

EMC PERSPECTIVE

## **Bridging the Gap to Agile: Scrum Status Reporting in a Waterfall Organization**

Some proponents have insisted that for agile to work, it must be adopted by the whole company simultaneously. For many organizations, this is not a practical approach. Our experience shows that it is possible to bridge the gap between agile teams and traditional “waterfall” organizations. This paper examines one approach.

## Table of contents

Agile gains momentum .....	3
Expectation gap .....	4
Familiar and comfortable .....	4
Maintaining discipline .....	4
The value of an intermediary .....	5
Scrum status reporting: a case study .....	5
A formula for translation .....	6
The value of the perception of data .....	8
Win/win success .....	9
Conclusion .....	9
EMC Consulting .....	9

# Bridging the Gap to Agile: Scrum Status Reporting in a Waterfall Organization

## Agile gains momentum

Many companies are moving to agile software development to improve quality and reduce delivery times. In contrast to traditional software project management—often referred to as “waterfall” because of the way work is broken down into a series of sequential steps—agile methods rely on short, iterative cycles and close collaboration between the product owner and the development team.

Waterfall project management is predictive. It assumes that requirements can be defined up front, projects will follow a sequential flow, and that once a phase is complete, it will not be revisited. Agile, however, is adaptive, with built-in mechanisms for the continual revisiting and revising of objectives, priorities, and assumptions. This makes agile well-suited for projects in which requirements are likely to change or new tools are being used.

As organizations struggle to meet increasingly demanding requirements in ever-more complex and fluid environments, the use of agile development is growing.

As the graph from Forrester Research, Inc., (“Inquiry Insights: Agile Development,” October, 2008) in Figure 1 shows, interest in agile keeps growing.

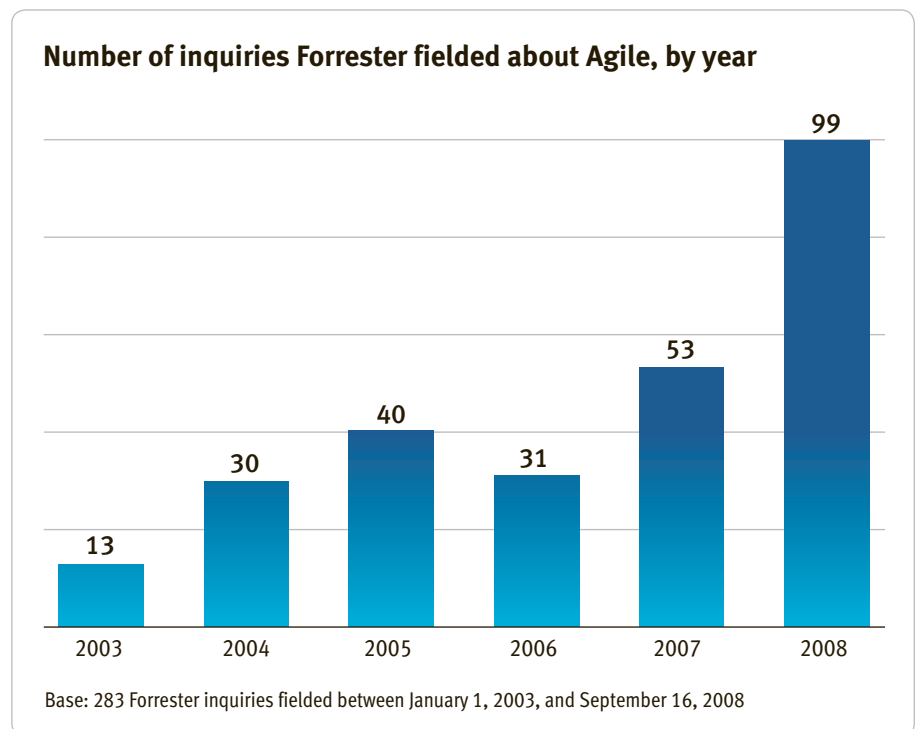


Figure 1 Interest in Agile continues to grow

One of the most popular and tested agile development methodologies is called Scrum, after the rugby term for moving the ball forward as a group. Among the companies using Scrum are IBM, Microsoft, Sun Microsystems, Siemens, State Farm, and Yahoo!, which has several hundred developers working in Scrum teams worldwide.

