Shock Therapy
A Bootstrap for Hyper-Productive Scrum

Jeff Sutherland, Ph.D.
Scrum Training Institute
Boston, MA USA
jeff@scruminc.com

Scott Downey
MySpace
Beverly Hills, CA, USA
Practical.Scrum@gmail.com

Björn Granvik
Jayway
Malmo, Sweden
bjorn.granvik@Jayway.se

***Abstract*—A properly implemented Scrum framework enforces a few simple constraints that cause a team to self-organize into a state that achieves 5 to 10 times waterfall performance. Yet the majority of Scrum teams never achieve this design goal. Teams do not know how to sequence work to deliver working software at the end of a sprint. They do not know how to work with a Product Owner to get the backlog in a ready state before bringing it into a sprint and do not know how to self-organize into a hyper-productive state during a sprint. A pattern is emerging at MySpace in California and Jayway in Sweden, for bootstrapping high performing Scrum teams. Rigorous implementation of Scrum by an experienced coach creates a total immersion experience akin to Shock Therapy. Teams are trained on exactly how to implement Scrum with no deviations for several sprints. These teams consistently achieve better than 240% improvement in velocity within a few weeks. They are then able to self-organize on their own to continue to improve performance. For many developers and managers, the experience is a wake up call to agile awareness. Unfortunately, management tends to disrupt hyper-productive teams by disabling key constraints in the Scrum framework. Team velocity then falls back into mediocrity. Velocity data is provided on five hyper-productive teams at MySpace and one team at Jayway. In all but one case, management "killed the golden goose."**

Keywords-agile, scrum, hyper-productivity, shock therapy

I. Introduction

The average Scrum team delivered a 35% improvement in velocity at Yahoo [1] where teams properly coached delivered 300-400% improvements. The best Scrum Master at MySpace peaked at 267% of initial velocity after 12 weeks and averaged 168% increase in velocity over 12 Sprints. Most teams were less successful.

We define Hyper-Productivity here at 400% higher velocity than average waterfall team velocity with correspondingly higher quality. The best Scrum teams in the world average 750% gains over the velocity of waterfall teams with much higher quality, customer satisfaction, and developer experience. We have see this in the U.S. [2], Russia [3], the Netherlands and India [4], and from Software Productivity Research data on agile teams [5]. The problem addressed in this paper is what to do about the 90% of Scrum teams that never deliver this capability.

II. Problem Statement

The experience here is from MySpace in California and Jayway in Sweden. Scott Downey is an experienced Scrum coach at MySpace that took on the role of ScrumMaster in the five teams discussed in this paper. He has been designated by MySpace management as the Agile Coach for the company where most of the teams are waterfall or partial Scrum implementations with project leaders.

At Jayway, detailed data from one team is available on a project with a large telecomm company. The Scrum Master was Fredrik Källbäck from Jayway assisted by Jayway seasoned programmer/architect, Adam Skogman. Three developers from a competitor consulting company were on their team.

Experience with previous Scrum teams convinced these coaches that Scrum was an interrelated set of parts where the whole has much more value than any part. Early Scrum teams with partial implementation of agile practice achieved modest gains. Here they want to achieve the design goal of Scrum – hyper-productivity.

Björn Granvik, CTO of Jayway, implemented the model described here with two other teams for which data is not currently available. Results were comparable.

A. *Scrum is an Ecosystem*

Experienced agile coaches recognize that Scrum is based on complex adaptive systems theory. It is not a methodology, process, or procedure. It is a framework based on enforcement of simple constraints that will cause a average team to self-organize into a hyper-productive state [6].

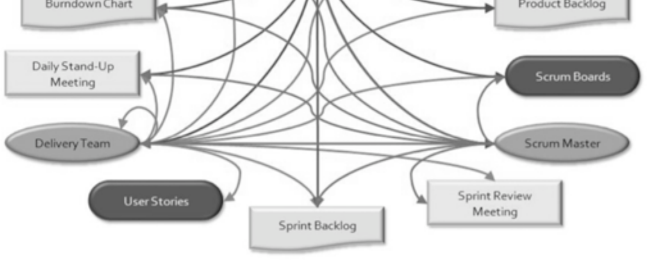

Figure 1. Scrum is an ecosystem.

