen two big challenges in the relationship between the developers and the product owners. In the early adoption phase of Agile, there is skepticism about what is going to change on both sides of the fence. Product owners like the idea of flexibility, but they also have to make plans based on product roadmaps. These plans play an important role in the corporate planning process as revenues are projected based on new products and features. Once the product owners begin to work better with the teams as a partner in the process, the relationship shows dramatic improvement. As the maturity of using the process sets in, the pressure comes to “lock down” what will be in a release, so the customer facing folks can begin preparing launch materials. Thus flexibility becomes something that has to be managed carefully in the

agile process from both the product owner’s and development team perspective. The second challenge comes as a result of this pressure where the agility becomes a “weapon” used by the development team to divert some responsibility to provide good estimates and accurate progress tracking. The response from product owners comes in the form of frustration with the perception that they see as a lack of commitment on the team’s part.

Another area that has presented some challenges is the Sprint Review. The Sprint Review is an ideal opportunity for product owners to see the finished work and provide feedback to the development team. At the start, the product owners were excited about seeing working software. After experiencing about 6-8 months of iterations, they began to express some hesitation about giving feedback. At the Sprint Review it seems too late to give meaningful feedback and then have it incorporated into the product without having to “pay a price” for adding it to the backlog. Implementing any recommendations would require rework and take away from other work that needs to be done. Instead, the product owners would prefer seeing a detailed spec before implementation begins, so they can provide feedback before the work begins. To the development team that translates to the message that they want to go back to Waterfall’s up-front requirements phase.

4.3. Stakeholders

Sprint Reviews are an integral part of Scrum. For teams, the Review presents them with the opportunity to showcase their work and interact directly with the stakeholders. To stakeholders, the Review provides a unique opportunity to observe the functionality as it is being implemented, and provide feedback that can be incorporated into future Sprints. The Review also enables the stakeholders to interact directly with the development team, an opportunity they would otherwise rarely have.

After the initial enthusiasm wears off, and you have anywhere between 12-24 sprint reviews in a year, stakeholders tend to get bored with some aspects of the process. While important for the release, some of the accomplishments presented by the teams are not always of interest to the stakeholders. Stakeholders also have a tough time keeping track of the cumulative additions to the product deliverable. With many teams this can get very confusing. The stakeholders may miss Sprint Reviews or attend only part of it. The Agile techniques provide guidance for how to make a point with the stakeholders, which is clear on a shorter term project. However, on these longer term cycles it is understandable how this event can slip down on their priority list of things to do. The development team is impacted because they perceive this as a lack

of interest in the work they are doing. The result is a loss of some of the enthusiasm and excitement they get out of the Agile process to feed the next iteration.

5. Recommendations

As the issues surfaced over a long period of time, we tried a number of approaches. Some worked for us, and some worked against us. But with Agile, we always had the mechanism to stop and try something different to improve it. Most of the approaches that didn't work could be summarized as the attempts by the managers to "over-engineer" the solution instead of presenting it to the team(s) to solve. We saw this in our behavior, and we've seen others heading down the same path. It's truly a learning experience.

The following are some of the recommendations we would have for others who are facing or will eventually face some of these same issues as they achieve success with Agile practices.

5.1. Establish a Strong Foundation

5.1.1. Strong leadership from Senior Management.

It cannot be said enough how important this is. We believe that Agile works best out in the open but it can be out there only if there is strong support from Senior Management. In our case we had a VP of Development who not only gave us the means to go Agile but also the independence to learn from our mistakes, and shielded the Development organization in its transition. The senior leaders must also work with other organizations to help them understand what is changing in development and how these organizations can benefit from the changes.

Leaders of organizational change initiatives must develop skills to recognize when to shift gears between **leading** and **managing**. When there is ambiguity and chaos, leadership is needed. When there is an established process that needs to be monitored and optimized, management is needed. Once a process becomes set and repeatable, the management of that process is critical in communicating the progress, monitoring project metrics, and keeping senior management informed of team performance. Leadership is a critical skill when there is not a clear path. When there is a new process or technique being tried it is important for the leadership to guide and support the team.

