This white paper explains how an organization decided to move to the Scrum framework for delivering software, and launched 18 Scrum teams over a six week period. Although there is a tremendous amount of detail that cannot be expressed in a short paper, there is enough detail to understand the difficulties that may be encountered when implementing a new process. These difficulties were magnified because of simultaneous rapid organizational growth.

In summary, the ability to deploy a new software development methodology while simultaneously adding large numbers of developers was possible because of the following three major contributors:

- The CEO and CIO, by providing direct support as follows:
  - frequent corporate communication
  - active participation in a newly formed steering committee
  - delegation of authority to the implementation managers (the following two areas below)
- The Performance Improvement Team, taking responsibility for the following:
  - establishing and communicating a timeline of organizational deliverables
  - setting procedural standards under the new methodology
  - delivering training to all levels in the organization
- The Application Development executive and QA executive working together to:
  - create a new rapid acquisition and training plan for staff augmentation
  - develop a new, simple resource management tool to account for frequent changes to staffing levels and assignments
  - enforce the new methodology

Although this is a true story, out of respect for confidentiality the name of the organization has been omitted. The company in question will be referred to as ABC Systems.

The generic term "developer" used in this article refers to both application software developers and quality assurance testers.

Planning For the Year's Work

ABC Systems is a company that provides services to customers primarily through web-based applications. Although ABC's customers are quite varied, many of them are looking for applications that will support their holiday season business. Because of this, requests for custom application development begin arriving in large numbers as the summer months approach in anticipation of having project work completed before Thanksgiving.

Coming into the Spring of 2004, ABC Systems was faced with having to plan the project work expected for the remainder of the year. Historical project request trends coupled with a large backlog of important internal projects revealed that the software development needs for 2004 would far surpass any previous year's efforts.

Planning meetings that lasted for hours at a time over many weeks were established in an attempt to plan out the flow of work expected. We had 18 projects that had to be started in the June – July timeframe, followed by other projects that could begin later, but all must be completed before the holiday season. No matter how the project backlog was re-organized or re-sorted, the end result was always the same: if ABC wanted to complete the internal initiatives *and* satisfy customer requests, a significant increase in IT staff would be required.

Problem 1: A History of Failure

ABC had accumulated an unfortunate history of failed projects. As is the case with many organizations, not all projects were failures, but enough had seen problems that caused many customers to loose faith in our ability to deliver quality software on time and on budget. Unknown to us, at least one of our customers resorted to internally padding our estimated delivery dates because of the numerous times we had failed to deliver on time.

Why did we have such a high rate of disappointment? There were numerous reasons, stemming from a lack of well-defined and adhered-to project management processes. This resulted in poor client communication, unclear requirements, a lack of collaboration between IT and the business owners representing our customers, and unrealistic estimating and scheduling. Again, we had our successes, but it doesn't take many failures to lose a customer's confidence.

We were, however, in a great market position, as our competition did not have as diverse a product offering. They simply could not effectively compete against us…yet. We knew that this year would be our last great opportunity to prove to our customers that we could indeed deliver as expected. If not, our competition would have an equally great opportunity to catch up with us and threaten our customer base.

Problem 2: The Need to Expand

Because of the seasonal nature of our business, we had little interest in pursuing new development hires for a temporary rush of work. This would leave us with unwanted excess development capacity at year end. We needed to consider a new model that would utilize contract developers.

ABC had used contractors before, but those experiences had been generally unfavorable. Regardless of the quality of the contractor, we were just not organized in a manner that

could take advantage of contract help effectively. If we were going to use contractors, we must implement a new way of incorporating them into our projects quickly and effectively.

There were other problems, too. For example, our budget wouldn't support the amount of work expected, regardless of the revenue opportunities. Our facilities couldn't accommodate the number of people required. Supporting staff from the Data Center, Database Teams, Project Management, etc., could not accommodate such project and staff growth without themselves also growing. This was not just an application development issue, but an event that would have far reaching effects throughout the company.

Something Must Change

After weeks of planning, and as we were about to enter the summer months and face this large amount of work, we knew that if we did not do something different we would fail. The comical definition of insanity was ringing in my ears: "Q: What's the definition of insanity? A: Doing the same thing over and over again and expecting different results."

