



# How Agile are YOU?

**Agile project management uses ideas from lean manufacturing practices to aid software development**

**D**espite living in a world that to outsiders appears full of rules and constraints, software developers often see themselves as creative souls, champing at the bit to produce innovative, leading edge solutions for business and the consumer.

The worst scenario for a software project would be an unholy alliance between the 'it would be nice to have, could you just add...' business client and the unfettered 'it will only take a second to add that feature' software developer.

The tension between project teams delivering solutions and customers wanting instant answers is perhaps more dangerous in organisations delivering internal solutions than those providing consumer and business software. At least in the latter the sales, development and marketing teams should, in theory, all have one goal in mind, whereas internal business teams can often have multiple, sometimes conflicting, agendas.

But creating new products and upgrades for software has its own problems. Marketing and sales demand features that are perceived to give commercial advantage and they want them now, or by Q3 at the latest, and definitely before a competitor gets there first.

So how does one of the world's leading project management developers, Primavera, approach these problems to ensure that releases are timely and meet everyone's needs?

They have had a good reputation for producing reliable, efficient software ever since

Joel Koppleman and Dick Faris created the company in May 1983. It is still a privately owned company and has had 21 consecutive years of growth and profitability. More than \$4.5 trillion in project value is managed by Primavera software.

The company survived the temptations and the trauma of the dot.com boom and bust and has retained an enviable reputation, particularly in construction and engineering and now increasingly in areas like IT, telecoms and pharmaceuticals.

Agile work management is a radical shift from traditional project management priorities. Instead of calling for a project manager to predict, plan and assign work to individuals, track its completion and make any necessary adjustments on the way, Agile work management follows modern lean manufacturing practices and engineered process controls in complex development. Limited-size, self-managing, cross-functional teams apply their skills and technology to fulfil a pre-determined requirement and show working functionality to the customer at the end of each specified work period. The project manager is responsible for maximising productivity – similar to a coach for a sport's team.

The 450-member staff is based across the US and in London. Bob Schatz heads up software development with some 130 people. The core team is at corporate headquarters in Bala Cynwyd, Pennsylvania, with another team in San Francisco.

Schatz has been with Primavera for over three years and was previously with GE Aerospace. 'Sorry', he says, 'security means I can't talk about my work there.' He then went to Liquent, a start-up company that grew rapidly from seven people to 250. It developed enterprise software for pharmaceutical companies as part of the approvals submissions to the Federal Drugs Administration.

Schatz is quite candid that, like most other software developers, Primavera had problems coping with competing demands. People were well aware of the irony that they produced project management software, but sometimes 'found it a little challenging to do it themselves'.

'We were using a traditional waterfall approach when I joined the team and we made a lot of adjustments to improve efficiency, but towards the end of developing every release we were still running up a lot of overtime and working weekends', he says. The net effect was tiredness, boredom and demoralisation among team members. So he looked around for a way of solving this common dilemma for IT developers.

About 18 months ago, with the backing of Faris and Koppleman, he decided to take a different approach to the problem and



engaged two of the world's leading experts in Agile development, Scrum techniques and XP Agile – Ken Schwaber and Bob Martin – to advise how it could be done.

Three years ago, only early adopters were using Agile, but Ken Schwaber says that over the last 18 months some 1300 project managers have been trained to manage Scrum projects. His predictions for '05 are that average productivity can increase by 200% using Agile and that 'business models for cooperative work will emerge and become dominant'; but he also warns that only a third of companies that develop software will adopt the ideas because 'it is very hard to implement'. The problem is the degree of organisational change required. But he also predicts that of the two-thirds not using Agile, they will pass on the work to those that do.

In a nutshell, Agile is about producing business value as quickly as possible by incremental development. Teams provide usable, 'finished product' (ie, potentially shippable and complete with documentation) typically in 30 days or less. Each 30-day period builds a part of the complete release until all the work is completed and is ready to ship to the customer. This requires multi-functional teams and close collaboration between customer and developer.