The move to agile methods, such as Scrum, Lean, and Extreme Programming, has resulted in higher quality software that better meets the needs of users, lowers costs, and provides faster time to solution for many organizations.

## Expectation gap

Despite these advantages, few development organizations move to agile development all at once. Most opt to begin with small projects or portions of projects. Because of the considerable philosophical and practical differences between agile and conventional project management, however, an additional challenge arises. For example, how can agile teams meet the expectations of others in the organization who expect traditional project plans and status reporting?

Even when an entire development organization moves to agile, other parts of the business will likely continue to expect the usual plans and reports. Wherever such an “expectation gap” opens, it needs to be recognized and addressed.

## Waterfall: familiar and comfortable

For decades, waterfall project managers have presented stakeholders with detailed plans for the entire development process, along with documented commitments and regular status reports on how well the project is meeting key milestones. In addition to being familiar, such documents give stakeholders a sense of comfort and control—although, if initial assumptions are incorrect or requirements change, quite a lot of work can be done “to plan” before the need for change is recognized, and much time, effort, and money can be wasted.

In contrast, agile teams approach development as a series of small, incremental efforts. In Scrum, for example, these are called “sprints.” Each sprint begins with a planning meeting between the product owner (client) and the Scrum development team.

Beginning with an initial list of all desired functionality (known as “the product backlog”), the product owner prioritizes which increment of functionality is to be pulled from the backlog for each sprint. The team then breaks the increment of functionality down into its components and comes to a consensus about what can be delivered in the developer hours available during the next sprint. (The duration of sprints varies from project to project, typically lasting between one week and one month.) Scrum teams strive to deliver a piece of working software with each sprint that can be user-tested and give the product owner the opportunity to add new requirements to the product backlog or reprioritize as necessary.

This iterative approach uncovers problems quickly and drives close, frequent collaboration between the product owner and development team to solve them. Such continual, incremental revisiting, however, makes it very difficult to provide a meaningful, detailed, long-range plan or to report on project progress using traditional waterfall tools.

## Maintaining discipline

Those outside of the agile process often perceive its frequent reevaluations, prescriptions for face-to-face meetings over written documentation, and inability to commit to long-term deliverables and dates as a lack of discipline.

In fact, agile development relies on time-tested processes. Scrum, for example, follows a simple, but well-defined framework—with three roles, three types of meetings, and three ways of prioritizing and tracking tasks. Experience shows that compromise of agile processes or principles can seriously compromise results. One example of a Scrum discipline that organizations often have a hard time maintaining is that once a sprint has begun, the team stays fully focused on delivery of that next increment of software and is not interrupted or distracted with further input or changes in direction until the next sprint planning meeting.

Shielding agile processes and teams from outside pressures, intrusions, and compromise, then, becomes another critical factor in project success.

## The value of an intermediary

One way to bridge the gap between agile and traditional stakeholder expectations and protect the integrity of the agile process is by establishing an intermediary between the agile team and the waterfall organization.

The value of an intermediary is two-fold:

1. To act as a conduit, helping to educate and earn the trust of waterfall stakeholders through proactive communication and by translating agile output into data that they can understand; and
2. To act as insulator, keeping the concerns and pressures of the waterfall organization from compromising agile processes or interfering with the team's focus and productivity.

Characteristics of an effective intermediary include:

- **Experience in both agile and traditional project management.** As a conduit, the intermediary needs to be able to credibly “speak both languages” and translate agile artifacts into a form that can be easily and quickly grasped by waterfall managers.
- **Able to resist political pressure.** As an insulator, the intermediary needs to hold firm against “top-down” demands that could compromise the agile process or distract the team—one reason an outside consultant can be effective.
- **Excellent communication skills.** Frequent, proactive communication and a skillful shifting of the conversation early in the process from the defensive (“Why you can’t get a Gantt chart...”) to the positive (“Here’s what I can show you...”) are critical to winning the confidence of the waterfall stakeholder.

## Scrum status reporting: a case study

A recent EMC® Consulting Services engagement at a \$2 billion services company illustrates how an outside consultant, acting as intermediary, was able to satisfy the expectations of a waterfall organization, while also insulating the agile team from distraction and process compromise.