5.1.2. Frequent coaching and retrospectives. Good change agents can help an organization recognize that a transition will bring many changes, and changes mean that people will experience discomfort. By supporting each other, the changes will be much easier. We were fortunate to have senior managers

doing an excellent coaching job, as well as bringing in recognized, experienced Agile advisors from the outside. Someone with name recognition or demonstrated success will significantly reduce some of the early resistance in the team. Every organization has to adapt Agile to suit its own culture and requirements. This means learning from mistakes and not getting disheartened or reverting to old ways. The teams sometimes tended to look to the managers for guidance and decisions. This was clearly discouraged by the VP of Development so that the teams learned how to resolve issues, and managers learned how to not instinctively jump in. This was a behavior change that needed to be recognized as a step backward first. Next to strong leadership this was the most important feature for our successful transition.

5.1.3. Some "cleaning the stables" of people who could not adapt.

Not everyone managed the transition well and very early on a few people who were set in their ways and became obstacles to moving forward were coached to help them in their transition. An important point to make is that everyone adapts to change at a different pace. In the Geoffrey Moore model [6] some people are early adopters and some are laggards. Again, the senior leadership helped try and coach these people first, and then supported them in finding other opportunities either in the company, or outside. However, over 95% of the people stayed on for the wonderful ride ahead. You should never underestimate the power of even a single person regardless of position to make your transition a success or failure.

5.1.4. Leadership training for all team members.

Agile makes demands on every team member to take on a leadership role. Leadership comes in many flavors, but it's important to view leadership in an agile context not as a position a person has, but the actions they take in helping to move their team in a positive direction. In times of change, leaders emerge throughout the organization. When the cross-functional teams were created the traditional reporting structures no longer would work since the functional leads were not necessarily in the same team as the team members. As a result of this and for a larger goal of growing the organization, leadership classes were initiated in house for all team members. Everyone began with establishing knowledge and understanding of "self" and then we explored topics in communication, power & politics, motivation and rewards, vision development, and team dynamics. We learned and applied our learning with each iteration. This experience helped everyone find their strength and contribute to the efforts. Managers learned how to

be better at coaching and establishing excellence in their specific disciplines.

5.1.5 Adopting the Agile language. One of the most important aspects of a culture change is the adoption of a new “language”. Some Agile practices have interesting names that some organizations would prefer to substitute with their own terms that “fit” better in their previous culture. We think that if you do that, you can expect that some changes won’t seem to “stick”. Words like Scrum, Sprint, Planning Game, chickens, pigs, burndown, stories, are important to driving a change in the way people think. If you choose to substitute your traditional terms, you can expect traditional results.

This language also must carry on to helping drive team accountability and responsibility. Getting people away from non-committal terms to terms where they take responsibility helps establish the discipline in the culture. Dropping the word "might" in favor of "will" when planning or meeting as a group. Stopping people from using the word “hope” when letting others know when they will complete something. Establishing confidence and taking responsibility to be held accountable for getting a task done is tough, but delivers tremendous results. It must also be coupled with a mechanism to seek and offer help with your teammates. Asking for help early is a behavior to be rewarded. This can be made a fun transition by using the \$1 fee as a fine for using the old words, and recognizing when someone uses the new words.

5.1.6. Leads and Managers Use New Skills. Establishing self-managed teams and getting them to work together is certainly a challenge. You count on your leadership to help get people through the change process. Our VP established a systematic program to develop the skills and behaviors that we needed to not only make an adoption of Agile a success, but any future changes as well. It cannot be assumed that people who are good at managing will naturally be suited to leading. There must also be a realization that the managers and leads in the organization will, like everyone else, begin to question the changes in roles with the move to Agile. If you’re a manager in an organization it probably took a lot of sweat to get there, then someone springs self-managing teams on you and that has to put some questions in your mind. So particular attention must be paid to helping the people you count on to lead the change, learn new skills that will answer the questions about their role in the organization as it changes. Quite often in our transition, a number of approaches that didn’t work out could be attributed to managers trying to “manage” when they should have been coaching.

