

# Take No Prisoners

## How a Venture Capital Group Does Scrum

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

Igor Altman  
OpenView Labs  
Boston, USA  
ialtman@openviewpartners.com

**Abstract**—In 2007, OpenView Venture Partners decided to adopt Scrum as best practice in software development in its portfolio companies and Scrum as the standard practice in internal operations. It is one of the first high-performance non-software Scrums that delivers twice as much value in fewer working hours. The model at OpenView provides data and a working manual on how to do Scrum outside of software development. Their aggressive removal of impediments (take no prisoners!) distinguishes them from Scrum implementations that are unable to remove institutionalized waste.

*Keywords-agile, scrum, venture capital*

### I. INTRODUCTION

This paper introduces a successful model for implementing Scrum in OpenView Labs, a division of OpenView Venture Partners that supports a growing number of portfolio companies for a venture capital group. The teams do not do software development. They help implement best practices in management, sales, marketing, finance, development and customer support for portfolio companies. Their Scrum implementation is repeatable, proven across many projects, and recommended for teams in all areas of business.

#### A. The Context

OpenView Venture Partners was founded in the Fall of 2006 with nine team members and an initial \$107M investment fund. A new kind of venture capital fund, OpenView provides hands-on operational value to each portfolio company in addition to investing in the best software companies and taking seats on their Boards.

By third quarter of 2007, the OpenView Venture Partners core team grew to thirteen professionals, and an investment portfolio of six software companies. The new portfolio companies brought high demands for value-add projects from the operational experts. Long work hours that extended into late nights and weekends became common. The combination of new value-add projects, unprecedented deal activity, and new team members resulted in OpenView's founder, Scott Maxwell, having to spend increasing amounts of time managing individuals and their projects within a flat hierarchy.

OpenView decided to shift a team member from the investment effort to create a new organization, OpenView

Labs, focused entirely on value-add and due diligence. This helped, but the answer to scaling OpenView over the next 12 months while REDUCING the time Scott needed to devote to managing people and INCREASING the value-add to each portfolio company remained elusive.

#### B. OpenView Meets Scrum

OpenView was familiar with Agile Development since its investment in VersionOne ([www.versionone.com](http://www.versionone.com)), an Agile Project Management tool company. Most OpenView team members thought Agile relevant only to software development and never considered it for internal consumption. This perception changed after Jeff Sutherland [1] met with Scott Maxwell. In Scrum, Scott saw simple principles and ideas that could be applied to make OpenView more productive and self-managed. A few months after the meeting, a Scrum Master Certification course for everyone completed the introduction of Scrum into OpenView Venture Partners.

Here we describe two phases of Scrum adoption by OpenView:

- Phase I – Getting Started with Scrum
- Phase II - Scaling up Scrum

Phase III is being implemented at the time of writing of this paper and radically reorganized the Scrum teams. High team velocities caused the Product Owner function to be a bottleneck and this will be a topic of a future paper.

### II. GETTING STARTED WITH SCRUM: TRIAL BY FIRE

Phase I of our Scrum implementation started during one week in January 2007.

#### A. Team structure

Two Scrum Teams were formed, the OpenView Labs value-add team and the deal team that searches worldwide for new company investments. (NOTE: The rest of this case study will focus only on the OpenView Labs team)

Four people formed the Labs team and three people were on the Deal team. Each team had its own ScrumMaster, though in both cases the Scrum Master is a key member of the team and is in the critical path working on Sprint Backlog.

Sprints were established at one week lengths, from Monday through Friday.

### B. Product Ownership & Backlog

Scott Maxwell, the founding partner, is Chief Product Owner for OpenView and the Labs Product Owner, although the Scrum team is effectively a Product Owner Scrum Team and generates the majority of the stories, with the Chief Product Owner adding a minority of the stories and prioritizing the entire backlog.

The Labs team has a story backlog housed in an online spreadsheet in Central Desktop ([www.centraldesktop.com](http://www.centraldesktop.com)), a hosted collaboration tool and an OpenView portfolio company. The backlog is a collection of each team member's individual projects in progress.

The OpenView Labs Scrum team focuses on three overarching objectives:

- Execute operational value-add projects for OpenView's portfolio companies. These projects are coordinated by OpenView senior point people who also sit on the companies' Boards and are outside of OpenView Labs.
- Execute due diligence on OpenView prospect portfolio companies. These projects are coordinated by the deal team.
- Execute projects to institutionalize and build out OpenView's value-add capabilities. These projects are coordinated directly by the Chief Product Owner.