EMC was initially engaged to help move the entire development team to Scrum in order to rebuild the company's core business application on a web-enabled, SOA platform. The application—which automates every aspect of their business, from contracts, to billing, to payroll—had taken more than two years to develop, was now 10 years old, and ran on old technology. Initial requirements were no more specific than, “we want the current application in .NET.” In addition, there was no documentation for the application, it had been modified numerous times over the past decade, and a “hard deadline” for delivery had been established, which was only eight months out.

The senior business sponsor of the project, who did not have the time or initial interest to learn about Scrum, asked for a traditional waterfall-style Gantt chart of milestones with weekly status updates; something that Scrum could not credibly deliver.

EMC proposed instead that a senior EMC consultant, certified as both a Scrum Master (Scrum Alliance) and a Project Management Professional (PMI), act as intermediary. The consultant would help the waterfall manager understand why the Scrum process did not lend itself to traditional Gantt chart reporting—and figure out how to translate Scrum data into status updates that would meet his needs.

The consultant quickly determined that behind the senior manager's request were regular status updates on project progress—presented in a format that he could then use to report to his management.

## A formula for translation

Fortunately, the product owner on the Scrum team had been part of the original application development effort and had remained engaged with the product ever since. He provided a list of functionality which, along with use cases, was used to create the product backlog. Fortunately as well, the Scrum team had decided on sprints of one week (five working days) in duration, which would provide the consultant with weekly data that he could use to prepare some kind of update for the senior manager.

The Scrum team used “planning poker” to estimate the complexity of work and determine what they thought they could deliver in the next sprint (with any unfinished work going back into the product backlog). In planning poker, team members discuss the tasks required to deliver the prioritized increment of functionality and then choose and reveal a card all at once with a Fibonacci number (0, 1, 2, 3, 5, 8, 13, 21...) representing their estimate of complexity.

Outliers (those showing the highest and lowest number) are given a chance to explain their estimates and the group again plays their cards. Studies have found that group estimates obtained this way are more accurate than simply averaging individual estimates, because individuals are not influenced by the estimates of other team members.

As part of each sprint planning session, the team also applied a formula it had developed to convert the agreed-upon Fibonacci number into a time estimate, to gauge what functionality they thought they could deliver in the sprint.

The product owner had already determined his own Fibonacci estimate, along with a formula he had developed to convert each Fibonacci number into a time estimate.

At the end of a sprint planning session, the EMC consultant had for each item:

Item Description

Team F Number  
Team Time Estimate  
Original F Number  
Original Time Estimate

The EMC consultant tracked the variance between the product owner’s original estimates and team estimates after the planning session for both the Fibonacci numbers and hours. (During sprint reviews, the team could also have tracked the amount of time it actually took to complete items, but that was not a practice the team adopted.)

Within a few weeks, a relatively stable variance emerged for each type of estimate. The product owner’s original work estimate was averaging 128 percent of the team’s estimate for the same task—and his original Fibonacci estimates were 151 percent of the team’s estimates. (The product owner intentionally padded his estimates by assuming no reusability, that is, all elements are built from scratch.)

Multiplying the original estimated hours against the variance could, therefore, produce a number close to what the team would be likely to estimate—even though the team had not yet evaluated the work remaining in the product backlog. Likewise, the original Fibonacci estimates could be multiplied against the variance to come up with the team’s Fibonacci estimates. For example:

- According to the original product owner’s estimate, the total number of developer hours required to complete the project was 5,658.
- Applying the variance (128 percent) between original estimates and team estimates results in a total of 4,420 for the number of hours the team would estimate to complete the project.
- Applying the variance (151 percent) between the product owner and team Fibonacci values results in a figure of 3,747 for the total number of hours of work remaining.

<sup>1</sup>Molokken-Ostfold, K. Haugen, N.C. “Combining Estimates with Planning Poker—An Empirical Study,” IEEE. Software Engineering Conference, April 13, 2007.

The consultant then plugged these data points into the traditional waterfall-based, three-point PERT Analysis formula:  $(O+P+M)/3$ , for which:

M = Most Likely (corrected original work hours total) = 4,420

P = Pessimistic (original hours total) = 5,658

O = Optimistic (corrected F hours total) = 3,747

Or, in this instance,  $(4,420 + 5,658 + 3,747)/3 = 4,608$  hours of total work required to complete the project.

