WHITE PAPER

Six Things Every Software Executive Should Know About Scrum

Jenny Stuart, VP of Consulting, Construx Software

Erik Simmons, Senior Fellow, Construx Software

Melvin Perez, Senior Fellow, Construx Software

Version 2 1, July 2018

Scrum is the most commonly adopted Agile approach in the software industry today. This paper explores six important things software executives should understand when their organization is considering adopting the scrum framework or improving their current adoption. It includes information about what Scrum can and cannot provide and advice on executive considerations for a Scrum adoption.

Construx

Copyright

Construx Software Builders, Inc.
10900 NE 8th Street, Suite 1350
Bellevue, WA 98004
U.S.A.

This paper is for informational purposes only. This document is provided “As Is” with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Construx Software disclaims all liability relating to use of information in this paper.

Contents

#1 - A Majority of Agile Teams Are Using Scrum

Scrum is a team-level workflow process for Agile software development and is the most commonly adopted Agile approach in the industry today. It is part of the movement and methodologies resulting from, and continuing since, the Agile Manifesto,[1] which was published in 2001.

Agile methodologies were initially developed to deliver value early and be responsive to change. Early adopters were typically organizations with volatile business requirements and in which the end state of the development effort was fluid. Since then, Scrum has been adopted in numerous software and systems development organizations. The 2018 State of Agile Development survey[2] reported that over 70% of teams using Agile choose Scrum or Scrum in combination with Kanban or XP. Hybrid, the use of multiple methodologies, is the next highest percentage at 14%.

Today, many organizations are using Scrum in situations without volatile requirements, on large projects, and on cross-geography projects and in geo-dispersed organizations. If effectively adopted, Scrum helps organizations and teams achieve the balance of predictability and flexibility that supports the business needs well enough to make customer commitments.

[1] Agile Alliance, *Agile Manifesto*, http://agilemanifesto.org (2001)

[2] VersionOne, *12th Annual Survey: 2018 State of Agile Report (2018)*

#2 - Scrum Offers Multiple Benefits

Many organizations are interested in Scrum but wonder what value it can bring to the development organization and to the business. Construx's clients have found that Scrum can enable the adopting organization to do the following:

- Insert high-value features, even late in the development cycle. In the waterfall software development model, all work was completed sequentially. As shown in Figure 1, this resulted in organizations having limited working software until the project was nearly completed. In other words, the business did not realize any of the benefits or value of the project until it had expended all or almost all of the cost to build it.

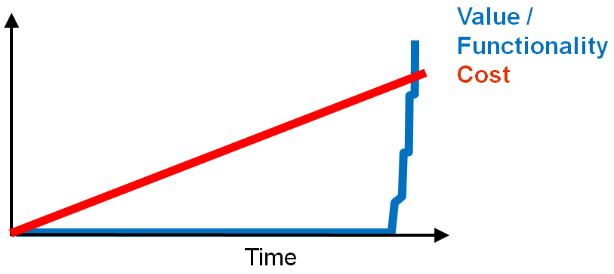

Figure 1 Traditional Project Value Delivery

Scrum enables organizations to realize value throughout the development cycle in short increments. Many Scrum teams front-load the development process so that the most valuable items are delivered first. While this could lead to diminishing returns later in the project, Scrum also permits high value features to be introduced late in the development process. These new, high-value features appear often on Scrum projects; they are discovered through the early use of system features and short feedback loops that characterize Scrum projects. This dynamic is shown in Figure 2.

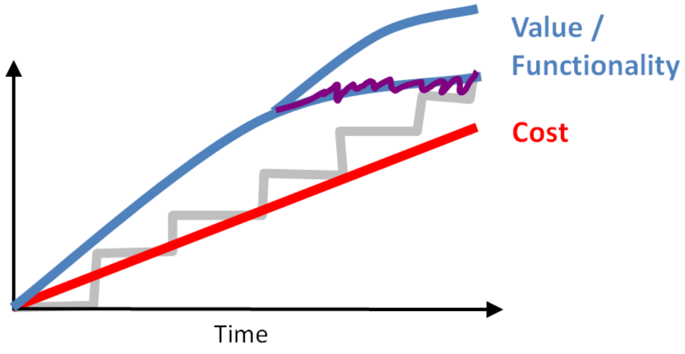

Figure 2 Scrum Enables Delivery Opportunity

- Respond to a dynamic marketplace. Scrum uses one to four-week iterations, explicitly orders the work to be completed, and brings the completed work to a potentially releasable state. These Scrum elements ensure the functionality delivered after every iteration can be released to customers (though Scrum does *not* require releases after every iteration).

  Scrum provides numerous points where higher value customer requests can replace lower priority work already in the queue. In highly dynamic markets, the final feature set delivered to market can be radically different and of greater business value than what the organization initially thought of delivering.