Any system will settle into the lowest possible energy state. Consider the water in a toilet. It is without motion and flat. When you flush the toilet you introduce energy into the system and enforce constraints which cause the water to swirl into the same motion every time. As soon as the energy input stops, the water returns to a flat and motionless state.

The difference between the highest and lowest performing software development teams is 1:2000 [7]. This is more than two orders of magnitude greater than the difference between the best and worst developer on a project [8]. The average software development team is in a placid state where velocity is slow, quality is low, customers are unhappy, and management is upset. We want to introduce energy into the team and enforce constraints that systematically product high velocity, high quality, happy managers, and ecstatic customers.

The Scrum meetings are designed to raise the communication saturation level of a team in order to align their focus and facilitate team spirit. This introduces an energy flow into the system which is constrained by the ordering of the product backlog, the required ready state of user stories, a strong definition of done, and continuous process improvement through removal of impediments. Velocity of the team, quality of the software, satisfaction of the users, and revenue for the company will always increase several hundred percent if communication saturation goes up and Scrum constraints are properly enforced. Waste will be flushed from the system and the team will go from strength to strength.

When implementing Scrum, it is therefore essential to understand Scrum as an ecosystem of interdependent parts. Each of the three Scrum roles (Product Owner, Scrum Master, and Team) is dependent upon every meeting, artifact, and best practice in the ecosystem. If any part of the ecosystem is dysfunctional the whole system deteriorates to mediocre gains in performance and quality.

B. *Strategies for Implementation*

Most previous Scrum implementations at MySpace and Jayway were based on “Team Discovery” or “Novice Leadership.” Individuals read some books, get some training, and start implementing Scrum while inspecting and adapting. This leads to a hybrid implementation of Scrum where key pieces are deemphasized or missing (ScrumBut [9]). For example, informal surveys show that 50% of the Scrum teams worldwide cannot get software tested at the feature level by the end of a Sprint violating the second principle of the Agile Manifesto. This creates increased rework and poor performance.

MySpace and Jayway needed a way to rapidly start up a new Scrum team where important constraints were enforced to consistently deliver hyper-productivity in a short period of time. Here, we present an approach that works in waterfall environments even with minimal management support.

The difference between this and “ordinary” Scrum is that:

- Coaches did not wait for teams to self-organize on their own. The method and technology was firmly established. The team learned to self-organize while following these constraints.
- Acceptance Test Driven Development (ATDD) was used. Testers/business analysts would deliver test cases that were implemented directly by the programmers. Only after this was the actual code completed. Testing was accomplished as soon as possible after code completion and before the end of the sprint.

The need for ATDD as a best practice has been carefully documented by Systematic Software Engineering [10, 11]. As a CMMI Level 5 company, they have developed the most comprehensive data available for hundreds of teams showing that ATDD will consistently double velocity and reduce defects by 40% in a company that already has one of the lowest defect rates in the world.

III. Shock Therapy Recipe

In order to cut standard Team Discovery or Novice Leadership bootstrapping times by 50% or more, the following steps were used at MySpace when orienting the teams into a proper Scrum posture. These steps can be easily implemented by an experienced coach. A new ScrumMaster needs to be aware that these steps are critical for achieving high performance of teams.

For a novice Scrum Master, failure to implement these steps will consistently incur the cost of poor velocity and quality. The results here can show the novice Scrum Master which are the important features of Scrum that must be implemented to guarantee high performance. Novices will have to do their best to convince teams to follow best practices. For teams in this paper, the Scrum Masters had enough experience and management support to enforce the right practices and the leadership capability to get the teams to cooperate. With the right coach, resistance is futile.