Given the firm deadline date, it was also possible to calculate the number of developer hours remaining for the project on a weekly basis (factoring in planned vacation and unexpected staffing changes or absences as they occurred). The consultant also tracked planned hours of work against actual hours worked for each sprint.

This enabled the consultant to show trends in performance by comparing Actual Hours Worked vs. Hours Planned (Figure 2).

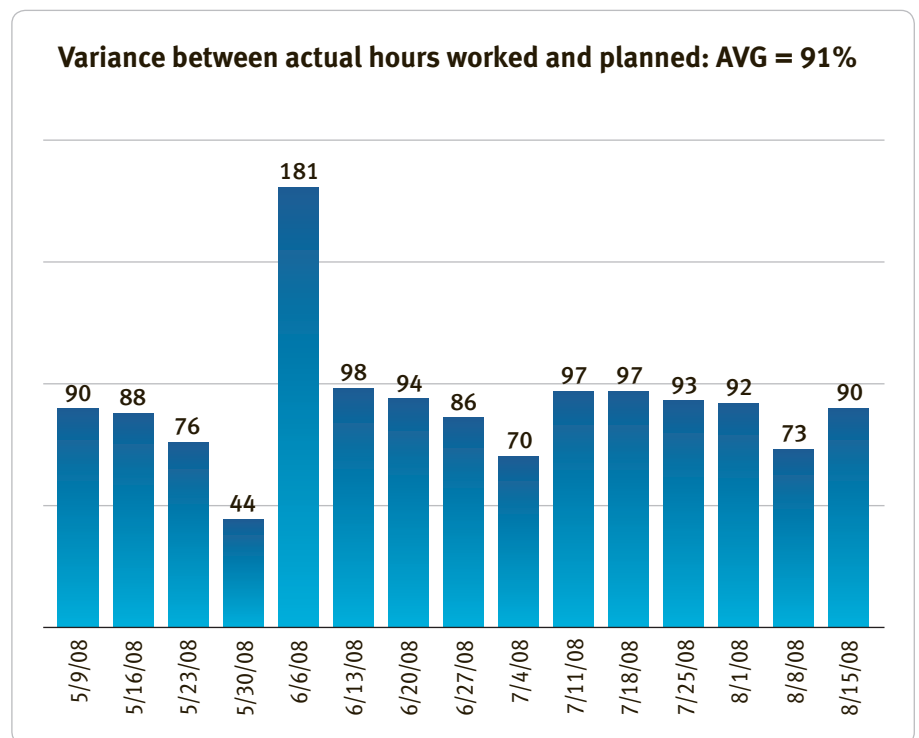


Figure 2

By comparing estimates of work remaining (and correcting for the variance between planned vs. actual hours worked), the consultant was able to provide the senior manager with a weekly estimate of the number of weeks required to complete the project. It should be noted that this number fluctuated significantly from sprint to sprint. At one point, the team was ahead by five weeks, at another, it was behind by 13 weeks.

In addition, the consultant tracked status of each backlog item according to whether it was:

- Submitted—Use Case Submitted
- In Process—Being Developed
- Testing—In Testing (Done per Developers)
- Done—Work Complete (Approved)
- Deferred/Deleted—Not to be completed during this phase or no longer requested

This data was then used to create a graphical representation (Figure 3) that provided the waterfall organization with a snapshot each week of where the project stood against delivery.

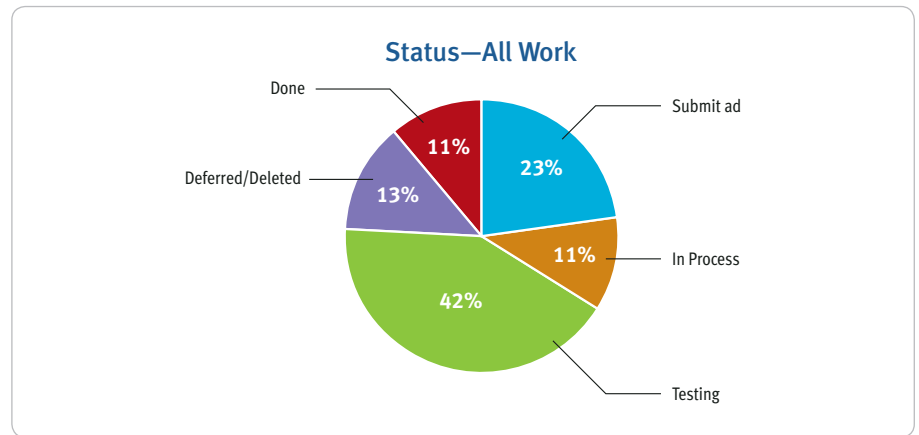


Figure 3

Eventually, categories were rolled up into an even simpler form, as shown in Figure 4.