- Measure progress based on delivered functionality. Historically, progress in the waterfall life cycle was monitored by the completion of requirements specifications, design documents, and other activity-based progress. In contrast, Scrum projects are monitored by how much functionality has been accepted as complete by a customer representative called the Product Owner. Scrum provides clear visibility into the quality of the functionality delivered to date, which leads to greater visibility into schedule and delivery risk than is possible with other methods.

- Incorporate customer feedback earlier. The frequent iterations of Scrum provide numerous opportunities to demonstrate, beta, preview, or otherwise expose the customer base to new features throughout a project. In some cases, customer feedback occurs during each Sprint (iteration), since customers can participate in the demos that are an intrinsic part of Scrum. In other cases, the Product Owner, product manager, or marketing representatives gather feedback, either as major features are completed or on a regular heartbeat. Whatever the cycle, Scrum enables more frequent demonstration of working functionality.

- End a project early with delivered value. With incremental delivery, organizations have the option to end a project early and ship with the functionality delivered to date. Our clients often find that they were too aggressive initially regarding what the market really needed, and a less-costly solution provides the vast majority of stakeholder value. Scrum allows this to occur naturally rather than forcing an organization to guess up front about what the market really needs.

Beyond these benefits, Construx has also seen organizations where Scrum adoptions helped to drive improvements in product quality and software development productivity.

#3 - Scrum Is Successful in Diverse Environments

Scrum has proven to be flexible and adaptable to diverse settings. It is being used successfully in the following situations:

- Very large projects and multiproduct organizations
- Geographically dispersed teams
- Commercial, internal, scientific, and embedded systems
- Mission-critical and safety-critical applications
- Regulated industries (for example, industries regulated by the SEC, FDA, and FAA)
- Organizations with stage-gate product development models and PMOs

The success (or failure) of Scrum is all in how it is adopted. For more information, see Construx's *10 Keys to Successful Scrum Adoption*, *5 Most Common Gaps in Scrum Adoptions*, *Staffing Scrum Roles,* and *7 Pitfalls to Enterprise Agile Adoption* white papers.

#4 - Scrum Is Not a Cure-All

While widely used throughout the industry, many organizations have adopted Scrum and have yet to receive all of the desired benefits. It is important for executives to understand the following aspects of an agile transformation. Scrum is *not* any of the following:

- A goal in and of itself. Organizations move to Scrum because they perceive it can bring various benefits to the business. For some organizations, it's the ability to respond to a changing marketplace. For others, it's the improved project predictability that comes from continuously delivering near-production-quality software.

  When adopting Scrum, the executive team needs to understand and communicate what it wants to improve by moving to Scrum. Because of Scrum's history as an approach that responds rapidly to change, development teams can believe that flexibility is what the business wants. In many cases, however, the business doesn't want to trade predictability for flexibility. It is important that everyone have a shared understanding of the goals.
- A Silver Bullet for productivity. While many teams have seen significant productivity gains when adopting Scrum, some organizations report only minimal improvements in productivity. Scrum, instead, provides them with the ability to respond to a changing marketplace, improved product quality, feature-set visibility, or other benefits discussed elsewhere in this paper.
- An excuse for poor practices. Scrum is not an excuse for code-and-fix programming or other bad practices. When an organization adopts Scrum, teams need to ensure their development practices (such as code reviews, Test Driven Development, automated unit testing, and continuous integration) support a rapid delivery cycle. Like other Agile methods, Scrum requires teams to perform many practices much more often than they might in a traditional waterfall environment. Some practices become nearly continuous, such as DevOps or Continuous Integration/Continuous Deployment (CI/CD).
- An excuse to abandon all documentation. When adopting Scrum, the organization needs to determine what documentation is no longer needed, and what must still be produced. For example, are Product Requirements Documents, Functional Requirements Specifications, and other such items replaced by a product backlog or similar Agile artifact?

  The organization needs to ensure that any required documentation (such as architecture specifications, release notes, and test cases) necessary for compliance, effective long-term development work, internal governance, and so on is produced. In most cases, these specifications can be produced incrementally as the work progresses in Scrum.

#5 - Scrum Exposes Everything

According to Ken Schwaber, one of the creators of Scrum, “Scrum exposes every inadequacy or dysfunction within an organization’s product and system development practices. The intention of Scrum is to make them transparent, so the organization can fix them.”[3]

Construx’s experience is that, when used as a team-level workflow method, Scrum alone will not solve team or larger organizational-level issues. The major items to consider are:

- Good requirements still require focus and expertise. Scrum often changes the way organizations capture requirements. For example, they may begin to create user stories, estimate them using story points, and capture them in an ordered product backlog. But using these techniques and practices within Scrum does not ensure teams create high-quality requirements. For example, an organization with challenges in understanding, synthesizing, and selecting the most important requirements across a diverse marketplace will continue to have the same issues after the teams have moved to Scrum.