Value-add projects for each portfolio company are discussed at the Monday morning Investment Committee meeting that includes all the portfolio company OpenView senior point people and Labs senior team members. Labs team members both add projects and receive them.

### C. Stories

Stories are vague and unclear in this early phase of Scrum adoption. Definition of done is different for every story and is typically unclear.

Understanding the specifications of each story takes a lot of back-and-forth during the week between the Labs team members, portfolio companies, the senior point people, and Scott Maxwell, our Chief Product Owner.

### D. Sizing and Planning

The Labs team has a Sizing meeting every Monday where stories for the next sprint are sized and unclear stories are sent back to the product owners.

Sizing is done in 'perfect hours'. This is the amount of hours it should take an average team member to complete the story under ideal conditions with no interruptions. A perfect hour is only counted as done if the entire story is done.

Perfect hours are used instead of points because:

- They are conceptually easier for the team to understand
- A significant number of Labs stories are time-based, like meetings, calls, and time-boxed research stories

This measurement creates an ongoing problem in determining velocity as there are no standard reference stories that are stable in size.

Sizing is done in a fashion similar to planning poker. On a count of three, each team members indicates using their hands how many 'perfect hours' they think a story is, and differences are discussed. No story should be more than two days of work.

After sizing, Scott prioritizes the stories, and the Labs team indicates which stories it plans to get done this sprint. The Labs team then plans the stories, breaking them down into tasks, identifying dependencies and impediments, and team members decide which stories they will do.

### E. Daily Scrums

Each team has a daily Scrum.

In the meeting, the Scrum Master asks each member:

- What did you do yesterday?
- What will you do today?
- What are your impediments?

Everything is run off of the online spreadsheets in Central Desktop.

### F. Retrospective

A retrospective is held every Monday morning to review last week's sprint. The team discusses what went right, what went wrong, and determine what the team can do better.

### G. Velocity

Initial velocity is estimated by assuming that each team member can take on approximately 20 perfect hours per sprint, giving Labs a starting velocity of 80. Thus, velocity is really equivalent to a focus factor of 50% for a 40 hour work week [2]. Unplanned stories are tracked separately.

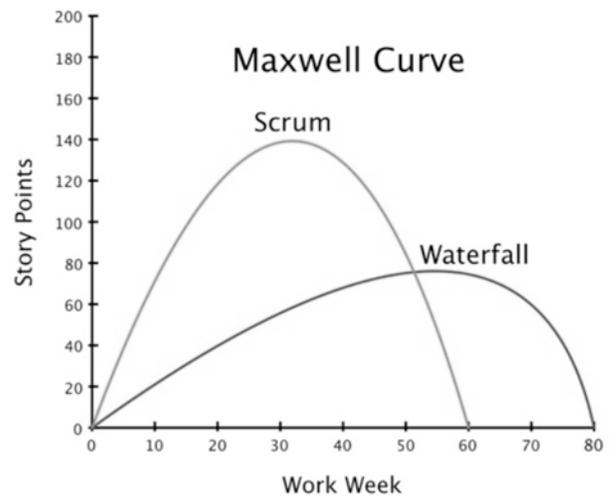


Figure 1. Scrum gets more done with less work

Velocity stays relatively flat at around 80 from sprint to sprint. The Labs team takes on too many stories and does not complete all tasks in Sprints successfully.

The team and Scott recognize that using perfect hours instead of story points means that the measured velocity can only increase if:

- The team removes impediments that increase its focus factor on the stories (i.e. spend less time on overhead and more time getting high impact work done)
- The team works more hours

Furthermore, improvements and impediment removals that decrease the amount of time spent on individual stories simply shrink the story size and increase the amount of work the team can do in the same amount of time without increasing the velocity. Yet the team is comfortable with these issues and continues to use perfect hours.

Within months, as the team's productivity improves and more gets done in less time, Scott initiates a culture of working fewer hours and no weekends. Within a few months, our founding partner had discovered that Scrum can double output by working less. He introduced the concept of the Maxwell Curve.