Enter our President and CEO. He had been reading about agile software development, particularly Scrum, and circulated an overview article on the subject. He then called several key employees into his office, one at a time, to get their opinions on the outlook for the coming workload , and what they thought of Scrum. There appeared to be a consistent response: first, our projects will fail if we don't change, and second, Scrum would be a good method to consider.

In early June our CEO gave the word to proceed with a pilot implementation of five Scrum projects. Our Performance Improvement Team began reading though every book and article written on the subject. We didn't have much time, as we needed to begin many of our projects in the June – July timeframe; we were already late!

The Performance Improvement Team quickly developed training and documentation material, which included new corporate standards that would be required for any new Scrum project. These new standards were necessary to ensure we would not launch a Scrum project if there was a high risk of failure. Some examples of basic elements required before launching a Scrum project included:

- The project must have a healthy Product Backlog, which includes estimates and priorities
- The key roles of ScrumMaster, Team Members, and Product Owner must be explicitly defined for the project
- All Team Members expected to work on a Scrum project must be adequately trained in the process

They also developed a timeline of events that must take place in order for us to launch successfully, including numerous training sessions.

Preparing for Launch

While Performance Improvement was busy organizing around the procedural details of a pilot launch, Application Development and QA needed to organize around the necessary activities that would allow us to build five pilot teams of developers. These teams would operate with an average of five developers per team, and many team members would need to

be taken from their current work assignments and environment for placement on new the new Scrum teams.

Scrum expects that team members are dedicated to their project assignment, but ABC Systems had not operated this way. Most developers were working on numerous activities simultaneously. A large re-assignment undertaking also needed large amounts of communication to the developers so they could understand why we were going through so much change, and why certain individuals were being carved out for these special Scrum teams. Feedback from the developers revealed that some were jealous that they were not being selected to move to Scrum. The primary appeal was the ability to work on one project rather than daily being pulled in multiple directions. The developers began to talk about the future day when they all could concentrate on one task at a time!

As we considered the difficulty facing developers who are trying to be productive while working on numerous activities, we recognized that there are two primary types of work that keeps developers from focusing on their assignments: bugs, and small requests. Bugs, of course, arise unexpectedly and disrupt our plans, and small requests are submitted with an expectation that they will be turned around quickly because of their small effort. In both cases, there's an understandable expectation that the work will be addressed sooner rather than later, leading us to interrupt whatever major project work is in progress.

One of the "safety nets" that we employed to protect the larger project work was the use of two development teams that were dedicated to work only on bugs and small requests. With these two teams focused on those interruptive activities, the project team members could focus on their assignment with greater consistency.

Explosive Growth

As we worked feverishly towards launching the pilot Scrum teams, we also had to address the need for additional development resources. Management worked on assessing the financial impact of augmenting the development staff in order to deliver on the 2004 project plan, and proceeded with the approval to bring in more help.

We selected several outsourcing companies to help us with providing the additional developers. We intentionally limited the number of companies we would work with, expecting this would minimize duplicate candidates and also decrease vendor administration. We narrowed our choices to three companies that provided resources from three different markets: one company specialized in the local market, another in the national market, and the third in providing developers from India. With access to such a diverse pool of candidates, we could ensure a steady stream of available candidates. Because of our tight timeline, we created new hiring and testing criteria to help filter qualified candidates early in the process and prevent undo evaluations that would most likely end in rejection.

To indoctrinate the new developers, we created a fast track in-house technical overview training program that all newcomers were required to attend. We then assigned several of our more experienced developers with the responsibility of mentoring the new developers. Additionally, the senior technical architects created "office hours" on a public calendar at our request. These office hours were reserved times where the architects would make

themselves available for several hours a week to any developer who had questions or problems. These enhancements, coupled with the steep learning curve of the highly interactive Scrum environment, eventually allowed us to see productivity from new developers in just weeks rather than months.

We also had an interest in changing the role of the QA team members. QA was mostly represented by individuals who were experienced *user* testers. If we were to create truly cross-functional self-managed teams, we would need to move towards building a QA department with *technical* testers, individuals who could work collaboratively with application developers in a highly supportive and interactive manner, possessing knowledge of and experience in application development. The new hiring efforts would also account for this migration to a new kind of QA team member.

