



Meeting Customer Demand for Agentless Technology

Using Agile Software Development Methodology

Table of Contents

ABSTRACT 1

ESTABLISHING SYSTEM REQUIREMENTS..... 2

LAUNCHING THE PROJECT AT BMC..... 3

REAPING THE REWARDS 3

CONCLUSION..... 3

APPENDIX — OVERVIEW OF SCRUM AGILE SOFTWARE
DEVELOPMENT METHODOLOGY 4

Abstract

Software vendors are often under intense pressure to continually enhance and evolve their products to meet changing customer needs. These needs are driven by two major factors:

- > As IT increasingly permeates business processes, and IT environments become more complex, IT staff may require tools that are more sophisticated to keep the infrastructure running smoothly and continuously improve the quality of the business services delivered — while also reducing costs.
- > As IT organizations adopt emerging technologies — such as services-oriented architecture, server and storage virtualization, and on-demand computing — they need advanced solutions to help them manage effectively.

To meet customer demands, many software vendors embrace the concept of agile software development, an approach based on following adaptive principles and working with teams that are focused on accelerating the process for delivering high-quality software that is easy to implement and manage.

This paper reviews how BMC® Software used an agile development methodology to responsively develop innovative agentless monitoring for automatic and discovery monitoring solutions. Agentless monitoring can simplify the installation and administration of such solutions and, as a result, help IT improve the quality of delivery of business services in the face of flat, or even decreasing, IT budgets.

This agile approach was taken because traditional software development methodologies would not allow developers to move quickly enough to satisfy customer needs. The BMC development staff decided to create a new development environment that would foster greater speed and agility — one that would tap the full potential of the developers' collective ability to deliver quality software more rapidly.

After evaluating several agile software development methodologies, the staff concluded that the Scrum methodology (see *Appendix — Overview of Scrum Agile Software Development Methodology*) would best suit the needs of this particular project. Why Scrum? Primarily because of its ability to get teams operating quickly and efficiently, with a focus on quick turnaround and maximized productivity. The results were gratifying: The staff accomplished four major releases of a flagship product in just one year, a fraction of the time that would have been required to develop this capability and level of quality using traditional methods.

Although the agile approach may not be appropriate for all projects, it served this project particularly well, primarily because of its stringent time-to-market requirements. The approach enabled the development staff to quickly create a solution, based on a new generation of technology that not only meets demanding functional requirements but also permits customers to easily absorb the new solutions and take advantage of their advanced functionality.

This paper reviews the highlights of the development project and describes the significant benefits realized.

Establishing System Requirements

Responding to rapidly changing customer needs, the development staff of the BMC Infrastructure and Application Management (IAM) Business Unit set out to evolve the company's discovery, monitoring, and management solutions to take advantage of agentless technology. The new solutions would provide full remote discovery monitoring and management functionality, while largely eliminating the need to install agents on infrastructure components.

Agentless technology simplifies the installation and administration of discovery and management tools, helping IT organizations increase operational efficiency and reduce costs. This protects investments in existing technology and at the same time provides the opportunity to deliver greater functionality so that IT organizations can be more responsive to their business customers. The new capabilities would include remotely executed automated recovery routines and dynamic provisioning of systems management software.

The development staff knew that agentless operation involves a quantum leap in technology and functionality. With traditional software development methodologies, such an effort would take years. With the Scrum approach, however, the staff was confident it could deliver solutions in a much shorter timeframe.

The staff began by establishing the requirements for the new solutions. First, of course, was the fundamental requirement for agentless operation. To this, the staff added three more requirements:

1. Develop a hybrid agent — The developers realized that some operating platforms impose limitations on remote discovery and management that make it impossible to satisfy all requirements with agentless technology alone. So, they added a requirement to develop a hybrid agent to augment the agentless technology. The hybrid agent would provide the best of both the agent and agentless worlds. It would offer in-depth discovery monitoring and management, yet would be thin and unobtrusive to the target system. What's more, it would be virtually invisible to system administrators, in that it could be deployed and updated automatically and dynamically to target systems — without any administrator intervention.