Primavera had previously used a traditional approach with a document from marketing, setting a new release in motion. Development staff and project managers would then break the work into tasks and organise them into a



**Primavera have found Agile so valuable that they have worked with their own API to develop applications that bring Agile practices together with enterprise project management. These add-on modules are being used by Primavera's development team and a few customers that have committed to Agile practices and have invested in Primavera solutions. Schatz believes that the combination of Agile solutions with traditional enterprise project management software will actually accelerate the adoption of Agile in the software industry. As this solution is developed, Primavera will most certainly make this generally available.**

9-12 month project schedule. Hundreds of tasks were divided among analysts, documenters, designers, programmers and testers. Project management was a command and control structure, and those few people who made decisions were never close enough to the actual work. Inevitably, as time went on, requirements changed or were added in the middle of release and relations strained between development and other departments. Finally, this led to the typical charge to the finishing line with all the problems that entailed.

With Schwaber's help, Schatz implemented the use of Agile and of Scrum – an Agile process for complex projects. The new ideas had an immediate impact on the company's culture. Out went the neat blocks of cubicles, where individuals sheltered behind their low-rise screens, in favour of self-organised team rooms. To a company used to a more formal use of office space, it must have seemed as if anarchy had just broken out.

They were reorganised into teams with 7-10 people in each team. And all the managers were trained as ScrumMasters – the Scrum equivalent of a project manager. The ScrumMaster's role is to facilitate collaboration between the marketing department and development teams as they build functions for the monthly deliverables.

Unlike a traditional project manager, the ScrumMaster is not responsible for the schedule. Schwaber says: 'It may seem risky, even foolhardy, to trust the team to plan and execute its own work. However, it has worked on thousands of projects. It is productive because, firstly, the project manager doesn't have to tell the team what to do and then to keep the plan up to date as changes are required. Secondly, the team works more effectively without having to rely on external authority for any changes.'

It certainly worked at Primavera for developing the 4.0 release, which involved integrating a new workflow and collaboration system into their project management software.

At the end of the first iteration – or Sprint as it is appropriately called in Scrum terminology – the management and marketing review team were amazed at what had been achieved just one month into the development cycle. And it is at each review stage that the customer community decides on the next priorities having seen a working component or function of the product.

The success of the first Sprint led to the whole release being managed through Agile techniques. But the physical co-location of teams and the mutual self-help between coders, analysts and testers was only part of the changes required. It was soon realised that producing a mini-release every 30 days required different engineering practices. So eight months after introducing Scrum, Schatz brought in Bob Martin and a team from Object Mentors to implement Agile Extreme Programming (XP) to ensure that speed of development was not compromised by quality problems.

Under the rules of XP, a feature is not done until it passes both its unit tests and acceptance tests. Programmers write the unit tests, and marketing and quality assurance write the acceptance tests. In both cases, they are



**'Agile has had a very positive impact on our ability to deliver quality software in record time to our customers. It empowers the people doing the work and pushes us to make the tough decisions early in the process. Since Agile assumes that change is inevitable, we can move quickly to respond to market changes and customer requests.'**

*Dick Faris, CTO*





*Bob Schatz, vice president for software development, instigated the use of Agile and XP at Primavera*

written before the code is developed. The tests are automated and become the detailed specification and completion criteria for each Sprint and each feature.

This type of cooperative work between testers, analysts and coders is known as test-driven development. It also addresses the problem posed by cross-functional teams of getting everyone involved from the start.

Object Mentors also concentrated on making sure that quality was not ignored in the drive towards feature delivery. They taught the teams how to design solid systems, create clean code, clean up legacy code and 'keep' the design and code clean for each Sprint.

Schatz is keen to emphasize that the teams choose their work for a Sprint from a prioritised list produced by the internal 'customer'. 'It's the amount of work that each team knows they can accomplish in the time,' he says.

As all the skills to complete the work are to be found in each team, they will work across boundaries to help each other: designers and testers, for instance, can have a much better understanding of each other's work and can assist each other rather than being in functional silos.

As part of their internal self-management, the members of a team meet for 15 minutes every day to discuss their previous day's work, their current day's work and any obstacle that might be in the way. There is a coach for each team who is responsible for taking that obstacle away.