A. *Lay the Foundation*

Novice Leadership and Team Discovery approaches at MySpace and Jayway allowed the teams to become distracted by new terminology, roles and artifacts. In the absence of strong, experienced leadership, most teams spent their formative months focused on aspects of the framework rather than on delivering value to the customer. They also under-emphasized or failed to implement critical elements of the Scrum Framework, which sets them up for limited success at best.

These mistakes often led to a measurable initial *reduction* in value delivery rather than the expected increase that drove the decision to implement Scrum. To avoid this pitfall, the Shock Therapy coach, Scott Downey, fully enforced the complete Scrum Framework for teams described here.

Scott found it was critical at the outset that the entire team participate in training so that everyone had the same understanding of goals, mechanisms, definitions, and responsibilities they will share going forward. Teams in this study have participated in an internally developed *Introduction to Scrum* course that covers twelve key points Scrum as well as the most impactful environmental factors of the MySpace technical and organizational structures. Until the Scrum Product Owner, Scrum Master, and entire Delivery Team participate in training, no further steps were taken to bootstrap that team.

B. Stabilize the Environment

The legitimate degrees of freedom in the Scrum Framework are often confused with Framework elements themselves. This can lead to accidental, dysfunctional hybrid models. Having a strong, experienced, and empowered ScrumMaster is critical to getting teams functioning quickly and realizing the benefits of Scrum. To achieve this, the Shock Therapy coaches at MySpace and Jayway take many of the legitimate degrees of freedom off the table by providing an additional but temporary structure that could be viewed as a "Default Profile" for new teams. Through practice and demonstrated proficiency, teams earn the right to change these Default Profile settings (but never the Scrum Framework).

Before changes in the Default Profile could be made, the teams in this study were required to complete three consecutive, successful Sprints, demonstrate a 240% increase in Velocity, and have a solid business reason to make a change that was agreed to by all team members.

Default Profile rules were applied consistently for the MySpace teams in this paper. At Jayway the same conceptual approach is used with minor variations.

- Set Sprint Length

The Shock Therapy Coach decides Sprint Length. Shorter Sprint lengths are recommended to facilitate more rapid inspect/adapt cycles. All teams in this study used one week Sprints.

- Set the Definition of Done

The Shock Therapy Coach provides an initial definition of Done that should be applicable to 80% of the work the team will pursue. Our initial definition of Done includes, at minimum:

- Feature Complete
- Code Complete
- No Known Defects
- Approved by the Scrum Product Owner
- Production Ready

Although approval of delivered work is the domain of the Scrum Product Owner, during the Shock Therapy experience, the Coach must also agree that the work has met the agreed state of completion or s/he, too, can reject the work and direct it back to the Product Backlog.

- Strictly Filter User Stories

Only properly formed and supported User Stories are allowed into the Sprint by the Coach. Improperly formed Product Backlog content is rejected by the Coach on the team's behalf before the Planning Meeting.

- Sprint Backlog Items

Sprint Backlog items are accepted at the highest level of granularity that passes the INVEST mnemonic [2]. At no point are cards broken into a list of tasks in pursuit of a task list alone.

- Only Estimate in Story Points

Estimation is in Story Points only. No estimates in Hours are ever solicited or tracked, and team members are discouraged from thinking of tasks in terms of time.

- A Physical Scrum Board Must Exist

A physical Information Radiator is designed by the Coach and serves as the focus of the daily 15-Minute Stand-Up Meeting. The simplest board with the minimum number of columns is recommended. Teams in this study used boards that displayed only Product Backlog, Sprint Backlog, Work In Progress, and Done. No "Waterfall" columns (e.g. Design, Dev, QA) were allowed. A physical board will be maintained even if software tools are used to provide visibility to remote locations.

- Respect for Team Meetings

A penalty for tardiness or unexpected absence from any team meeting is agreed to and enforced by the team. It applies equally to all team members, regardless of rank, role or excuses.

- The Sprint Planning Meeting Length

The Sprint Planning Meeting will be 5-10% of the Sprint length in duration, and will include Sprint Review, Retrospective, Product Backlog Presentation, Estimation and Commitment of the Team.

