

Motivational Aspects of Scrum: How the Process Drives Focus and Helps Mobilize Creative Energy

By Peter Vaihansky, Senior Vice President, DataArt

In the past few years, **Agile development methodologies** have clearly gone mainstream, with **Scrum being perhaps the most popular Agile process**. Organizations large and small, from market leaders like Google, Microsoft and Salesforce.com to small companies and startups all over the place, are either switching their software development process to Scrum or going to it as their default approach. Since **software development** is a highly social and creative process, it is important to take into account the dynamics of human interaction and human psychology in order to understand how to build better software more effectively.

Unlocking the Motivation

Many Scrum adopters report that switching to an Agile methodology changes the motivational climate in the development team and even the entire organization. Those who experienced a successful Agile transformation often say that they can't imagine ever going back to the old ways of working. As a rule, people are more excited about their work and are doing a better job at it. Why is that? There are very good reasons for this that stem from the very nature of the Scrum process. Let's look at several of the most obvious ones:

- **Everything is public:** Daily Scrums, or standup meetings, leave no room for anyone on the team to "slack off" quietly. Every day, everyone on the team essentially makes a public commitment in front of everyone else, and then reports on their progress. Additionally, Scrum emphasizes pull vs. push, which means that team members volunteer to implement a particular task as opposed to the project manager telling everyone what to do.
- **Everything is timeboxed:** Iterations in Scrum are short, which means that a deadline for something is always near. This provides a much needed discipline, especially in contrast with a large development effort that does not use iterative, incremental delivery.
- **Everything is visualized:** Co-located Agile teams are encouraged to create a "war room"-like atmosphere by using "information radiators": the burndown (or burnup) chart shows the progress of the entire project in time, the Scrum board (often implemented using sticky notes on a whiteboard) visualizes the tasks and who is working on what, architecture diagrams help keep the big picture in front of every member of the team, etc.

The Scrum board, often implemented with yellow sticky notes on a whiteboard, is particularly interesting. Typically the stickies have user stories written on them, along with the name of the person doing them, which makes it possible to physically move the tasks from "Checked out" to "In progress" to "Done". This drives focus and motivation at the same time: it is intensely satisfying to come up to the board, take the sticky with your task on it, and move it to the "Done" column!

Scrum and Flow

Of course, just deciding to "go Agile" does not guarantee that everything is automatically rainbows and unicorns from there on out. There are examples of failed Agile transformations, which often happen when yet another process becomes fashionable and is being implemented for the wrong reasons and/or without sufficient planning of ensuring the buy-in from all stakeholders – but this is the topic for another post.

However, productivity gains from doing proper Scrum can be both very real and very significant. Dr. Jeff Sutherland, one of the most passionate and influential Agile proponents in the world, first introduced the

Scrum process together with Ken Schwaber in a conference paper in 1995. According to one of his [published papers](#), Scrum teams who pay adequate attention to metrics and continuous improvement can achieve a hyperproductive state of up to 400% better than the average waterfall team, with some teams working even faster. In another paper titled "[Shock Therapy](#)", Sutherland states that the best Scrum teams in the world consistently average 75% gains over the velocity of waterfall teams with much higher quality, customer satisfaction, and developer experience.

Some of the potential reasons for these productivity gains start with "flow state". Flow, a concept that has gradually become popular in psychology over the last couple of decades, describes a certain state of heightened awareness and performance. When you achieve the flow state, according to the psychologist [Mihaly Csikszentmihalyi](#), who coined the term, "your whole being is involved, and you are using your skills to the utmost."

In his new book, "The Rise of Superman: Decoding the Science of Ultimate Human Performance," Steven Kotler discusses flow triggers – circumstances that facilitate the entry into the flow state. Kotler groups the 17 flow triggers into several buckets: psychological, environmental, social and creative. Since we are attempting to explain [team performance](#), our interest lies primarily in the social triggers, defined as "ways to alter social conditions to produce more group flow". It is uncanny how Scrum is almost precisely designed around these triggers. Could this be the reason why, as Scrum evangelists claim, you can get more done faster, with fewer people, and with higher quality, effectively breaking the "iron triangle"?

Most of Kotler's social and environmental flow triggers map closely to Scrum practices or principles. Let's look at the ones that fit:

- **Serious concentration:** To create flow in social settings, you need to ensure everyone has their maximum attention to the here and now, and is blocked off from other distractions. Scrum practices frozen scope for timeboxed iterations, i.e. no changes in the team's direction are allowed during a sprint. This is done in order to create the necessary period of uninterrupted, focused work, which is precisely what the teams need to achieve higher levels of productivity.
- **Shared, clear goals:** Groups need to be clear about what their collective goal is in order for flow to happen. Practices described above, like information radiators and visualizing things like scope, progress and task status serve exactly that purpose.
- **Good communication:** This one is almost self-explanatory. [Constant communication](#) is necessary for group flow, according to Kotler. Scrum practically forces constant, open, intense many-to-many communication, where everyone not only has a voice, but is fully expected to use it.
- **Familiarity/common language:** Kotler stresses the importance of common language, a shared knowledge base and a communication style based on unspoken understandings. This is almost precisely the description of the emergent tacit knowledge emphasized by all Agile methods.
- **Equal participation:** Flow is most likely to happen in a group setting when all participants have an equal role in the project. This is reflected in the cross-functional and self-organizing nature of Scrum teams, as well as such practices as collective code ownership (which stems from Extreme Programming but is widely used by many Scrum teams).
- **Risk:** There's no creativity without failure, and there's no group flow without the risk of failure. The group has to have some skin in the game to produce group flow. Scrum teams are constantly making clear commitments to their customer (internal or external, in the case of outside partner teams) and there is always the risk of underdelivering on those commitments.
- **Sense of control:** Kotler defines sense of control as the combination of autonomy (being free to do what you want) and competence (being good at what you do). Again, this is precisely how Scrum is set up: competent, highly motivated, self-organizing, cross-functional teams that are not micromanaged, but rather choose their own implementation path and are accountable to the customer for the ultimate results.

The flow state is a fascinating subject, but even more interesting to me are its applications in the world of work, not just in professional athletics or extreme sports. I hope researchers continue to study the secrets of productivity and ways to **build software** more effectively. Software already runs almost all aspects of our everyday lives, and we are likely to depend even more on it in the years to come, so we will all benefit from these kinds of research.

Original article — <http://www.nearshoreamericas.com/motivational-aspects-scrum-process-drives-focus-helps-mobilize-creative-energy/>