2. Mix and match agentless and hybrid technologies — The developers further realized that by permitting the agentless and hybrid technologies to work together, in any combination, customers could establish different levels of discovery monitoring and management for different system components. This ability would enable customers to focus attention on their most business-critical assets for the most efficient use of valuable IT staff resources. For example, an IT organization for an airline might manage the mainframe that hosts its passenger reservation system much more aggressively than it would a printer in the human resources (HR) department.

The ability to mix and match agentless and hybrid technologies would deliver another significant advantage. Customers often want the latest enhancements to certain solutions, but soon realize that absorbing new releases can be time-consuming. In the past, IT organizations were faced with a significant installation effort to deploy new or upgraded agents across hundreds, or even thousands, of IT infrastructure components. The combination of agentless and hybrid technologies, however, eliminates much of this onerous task, permitting customers to take ready advantage of new releases.

3. Coexist with existing solutions — Another major requirement was to permit customers to take advantage of the new agentless technology without needing to "rip and replace" their current BMC solutions. Instead, customers would be able to mix and match agent-based and agentless solution infrastructures and management applications. The solutions already in place would work the same way they did before the upgrade, protecting previous technology investments. The customer could then move to the new technology at the pace most suitable to the organization. This requirement mandated the need for enhanced technology that would enable the agentless solutions to subsume the architecture of the company's current monitoring solutions.

Launching the Project at BMC

Having established these ambitious goals, the developers set out to achieve them. Because it was the company's first experience with agile software development, and Scrum in particular, development management first had to lay some groundwork, which included educating the organization and forming Scrum teams.

During the educational phase, which took approximately four months, development management solicited buy-in at all levels of the organization, from senior executives to individual developers. People who had worked with Scrum in the past shared their experiences, and management distributed copies of the book "Agile Software Development with Scrum,"¹ written by Scrum co-creator Ken Schwaber.

In January 2005, the business unit began forming Scrum teams to start work on the project, creating only a few teams initially. Development managers heavily involved themselves in the project. They encouraged team members to pay close attention to the project, and were highly responsive to team needs and open to team members' suggestions.

The phenomenon of self-learning soon began driving and even accelerating progress. People started sharing experiences and insights, and learning from each other. This spirit of cooperation and collaboration became a major driving force. This eagerness to leverage collective knowledge is one of the primary benefits of the Scrum methodology.

One important lesson that the team learned early on is the need to ensure that all team members are empowered and involved, regardless of their discipline. For example, the position of *Scrum master* — the team leader who acts as a clearinghouse for issues and communication — has traditionally been held by a software developer. The development staff, however, soon discovered that it was highly beneficial to appoint Scrum masters from other disciplines as well. Toward that end, QA people were appointed Scrum masters on some teams, making them full-fledged members of the team. This tactic increased the quality of interaction between development and QA, and instilled in the team a deeper appreciation for the QA function.

Reaping the Rewards

The project not only met expectations but also exceeded them. The development staff accomplished a quantum leap in functionality of a flagship product line in a very short amount of time. The staff accomplished four major product releases in 12 months, delivering advanced functionality to customers, with releases that made deployment easy.

The new releases leveraged the strength of solutions built with BMC classic solutions that were formerly known as PATROL[®], combining that advantage with remote-monitoring capabilities. The result was a solution that offers a very low total cost of ownership, while providing the functionality customers need.

Conclusion

Using agile software development methodology, the development staff was able to meet customer demand for agentless technology in its discovery, monitoring, and management solutions. And it was able to meet the demand in the short timeframe customers required.

Because of its proven ability to shrink software development cycles, many software vendors are adopting agile software development methodologies to meet customer demands for continually enhanced functionality. And agile software development is by no means limited to software vendors. A variety of other organizations have in-house software development teams that need to develop and evolve business applications and other software quickly to help their internal customers adapt readily to changes in the business environment. These internal development organizations also can take advantage of agile software development methodologies to accelerate their software development projects. It is hoped that this paper will provide motivation and guidance to those organizations that are implementing agile software development.