Although we didn't realize this at the time, we would grow from 25 to over 120 developers in a three month period. As our numbers grew and teams formed, we needed to quickly respond to changes in our resource plan. We accomplished this through the most simple of methods. We created a "wall of resources," which was an entire office wall that held color-coded sticky notes. Each note represented a project, an application developer, a QA tester, a Product Owner, or a Scrum Master. The sticky notes were arranged on the wall with each project as a header for a vertical line of notes, one for each team representative. For easy reference, different colors were used for skill and employment status (full time or contract). Using this method, we could quickly and easily move or add resources as needed. Eventually we converted this to a spreadsheet, but during the rapidly changing implementation period, a manual tool was the quickest and easiest way to maintain our records.

*The Wall of Resources*

Management Oversight

As part of our implementation plan, we created a Scrum oversight committee, affectionately termed the "Super Scrum." This committee was vital to our success, as it included representatives from key areas of our company, including our CEO. The CEO's active participation in this committee showed the rest of the company that he was serious and committed about this effort, and was determined to see it work. The responsibility of this

committee was to review our implementation progress, but also to act as an escalation point for any issues that were preventing teams from making progress. The success of our Scrum launch was important enough to warrant the Super Scrum meeting every morning for an hour to review our progress.

This committee also discussed at length the methods used for selecting ScrumMasters. Since we had not yet worked under Scrum, we were unsure of the qualities of a good ScrumMaster. Rather than define a particular existing position from which we would select ScrumMasters, we decided to select candidates from a variety of areas: Project Management, Application Development, QA, and even Finance and Risk. Our CEO promoted the idea that becoming a ScrumMaster should be something that is strived for, something that would incite individuals to eagerly volunteer for the role. While not all ScrumMasters were able to adapt to their new role, we did create a solid base of ScrumMasters who refined the art of Scrum and loved what they did.

Both the CEO and CIO trusted those of us who were responsible for the implementation of Scrum, and delegated authority to us for all things Scrum. This allowed us to proceed rapidly, not needing to receive approval for all matters. Of course, since we were meeting daily, there was always an opportunity to discuss our latest plans.

We set our start date at the third week of June, and when the time came we pulled the trigger and launched our five pilot Scrum teams.

Launch!

Part of the pilot expectations included a close monitoring of those teams to ensure compliance with Scrum methods, as well as compliance with ABC Systems standards. Between Performance Improvement and IT, we shared the responsibility of reviewing team progress through interviews, observations, and team-building activities.

These teams were operating for less than two weeks when our CEO brought us together to discuss our findings so far. Everything was going well, the new teams were adapting well to the new expectations, and they were making progress on the projects assigned to them. We expected to allow the five teams to run for several more weeks, after which we could make a full assessment of their productivity and decide which teams could launch next, if any.

The problem was that we didn't have enough time to wait for a full pilot evaluation. We were already at the beginning of July, and the project backlog was beginning to grow as we waited for projects to start. Our CEO was insightful enough to recognize this. He could see that we were in a do-or-die situation. If we waited for the pilot review, we simply would not have enough time left in the year to get all the work done. If we did nothing, we would fail based on our track record of poor project delivery. He then pulled the next trigger: forego the pilot assessment, and launch all remaining projects on Scrum. We were shocked, but quickly understood the reasoning behind this decision.

Launch…Again!

We now had 13 additional project teams that needed to launch quickly. In addition to items noted above (budgets, facilities, etc.), several important activities needed to continue in greater force:

- Additional hiring and training of staff
- Scrum training for all team members
- Selection and training of new ScrumMasters
- Monitoring of teams to ensure consistency and productivity

While this situation was developing, our customers needed to understand what was changing at ABC Systems. Our in-house training was attended by not only IT developers and ScrumMasters, but also by our business representatives so they could communicate to their customers effectively. We would also occasionally have clients visit our offices, where we would take the opportunity to explain our new development methods. These explanations included the expectation that the customer must be involved, contributing to the needs of the project and reviewing output on a regular basis. Most of our customers were intrigued at the opportunity to be more involved with the project teams.

