

<http://www.fawcette.com/resources/managingdev/methodologies/scrum/default.asp>

Rugby, Anyone?

Scrum tackles software development from a patterns perspective to turn around a fast product.

by *Brian Noyes*

Posted June 28, 2002

In a rugby game, a *scrum* is a part of the game that is a cross between a kickoff and a quarterback snap in American football: a "play in which the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them," as defined by Merriam-Webster's Collegiate Dictionary. In a software development context, Scrum is an agile software-management process characterized, among other things, by quick daily meetings to report on project status. It has not received nearly as much attention as [EXtreme Programming \(XP\)](#) or [Rational Unified Process \(RUP\)](#), but it's gaining popularity and it's simple to understand. Like XP and RUP, Scrum tries to address the shortcomings of traditional software processes, where the assumptions that software development can be repeatable and well defined were often flawed.

Scrum approaches software development from a patterns perspective. Software development might not always be strictly repeatable and well defined, but certain patterns emerge. You can apply these patterns to address the issues of software development and achieve a highly effective and productive process.

Like many agile methods, Scrum puts a strong emphasis on communication, teamwork, and flexibility. By adopting practices that enhance these aspects of the process, you can plan the complexities of software development to a degree, but you can also tackle unplanned or unexpected events effectively to maintain forward progress.

Software process elements (patterns) in Scrum include Backlog, Sprints, Scrum Meetings, and Demos. Backlog consists of the list of prioritized requirements or features the customer wants the team to build. Backlog is a dynamically changing list that management constantly reassesses. The customer, marketing department, and developers can all add new items to the Backlog, but only the product manager is allowed to change the items' priorities.

Sprints are the basic units of scheduling, typically 30 days or less of development activity. A Sprint consists of a preallocated work unit from the Backlog that composes the work and features to be completed during the Sprint. During a Sprint, the Backlog that the Sprint includes is static—no new items may be added and the prioritization remains fixed for that portion of the Backlog.

Scrum Meetings are short daily meetings (targeted for 15 minutes) held without fail by the Scrum Team. Each team member is expected to answer three standard questions: What did you do since the last team meeting? What obstacles are you encountering? What do you plan to accomplish by the next team meeting? A Scrum Master leads the meetings and tracks the responses from all team members. This information is the primary source of process measurement, monitoring, and documentation. The key concept behind Scrum Meetings: If you have a function or process that is difficult to predict, sample more frequently to determine where things are going. This provides better feedback for both managers and developers, and it keeps people synchronized with the true progress of the whole team.

Finally, a Demo is the culmination of a Sprint and results in the functionality delivered to the customer. This might result in an informal or formal delivery, or it might simply be a demonstration of the work the team has achieved to date. The key here is that the work in a Sprint must be organized so the end result is a piece of functioning software that shows the new capabilities added in some form.

Scrum does not focus on a large list of specific practices, artifacts, phases, or milestones that would define an entire product lifecycle. It focuses more on defining some enabling practices and patterns that allow you to move quickly and deftly while minimizing the risk that it all will end in

chaos. Scrum is not a fully defined, by-the-numbers process you can use to perform all aspects of software development, but you can combine it with other patterns and processes to complete and complement your entire software process of choice.

Scrum is best suited for companies with highly charged, independent teams of developers. Along with that goes the requirement that management and customers are willing to ease the reigns of control and let the developers work without much supervision and monitoring. Management must be firm in selecting the set of features that comprise the Backlog addressed by a Sprint, and then stick to that set of features until the next Sprint. The remaining Backlog can evolve constantly during the Sprint, but when the next Sprint is started, it too takes a static set of items off the top of the list that becomes the feature set for the next Demo.

If you're looking for a process that produces a lot of documentation and proof of process, Scrum is probably not for you. You could supplement Scrum with extra people to monitor the progress and produce extra artifacts along the way, but it would likely impede the high-velocity progress that Scrum tries to achieve. Scrum has been adopted effectively in a variety of forms and in settings other than software development. A variety of testimonials and background information (see [Resources](#)) can help you decide whether Scrum might work for you.

About the Author

Brian Noyes is an independent software consultant with [Software Insight](#). He is an MCSD with more than 11 years of programming, software development, and project management experience. He is a technical editor and frequent contributor to [Visual Studio Magazine](#) and [.NET Magazine](#). E-mail him at brian@softinsight.com.