Nathan Quirk  
Article Review 2  
“Primavera Gets Agile: A Successful Transition to Agile Development” – Bob Schatz and Ibrahim Abdelshafi

In “Primavera Gets Agile: A Successful Transition to Agile Development,” Bob Schatz and Ibrahim Abdelshafi write about their company’s transition from a waterfall software process to an agile framework, specifically Scrum. Their company, Primavera, somewhat ironically developed project management software for the construction and engineering fields. Primavera’s developers often worked long hours with dramatic levels of overtime to meet objectives that were not always needed or wanted by the customer. Primavera made some initial steps to change the culture and methodologies, but after a particularly disastrous release, they decided more drastic changes were needed.

Their solution, Scrum, involved implementing self-managed teams – each with their own “product owner” – that would work in *sprints* to complete well-defined goals. At the end of a sprint, teams present their work to everyone – project managers, other teams, customers, where constructive criticism can shape the next sprint and refine project requirements.

Primavera was fortunate to enlist the help of Ken Schwaber, creator of Scrum, who helped guide them in their transition. Schwaber provided a fresh pair of unbiased eyes. Also important was the full cooperation of the executive side of the company. Their support made it clear to the Primavera team that this new direction had the full weight and backing of the company leadership behind it.

Initially, the development team was reluctant to adopt a new way of doing things, but several things aided their journey. Some highlights from this list for me were: **Focus on teamwork** – while this is easier said than done, putting in the effort to foster teamwork paid off for Primavera. **Use the established agile language** – initially the team was tempted to use their in-house language to conduct business, but the authors discourage this, as it leads to old thinking and potentially lapses into old behavior. **Don’t work overtime** – once a staple of their work diet, overtime at Primavera became seen as offering diminished returns and they successfully avoided it.

For their first Scrum-based project, Primavera carefully documented the results using their own proprietary software. At the beginning of each sprint, each developer team would start with a feature from the production backlog, and team members then broke that requirement into multiple 8 to 16-hour tasks. At the end of each day, progress would be noted and used during the next day’s *stand-up meeting*.

When Primavera’s product finally reached completion, they had some pretty impressive statistics. Quality as measured by errors per one-thousand lines of code went from .51 defects to .36 – a 30% increase in overall quality. In addition to heightened customer satisfaction, the authors noted that this decrease in defects allowed the team to move more quickly into a new project instead of fixing the last release. Primavera also saw improvements in their time-to-market metric, as they were able to deliver four months early.

The authors also detail some non-tangible benefits of the change to the team. Specifically, the team never worked nights or weekends. They were also able to work outside their traditional sphere of responsibility when they needed to help other members of their team reach a sprint goal. This enhanced the overall feeling of cohesion amongst the developers. Additionally, because of Scrum’s insistence on the integration of customer feedback, developers got a much better appreciation for what they were building and how it would eventually be implemented by the end user. This is turn gave them more satisfaction from completing something they knew would aid their clients. The company had a 100% retention rate during this time.

Primavera did note that there were some drawbacks at first. Scrum did seem to place too much emphasis on bringing “short-term-deliverables” to the table at the end of every sprint, they did find that not enough attention had been placed on less interesting infrastructure needs. This iterative release schedule also obfuscated the overall completion rate from the clients, who could see work being done but not how much progress was being made in the overall picture.

The authors notes thought that none of these drawbacks were deal-breakers, so to speak. Scrum had proven itself to them as the way of the future. They conclude by highlighting even more shakeups to their old model by incorporating some Extreme programing practices such as pair-programming and test-driven development.