One of the contributions ABC Systems made to the Scrum community was the consistent use of shorter Sprints of two weeks as opposed to the more common use of four weeks. Although many experimented with shorter Sprints, this was our defined standard. We decided on two weeks because of the particular market conditions and customer base at ABC. The products and services offered by ABC were relatively new, and therefore were prone to many changes as the market developed. Our customers were seeking to exploit opportunities in a highly competitive environment, and two weeks would allow them to change plans in short order.

Over the next four weeks we launched the remaining 13 project teams on Scrum: 18 Scrum teams in six weeks!

Cultural changes can be expected under Scrum, and we began to see some negative effects of these changes. Some employees were not adapting well, particularly those who were not part of a Scrum team, and therefore didn't have experience with the inner workings of Scrum. Those with strong tendencies towards control had a very difficult time accepting the changes towards self-management, and some needed to leave the company. Others, however, recognized this as one of the best experiences they had been exposed to in project management and application development.

Positive Feedback and Growth

Feedback from our customers was consistently positive. While some were still unsure of the methods, they all were pleased with the turnaround that was taking place at ABC. In at least one case, a customer asked us to slow down, as we were developing product faster than they could absorb!

Our CEO created a new concept for our customers, that of "buying" a Scrum development team. Rather than using the traditional work order contracts that define project expectations with a fixed cost / fixed delivery date clause, we now were offering customers the opportunity to "pay as you go" for development resources at the team level. Interested

customers would pay for the efforts of a development team to work on whatever priorities they deemed important, a Scrum principle that maximizes return on investment. The agreements allowed the customer to end their work requests with 30 days notice, but they could also continue extending the agreement indefinitely. Many customers signed up for a three month trial period, and proceeded to extend their dedicated teams well into 2005 and even 2006.

We continued to add projects as the months proceeded. Our successes lead us to expand Scrum into other areas, such as the Data Center and Network Operations. We also experimented with one-week Sprints in areas that had an unusually high degree of change. During the following year, we hit a new high of 32 concurrently operating Scrum teams.

Our CEO ensured that the entire organization understand our movement to Scrum. At quarterly company meetings there would typically be a presentation regarding the current state of Scrum affairs, allowing even those furthest from application development to understand how we managed software projects at ABC Systems. This also was a great opportunity to reinforce our intentions with those who may have still had doubts about our resolve.

Although the end result at ABC was extremely positive, the "last resort" scramble put tremendous stress on the whole organization. We could not have succeeded without creatively thinking through the many new problems that faced us. The "inspect and adapt" methods of Scrum resonated deeply with those of us who helped orchestrate this massive change. New problems would arise every day, and these needed to be addressed promptly to ensure productivity, but that's part of the process.

How Did We Do It?

It's difficult to note one particular reason why we were so successful. Some that stand out, however, are the following:

- a strong commitment from the leader of the company
- a dedicated implementation team who would not accept defeat
- the Super Scrum oversight committee that not only provided leadership and direction, but moral support for every employee
- direct, open communication across all management lines

Scrum is indeed simple and can be implemented in a day. ABC Systems proved that much more can be accomplished with just a bit more time than a day.

The author is Peter Borsella, representing the application developers. The following are those who contributed to the success of the events shared in this article:

Larence Park
Shashi Kapur
Tim Dorsey (currently with The Dorsey Group LLC)
Dora Swartz
Bryan Stallings

Peter's initial detailed exposure to Scrum came from "Agile Software Development With Scrum," by Ken Schwaber and Mike Beedle.

Peter Borsella is the founder of Winnow Management Corporation, based in Parkland, Florida. Peter is an IT professional with 22 years of experience in application development and IT leadership. He is also a Project Management Professional (certified by the PMI) and a Certified ScrumMaster Trainer (certified by the Scrum Alliance). His ability to contribute across a wide range of environments has taken him to companies such as First Data Corporation and eFunds Corporation and outside the United States to India and Hungary.

Peter is adept in administering all aspects of project management, as well as providing authentic leadership to create effective teams. An active and contributing member to both the PMI and the Scrum Alliance, Peter's speaking engagements focus on bridging the gap between traditional Project Management and Agile Project Management. His goal is to help others understand how to get the best return from any project by achieving higher levels of software quality, increased customer satisfaction, and cohesive teams that enjoy increased productivity.