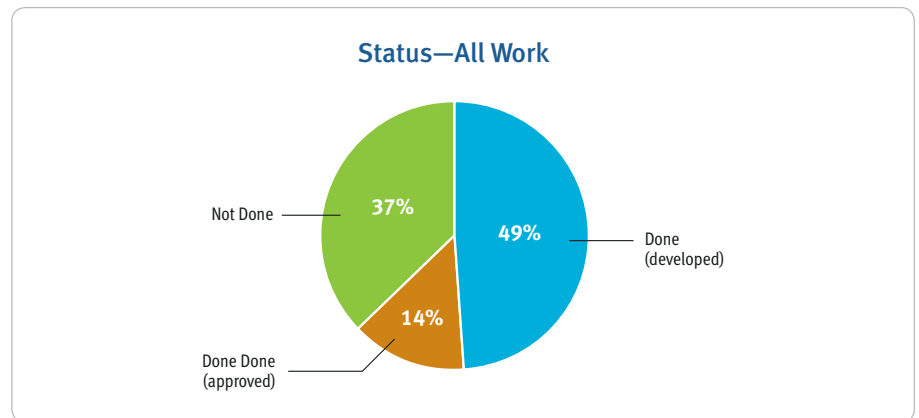


Figure 4

### The value of the perception of data

These calculations, performed week to week, provided little more than educated guesses as to how close the team would actually come to delivery of the requested functionality by deadline. Nevertheless, they provided trend lines, which showed the team moving closer to (or sometimes, further away from) target from week to week.

Over time, this data proved sufficient to increase the senior manager's comfort level, as well as to satisfy his managers.

Equally, or perhaps, more importantly, it protected the Scrum team from the distraction of questions and demands from the waterfall organization, enabling them to focus on development. In a post-mortem at the end of the project, no one on the development team reported awareness of any friction between the IT and business organizations over Scrum reporting, although early in the project, this had been a point of serious contention.

### Win/win success

At deadline, the Scrum team had delivered 97 percent of the requested functionality. What's more, it had accomplished in eight short months what had taken the original development team more than two years\*.

It managed to do this as requirements evolved, staffing levels fluctuated, and team members changed—and while learning and converting to Scrum.

Perhaps the greatest win, however, was the fact that when a new “Phase 2” development project was begun, senior management found the new methods of tracking and reporting completely acceptable. There were no further questions about the validity of Scrum estimates or sideways glances.

Senior management may still not fully embrace agile methodology, but they had gained enough confidence to let go of the waterfall plans and reports they had once demanded.

### Conclusion

Our experience shows that it is possible to successfully use agile methodologies in a non-agile organization. An intermediary with the right skills and knowledge can work with both the agile team and the waterfall organization to develop a solution that creates a protective bubble around the agile team and its processes, while at the same time provides waterfall managers with the information and reporting they need to understand and monitor project progress.

### EMC Consulting

As part of EMC Corporation, the world's leading developer and provider of information infrastructure technology and solutions, EMC Consulting provides strategic guidance and technology expertise to help organizations exploit information to its maximum potential. With worldwide expertise across organizations' businesses, applications, and infrastructures—as well as deep industry understanding—EMC Consulting guides and delivers revolutionary thinking to help clients realize their ambitions in an information economy. EMC Consulting drives execution for its clients, including more than half of the Global Fortune 500 companies, to transform information into actionable strategies and tangible business results.

\*It's worth noting that, when, as an exercise, the project scoped out using waterfall project management, it appeared to be two-plus years of work.



EMC Corporation  
Hopkinton  
Massachusetts  
01748-9103  
1-508-435-1000  
In North America 1-866-464-7381  
www.EMC.com

#### Take the next step

If you would like to learn more about ways EMC can help your organization bridge the gap to agile development, visit [www.EMC.com/consulting](http://www.EMC.com/consulting) or contact your local EMC Consulting representative.