C. Building Muscle Memory

During the Sprint, the Coach needs a singular focus on adherence to the Scrum Framework and Default Profile rules. It is best that s/he not be distracted by feelings of ownership over either the Product or the code. S/he must prevent multi-tasking, enforce working in priority order, encourage collaboration on the highest priority, and maintain the Scrum Board until the Team takes these things over, which usually happens naturally in the first several Sprints.

The Coach must constantly explain both rules and rationale used to derive advice or correction. As an example, when a lower priority card completes and the team member asks for more work, the Coach should not just advise, "Take the top item from the Product Backlog." Rather, s/he should step them through the logic. "Can you help expedite any card that is already in progress? If not, can you work on any of the committed cards that are not yet started? If not, is there a better way to redistribute work across the team based on your availability right now? If not, retrieve the highest priority item from the Product Backlog and commit only to as much as can be completed by the end of the Sprint."

It is important to engage the team in problem solving rather than always solving the problems yourself as a Shock Therapy Coach. When the Coach notices someone multi-tasking, a team member not paying attention during the meeting, an Information Radiator that is not moving properly, or any other systemic or behavioral sub-optimizations, s/he should ask the team if they notice anything happening that should not be happening. Ask them to find and correct the defects and be available to help them if they begin to fail.

D. Plan Your Exit Strategy

At no point during Shock Therapy can the Coach become personally involved or vital to the team's success beyond the bootstrapping experience. The Coach must not take on any fundamental tasks or fill in for any missing team members. It is critical to remember that the purpose of a Coach is to create self-sufficiency within the team. S/he must not become a foundational element of it and should seek to relinquish authority, leadership, and artifacts as soon as the team demonstrates an ability to absorb them.

IV. Results from MySpace

Here we have data on five teams from MySpace in California and one team at Jayway in Sweden using Shock Therapy. Teams at MySpace were implementing a web framework and tools to support hundreds of millions of users building their personal web pages. The team at Jayway was at a large telecom company producing mobile phone infrastructure.

A. Establishing Baseline Velocity

The baseline velocity (100%) is established for a team during the first Sprint. The Product Owner presents the prioritized Product Backlog in the Sprint Planning meeting. This is estimated using Planning Poker and story points [12]. The team selects what can be accomplished during the Sprint and the Product Owner determines exactly what is "Done" at the end of the Sprint. The number of story points completed is the baseline velocity. At MySpace, the baseline velocity is often significantly higher than their previous chaotic implementation of waterfall, so the baseline is conservative.

B. MySpace Team Data

Data on five teams at MySpace is summarized in Figure 2. The solid curve in the middle of the graph is average velocity for all teams for each Sprint. The upper and lower curves show the maximum and minimum achievement from the data.

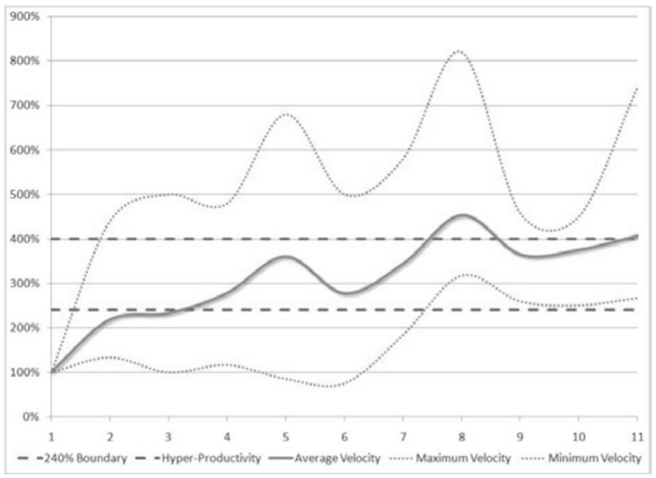

Figure 2. Velocity of MySpace Teams by Sprint

The lower dotted line is 240% percent of baseline velocity and the goal at MySpace was to achieve this in three one-week Sprints. Teams that achieve this typically go over 400% (upper dotted line) into a hyper-productive state in later Sprints. The low data points were from the only team in this data set where the MySpace Agile Coach did not assume the ScrumMaster role. The existing Scrum Master failed to enforce constraints.