BMC Software offers BMC[®] Performance Manager, which provides agentless monitoring for automatic and discovery monitoring solutions. For more information about BMC Performance Manager, please visit: www.bmc.com/performancemanager/.

¹ "Agile Software Development with Scrum," Ken Schwaber and Mike Beedle, Prentice Hall, October 15, 2001 (1st edition)

Appendix — Overview of Scrum Agile Software Development Methodology

Scrum development differs from traditional methodologies in its view of the process of systems development. Traditional approaches assume that the process is well understood, can be closely planned and estimated up front, and can be successfully completed based on that plan and estimate. Scrum, on the other hand, views systems development as a complicated process that can best be described as *overall progression*.

At its core, Scrum is an iterative, incremental process for developing any product or managing any work that produces a potentially shippable set of functionality at the end of each iteration.² Scrum emphasizes teamwork, self-organized teams, and the removal of development roadblocks within each team.

A Scrum software project is controlled by establishing, maintaining, and monitoring key control parameters, such as backlog (functionality yet to be implemented) and risk. Control is subtle and mostly indirect. Control is exercised primarily by such factors as creating an open work environment, encouraging feedback, establishing an evaluation and reward program based on group performance, managing the tendency to go off in different directions early on, and tolerating mistakes. One notable characteristic of Scrum is that it gives product management and other stakeholders great flexibility. Team members release a working product after every iteration, and priorities can be changed after every iteration to reflect changing market and customer needs.

The Scrum methodology consists of three distinct phases:

1. Planning and system architecture phase
2. Sprint phase
3. Closure phase

Planning and System Architecture Phase

In the planning and system architecture phase, team members establish the functionality to be included, and prioritize that functionality based on customer needs. They also set the release date — that is when the system will be released to the market, or, in the case of an internal application development project, released into production. The team identifies available development resources, envisions the architecture, and establishes the target operating environment.

Sprint Phase

At the beginning of the sprint phase, the project manager establishes sprint teams. The sprint phase comprises a series of sequential phases called “sprints.” At the beginning of each sprint, each team receives its assignment and is told to “sprint” to achieve its objectives. Each team can select its own means and tools to achieve the objectives.

At the conclusion of each sprint, all teams convene with the project manager and the enterprise’s senior executives to review progress. Project managers and superiors can change anything and everything at the conclusion of the review session. The sprint phase is completed when the objectives of the development effort are achieved in terms of functionality and quality, or when time and budget constraints are reached.

Closure Phase

The closure phase consists of finishing system and regression testing, developing training materials, and completing final documentation.

For Additional Information

For additional information on the Scrum development methodology, please visit: www.controlchaos.com/.

² “A CIO’s Playbook for Adopting the Scrum Method of Achieving Software Agility,” Dean Leffingwell and Hubert Smits, Rally Software Development Corporation, Ken Schwaber, and Scrum Alliance White Paper, July, 25, 2005



ACTIVATE BUSINESS WITH THE POWER OF IT.

About BMC Software

BMC Software helps IT organizations drive greater business value through better management of technology. Our industry-leading Business Service Management solutions ensure that everything IT does is prioritized according to business impact, so IT can proactively address business requirements to lower costs, drive revenue, and mitigate risk. BMC solutions share BMC® Atrium™ technologies to enable IT to manage across the complexity of diverse systems and processes — from mainframe to distributed, databases to applications, service to security. Founded in 1980, BMC Software has offices worldwide and fiscal 2005 revenues of more than \$1.46 billion. BMC Software. Activate your business with the power of IT. For more information, visit www.bmc.com.

About the Author

Israel Gat, Vice President of Infrastructure Management for BMC, is responsible for the BMC Performance Manager product line (formerly part of the PATROL product line), as well as the BMC® SmartDBA® and BMC® SmartDBA® Recovery Management suites of distributed systems products. Gat has been an executive for some of the world's top technology companies, including Microsoft, IBM, Digital, and EMC.