Helping teams recognize and learn from mistakes quickly and being strong mentors is the goal for these folks in the agile organization.

5.1.7. Programmers/Testers/Business Analysts learning to function as one unit. Even when team members learned to work in a collocated environment, there was resistance to let go of cross-functional boundaries. Early on, there was also a tendency among the teams to finish features without really completing them so as to appear as over-achievers at the Sprint review. We also had a tendency to over-engineer a product even after agreeing upon the estimates during the sprint plan with the product owner. The project managers spent a lot of time with the teams, teaching them how to work together as one unit. This was accomplished by coaching, team building exercises, and a lot of learning experiences over many sprints.

5.2. Teams

From the perspective of the teams, we did a number of things to keep the environment interesting. By giving some attention to this on a regular basis, you keep the concept of adapting to change at the forefront of everyone’s thinking. It had the effect of keeping minds sharp and always looking for ways to improve.

One of the things we did was to encourage people to work on different areas of the product. It takes some commitment to not always think of who is the “expert” in a particular area and give them the work because they know it best. You have to plan for some work taking a little longer, but the long-term benefits are outstanding as the flexibility of the organization grows.

As the teams got used to working together and the Agile process is second nature, we began the practice of rotating people and making new teams, rotating the Scrum Master role on the team, and changing the team room locations. Again, allowing for adjustment periods, supporting the teams, coaching them, and watching them adapt. This “stirring” usually took place after putting out a product release. In some cases this resulted in disbanding a high-performance team, but overall the practice helped keep our teams well-balanced and productive.

We encouraged everyone in the organization to extend their influence and help others in the software field learn to become successful with Agile. Allowing time for people to develop proposals for conference presentations, writing papers and articles, and hosting visitors from other companies to show them how we achieve success with Agile. We’ve challenged people

to continuously seek ways to improve and never tie them to boundaries. Everyone is encouraged to contribute new ideas and experiences to the Agile community.

5.3. Scrum Master

It is worth calling out the practice of the rotating Scrum Master here. One of the behaviors we noticed with teams is that the Scrum Master role often was treated as a position of power in the team. The intended role is someone who is coaching and guiding the team during each iteration, ensuring that the progress data is collected, making sure the planning meeting, daily scrum, review and retrospective take place, and also removing obstacles that the team identifies. With time, the team can begin to treat this position as their manager. And the person in that position typically detects and dutifully fills the apparent need. The result is a breakdown in the team's self-management practice. By rotating the responsibility at the start of each iteration, it diffuses the role and makes it a shared team responsibility and establishes a balance of power.

As your organization matures in its use of agile techniques you may find in necessary to shift back somewhat to having a stronger Scrum Master role when the focus turns more towards the management of the process and team metrics.

5.4. Product Owners

For Scrum to be successful, it is imperative to have the product owners established as partners in the development process. To reach this objective, the teams need to listen to the product owners, understand their concerns with Agile, and ensure their concerns get addressed. The teams need to be educated on how to have discussions about business value and to make that their main focus. In our case, the main concerns were related to the lack of predictability and the absence of upfront detailed specs.

When probing the product owners for the reasons they wanted upfront detailed specs, we discovered what they really wanted was to have an upfront visual of the major features being developed and how these features would interact with the rest of the system. To address that concern, along with their desire to have better estimates, we established the practice of using "Sprint 0".

Sprint 0 is a mini-sprint, at the beginning of a release cycle, focused on reducing the ambiguity surrounding risky or complex requirements on the release backlog. In our case, we spent Sprint 0 working on a simple wire frame of the new functionality on the release backlog, and did further analysis on some of the risky requirements on the

backlog, adjusting their weighted estimates accordingly. The key to the success of Sprint 0 was not to make it too long, so as not to turn it into a mini-waterfall process. The outputs of Sprint 0 were a wire frame of the new functionality, and better estimates for the complex features on the backlog. This is the only Sprint during the development cycle where teams are permitted to use wire frames and slides at the Sprint Review. By giving the product owners what they really wanted all along, we created the partnership that is necessary for agile to be successful. Adapting Scrum to the needs of the business is what makes this process so powerful. Each project that uses Agile techniques should assess the needs of the team and the product owners at the beginning of the project to determine if Sprint 0 is needed.