V. Results at Jayway

A team at Jayway in Sweden achieved the same effect using a similar strategy. A team doing two week sprints achieved 375% of initial velocity in six sprints. During the seventh sprint, management started removing resources to support a late waterfall project with approximately 100 people. These desperate attempts by management to add bodies to a late waterfall project have repeatedly been shown to cause further delays [13]. A competent management team would move functionality from the waterfall team into the hyper-productive Scrum team to get it done faster.

An interesting finding from the Jayway experience occurred six months later when the Scrum team was reunited. The team immediately achieved the high performing state they had accomplished previously indicating that hyper-productivity is team learning. It is as if the team learned to ride a bicycle together. Once they learn how to do it, even if disbanded, they can repeat it at a later date when they come together. This demonstrates the important of stable teams for high performance.

VI. Conclusion

Here we introduce a successful model for developing high performance Scrum teams implemented at MySpace in California and Jayway in Sweden. A forcefully and fully implemented Scrum led by an experienced coach can bootstrap a team into a high performing state in a few Sprints. The model discussed is a useful reference for novice ScrumMasters, showing them the key points of leverage for bootstrapping a new team.

The MySpace model is implemented in the midst of a company with variable processes and little management support (no process, some waterfall, a lot of ScrumBut, and a few high performing Scrum teams). It demonstrates that the model can work in any company with a good coach and will rapidly disintegrate under bad management. The Jayway model was implemented by a Scrum consultant in a leading telecomm company showing it can work in complex product development with embedded systems.

The value of the model is that it shows both that teams can consistently achieve a hyper-productive state and that disruptive environments will consistently destroy hyper-productivity. Yet teams can resurrect themselves given the right opportunity. These data provide management results on which to base a clear choice for performance over mediocrity in software development.

VII. References

[1] G. Benefield, "Rolling Out Agile at a Large Enterprise," in *HICSS'41, Hawaii International Conference on Software Systems*, Big Island, Hawaii, 2008.

[2] M. Cohn, *User Stories Applied : For Agile Software Development*: Addison-Wesley, 2004.

[3] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in *HICSS'40, Hawaii International Conference on Software Systems* Big Island, Hawaii: IEEE, 2007.

[4] J. Sutherland, G. Schoonheim, and M. Rijk, "Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams," in *Agile 2008*, Toronto, 2008.

[5] C. Jones, "Development Practices for Small Software Applications," Software Productivity Research 2007.

[6] M. Beedle, M. Devos, Y. Sharon, K. Schwaber, and J. Sutherland, "Scrum: A Pattern Language for Hyperproductive Software Development," in *Pattern Languages of Program Design*. vol. 4, N. Harrison, Ed. Boston: Addison-Wesley, 1999, pp. 637-651.

[7] L. Putnam and W. Myers, *Industrial Strength Software: Effective Management Using Measurement*: IEEE, 1997.

[8] J. Spolsky, "Hitting the High Notes," in *Joel on Software* New York: Fog Creek Software, 2005.

[9] K. Schwaber, "Scrum: It's About Common Sense," 2009.

[10] J. Sutherland, C. Jacobson, and K. Johnson, "Scrum and CMMI Level 5: A Magic Potion for Code Warriors!," in *Agile 2007*, Washington, D.C., 2007.

[11] C. Jakobsen and J. Sutherland, "Scrum and CMMI – Going from Good to Great: are you ready-ready to be done-done?," in *Agile 2009*, Chicago, 2009.

[12] M. Cohn, *Agile Estimation and Planning*: Addison-Wesley, 2005.

[13] F. P. Brooks, *The Mythical Man Month: Essays on Software Engineering*: Addison-Wesley, 1995.