The techniques have been adopted very successfully at Primavera. Schatz says: 'It has had a major impact on what we do. Release 4.0 had more functionality than any other previous release and we didn't work one weekend or one hour of overtime. We were even able to start the next version on the Monday as we delivered the previous version on the Friday before. This was amazing. People were happy, they were energised; it was just amazing.'

But, Schatz acknowledges Schwaber's view that implementing Agile is not an easy option. It requires a major cultural shift, particularly from managers who no longer 'tell everyone what to do'; and there can be resistance from staff who don't want to lose their offices or cubicles. His answer to the latter was quite simple. Take those who are willing to co-locate as a team in one room and tell them that if any other members of the team wanted to talk with them, they must come to the

'Agile processes provide a new form of software development (see [www.agilealliance.org](http://www.agilealliance.org)). Founded on an empirical model of process control theory, Agile processes deliver value iteratively and incrementally.

'Customers and development teams collaborate to wrest the greatest value from advanced technologies and emerging requirements. We call this value-driven software development.

'The four variables that control software development projects are the cost, the date of delivery, the quality, and the functionality. The traditional approach to software development attempts to fix these at the start of the project to form a contract between the customer and developers.

'In a traditional development project, constrained by a contract fixing cost, date, functionality, and quality, changes either are deferred until after the initial system is delivered or extensive change management processes are used to adjust the four aspects whenever a change is requested. This results in systems that don't provide the business value because they are out of date. Worse yet, it may result in systems that are never delivered because they are embroiled in the change management process. Agile processes avoid this dilemma by embracing the expected changes through value-driven project management.

Value-driven projects leave the determination of the four variables in the customer's hands throughout the project. The customer only commits to one iteration, usually of 30 days or less. The customer authorizes development iteration by iteration, and is free to change any of the variables based on progress to date and the business value that is provided.

Rather than contracting, the customer and development team collaborate to achieve business value. Prior to each iteration, the customer identifies the highest priority requirements that will create business value. The team identifies how many of these requirements can be turned into a product increment that delivers that value during the next iteration. The team then proceeds, for one iteration, doing their best to create this business value. The only deliverable required from the team is live, demonstrable business functionality working on a computer.

At the end of the iteration, the customer reviews the working system functionality with the team.'

*Ken Schwaber*

room. 'Within a week, all the team was co-located and no-one wanted to leave.'

From Primavera's viewpoint, the changes have been positive, but it has all required considerable hard work and the faith of senior management. But Schwaber says that Agile is no silver bullet. It is not a process to be taken off a shelf, imposed, and then results will flow automatically. The process that Primavera has undertaken is one shared with companies everywhere; the process of change and change management. It has included changes in relationships between individuals, departments and managers, in the tools employed, and in the way progress is perceived. The hardest change of all has been keeping a hand on the tiller while letting others figure out how to work together, rather than telling them how

to work together. And Schatz is convinced that implementing Agile should not fall into the hands of consultants. 'The moment they start offering Agile implementation, the value of the technique will be destroyed', he says.

What should impress Primavera's current and prospective customers is that they are not only providing leading-edge project management software, but are also able to demonstrate that they actively embrace new management ideas themselves. Schatz and his team have worked with a number of Primavera customers and prospects to help them understand the different approaches available and the challenges they will face in the adoption of Agile methods. His goal is to help make the benefits of this new approach a reality for Primavera customers.



*Ken Schwaber is president of Advanced Development Methods (ADM), a company dedicated to improving the software development practice. He is an experienced software developer, product manager, and industry consultant. He initiated the process*

*management product revolution of the early 1990s and also worked with Jeff Sutherland to formulate the initial versions of the Scrum development process. ([www.controlchaos.com](http://www.controlchaos.com))*



*Robert C. Martin has been a software professional since 1970. He is CEO, president, and founder of Object Mentor Inc. In 1998 Object Mentor embraced, and helped to define, the Agile development method of Extreme Programming. In 2000, Object Mentor took*

*this to the next step by helping to form the Agile Alliance, a group of professionals dedicated to the advocacy of lean and emergent software methods.*