5.5. Stakeholders

It is important to keep the stakeholders engaged in the Sprint Review process for many reasons, not the least of which is keeping the enthusiasm and energy level of the Scrum teams at a high level. To keep our stakeholders engaged, we started holding Sprint Review retrospectives with them to find out how we can maximize the value they derive from the reviews. In response to their feedback, we made many adjustments to the format. Again, changing it and trying different ways of presenting kept things interesting for everyone.

Stakeholders were sometimes reluctant to provide negative feedback to the teams, so we arranged a meeting, immediately following every review, where the stakeholders could provide their feedback directly to the product owners. In addition, we now provide them with a comment sheet to provide anonymous feedback.

Stakeholders also expressed their frustration for not knowing what was being done with the feedback they provide at every Sprint Review. In response, we created a Wiki-site where all the feedback from the Sprint Review was gathered and within a week of the review, the product owners would update the site with the action taken based on the feedback provided. By visiting the site, stakeholders would know whether their feedback would be incorporated in the current Sprint, added to the Release Backlog, or put on the Product Backlog.

6. Summary

As organizations around the world gain experience in their adoption of Agile, they will experience many changes. There will be a mix of positive and negative experiences, hopefully more positive than negative! The benefits of Agile are there

for the teams that commit to making them a reality. The Agile journey is a long one and there is no end. Once you gain the advantages, your challenge is to keep the process going. It is important to continually highlight the positive outcomes and not become complacent with success. In a sense, the worst enemy of long-term Agile success is the process itself. Leaders must find other ways to motivate their teams on a consistent basis. They can no longer rely on the firefighting atmosphere where heroics are counted on to get software releases out the door.

We have achieved great success with Agile in our organizations and we are running in the Agile Marathon every day. It takes a tremendous amount of energy and concentration to keep a successful implementation of Agile going over a long period of time. Almost four years of attending daily scrums, sprint planning sessions, sprint reviews, and sprint retrospectives give us the satisfaction of knowing that we have tackled a great number of challenges, made many mistakes that we learned from, and now have found ways to pace our teams to ensure our success over a long period of time. It is our hope that you use our experience as a tool to facilitate your own learning experience.

7. References

- [1] Schwaber, Ken, *Agile Project Management with Scrum*, 2004, ISBN 0-7356-1993-X
- [2] <http://www.agilemanifesto.org/>
- [3] www.fda.gov/ora/compliance_ref/part11

[4] www.sei.cmu.edu/cmm

[5] www.primavera.com/newsroom/articles/PrimaveraGetsAgile_052005.pdf

[6] Moore, Geoffrey A., *Crossing the Chasm*, Harper Business, 1991, ISBN 0-88730-717-5

8. About the Authors

Bob Schatz is the VP and Chief Development Officer for Solstice Software. Bob is responsible for leading the development of the Solstice enterprise integration testing products. Prior to this, Bob Schatz served as VP of Development for Primavera Systems, Inc. Bob has 24 years experience in the development of large enterprise systems. He holds a bachelor's degree in Computer Science from Temple University and is currently pursuing a Masters degree in Organizational Dynamics from the University of Pennsylvania. Bob is a leader in successfully implementing agile development techniques, such as Scrum and XP, and driving culture changes in organizations. Bob often speaks at industry events talking about the benefits and challenges of bringing agile techniques into an organization

Ibrahim Abdelshafi is the Vice President of Development for Primavera Systems. He's responsible for the development of Primavera's world-class project and portfolio management solutions supporting organizations in every industry around the world. He received his MS from Carnegie Mellon University and his MBA from the Wharton School at the University of Pennsylvania.