  Adopting Scrum can be a useful catalyst for making changes in these areas. For example, specific requirements training for Product Owners can occur during the Agile transformation.

- Large projects and organizations need more structure than small teams. Scrum as a team-level workflow method works well in isolation for individual teams or even a couple of teams working together in a single location.

  Today many organizations are scaling Scrum to large teams, teams of teams, and across significant geographic distribution. Large products require additional infrastructure and practices beyond Scrum. For example, a Chief Product Owner can be effective in these settings to establish a vision, create a product roadmap, and manage the top-level backlog for many teams to ensure alignment.

- Appropriate long-range business planning is still necessary. While flexibility and responsiveness to change are hallmarks of Agile and Scrum, effective projects must understand the direction in which they are heading. Most organizations using Scrum still need product visions, product roadmaps, and release plans, though they all take on an Agile flavor. These artifacts provide the larger picture in which Scrum operates. They provide a context for ordering the backlog, making tradeoff decisions, deciding when to pay down technical debt, and so on.

As part of Scrum adoption, an organization needs to monitor the issues that Scrum exposes and make explicit decisions about what it wants to change to better support

[3] Agile Collab, *Interview with Ken Schwaber*, http://www.agilecollab.com/interview-with-ken-schwaber (2008)

incremental and iterative development. The organization needs to make decisions about how Scrum fits within constraints such as compliance requirements, the inability to change the working processes of other parts of the organization, and so on.

For more information, see Construx's *7 Pitfalls to Enterprise Agile Adoption*, *Succeeding with Geographically Distributed Scrum*, and *Managing Technical Debt* white papers.

#6 – Culture Eats Strategy for Breakfast

"Culture eats strategy for breakfast," a quote commonly attributed to Peter Drucker, highlights the fact that an excellent strategy that is incompatible with the organization's culture is doomed to failure from the outset.

Scrum's values are *commitment*, *courage*, *focus*, *openness*, and *respect*. Scrum's approach builds on a foundation of self-organizing teams that choose how to accomplish their work without outside direction (or interference). Effective Scrum requires teams and stakeholders to be open about the work and the challenges of performing it.

Some organizations naturally align with Scrum's values and find it easy to adopt. Other adoptions struggle because the organization's culture has conflicting values.

It is important to recognize that Scrum does not work in isolation. Changes in the software development process also affect interactions with the human resources department, the program management office, product management, and so on. It is the overall organization's culture and values that usually cause the most difficult challenges to robust, successful Scrum adoption.

Advice for Software Leaders

Successful organizational change requires executive-level sponsorship and support. You can help the transformation to Scrum in the following ways:

- Learn more about what is required for a successful Agile transformation. Steve McConnell's *Agile Transformation* videos are an excellent place to start. Beyond that, Construx's Scrum white papers and the numerous books about Scrum are all useful resources.
- Understand and explicitly state the goals for the change to Scrum.
- Create an agile transformation plan.
- Allocate funds for all necessary resources (typically, training and coaching) to support the transformation.
- Ensure that the Scrum roles (Product Owner, Scrum Master, and Development Team) are all appropriately staffed.
- Validate that the new project-reporting techniques provide the necessary visibility into project progress.
- Ensure that the current rewards and recognition structure doesn't undermine the move to Scrum.
- Support the organization's transformation throughout the change. Executives should: lead by example, over-communicate goals, reward early adopters, and use their office, their auspice, their reputation, their budget, and their time to promote and support Agile transformation.

Please contact us for further discussion regarding executive-level concerns to make your transition successful.

Contributors

Jenny Stuart, VP Consulting

jenny.stuart@construx.com
+1(425) 636-0108

Erik Simmons, Senior Fellow

erik.simmons@construx.com
+1(425) 636-0100

Melvin Perez, Senior Fellow

melvin.perez@construx.com
+1(425) 636-0120

Construx

Construx Software is the market leader in software development best practices training and consulting. Construx was founded in 1996 by Steve McConnell, respected author and thought leader on software development best practices. Steve's books *Code Complete*, *Rapid Development*, and other titles are some of the most accessible books on software development with more than a million copies in print in 20 languages.

Steve's passion for advancing the art and science of software engineering is shared by Construx's team of seasoned consultants. Their depth of knowledge and expertise has helped hundreds of companies solve their software challenges by identifying and adopting practices that have been proven to produce high quality software—faster, and with greater predictability. For more information about Construx's support for software development best practices, contact us at consulting@construx.com, or call us at +1(866) 296-6300.