Investors typically drive companies to work hard and long as the peak of production in a traditional company requires more than a 40 hour week. Our teams were typically working over 50 hours a week, working late and on weekends. But Scott noticed that Scrum is intense. The peak of production is less than 40 hours a week. So he instructed the teams to work at a sustainable pace and avoid night and weekend work. At the same time, he set the expectation that production would double.

Ultimately, the focus on higher velocity in perfect hours with less actual hours worked drives the team to remove overhead, communication gaps, and increase story clarity to increase the number of perfect hours completed.

#### *H. Benefits*

Benefits of the Scrum implementation, though far from perfect, emerge immediately. The team is now self-managing, and Scott's time dedicated to managing individual Labs team members and projects drops. Communication within the Labs team goes from virtually zero to a significant level. Communication saturation is a key indicator of productivity [3] on teams and a strong feature of Scrum. Once each team member's projects become transparent to the team, about 30% of projects are seen as low value and eliminated, making more room for high value projects.

Many impediments emerge and are removed. Early on, each impediment is easy to remove and removal has a high impact on improving the productivity and quality of life of the team members. Common impediments include lack of clarity, lack of communication, and low value work. A common issue is that a given project is producing lots of outputs that no one cares about and is not producing the few outputs that people do care about. This issue is resolved by the Scrum team clarifying each project, and the Scrum Master scheduling calls with stake holders of each project and zeroing in on its exact requirements.

While team members are still working as individuals, they start helping each other, and some collaboration develops. Team members who were overwhelmed with working 10-20 projects simultaneously see a reduction in stress because the projects are now broken up into smaller stories with an online Scrum board, making things easier to manage, and low value projects go away

#### *I. Challenges*

Labs team members are still working mainly as individuals. Team members are specialized and lacking

cross-training. The daily Scrums are taking up to 45 minutes for a four member team. It is increasingly difficult for a four member team to both be product owners and executors. The work days are intense, and long hours start to become exhausting.

Communication between the Scrum team and its project stakeholders—the portfolio companies and the OpenView senior point people who manage those portfolio companies and sit on their Boards—is poor. It is unclear how the stories for any given portfolio company fit together. The Scrum team is unsure of the true impact of the stories.

While all team members were thrilled with help on their projects and other Scrum benefits early on, some are starting to show discomfort with functioning as members of a team rather than as individuals. Transparency and management by the team, as well as the existence of a Scrum Master, start to become sources of conflict, especially for one team member.

Several team members are still unsure that Scrum, which was designed for software development, makes sense for the work at OpenView Labs. They embrace some aspects of Scrum but reject others and prefer to stop the implementation where it is at that time.

The ability to take large projects and make them manageable by breaking them up into smaller stories creates the temptation to take on many projects. While early on the Labs team takes on 80 points and lands, as the team tries to push itself, it stops landing the sprint. It falls into a habit of taking on stretch goals and not landing, and velocity remains flat sprint to sprint.

### III. SCALING UP: FINDING THE RHYTHM

Once most of the basics of Scrum are in place, the Labs team experiments with the different components.

#### *A. Team Structure*

As the team grows from four to six and ultimately nine members, it becomes too large to be efficient. All meetings, from the Sizing to the Daily Scrums, are taking too long. It is increasingly difficult for any team member to be heard in a large group

The team splits into two and the team is allowed to decide how it will split. Members organize into two teams whereby more functionally specialized team members are on one team and the newer team members are on another. A new Scrum Master is appointed to the more specialized team

#### *B. Sprint Length*

After experimenting with two-week sprints, the Labs team and Investment Committee settle back on one-week sprints. In two-week sprints, the Scrum team found it more difficult to stay focused on the key goals and the number of unplanned stories grew.

#### *C. Product Ownership & Backlog*

Scott is the Chief Product Owner across the firm. Product ownership for portfolio value-add projects is shifted out of the Labs Scrum team and is formalized as the role of OpenView senior point people.

Each portfolio company now has its own backlog, housed in an online spreadsheet in Central Desktop. Each backlog is discussed in the Investment Committee meeting where priorities are reviewed and stories clarified. The Labs Scrum team still provides feedback into the backlogs but now every story and any changes must go through the senior point person.

Projects to institutionalize and build out OpenView's value-add capabilities are formalized. OpenView's value-add capabilities are organized around Practice Development Areas, by function (i.e. Sales, Marketing, Customer Service, R&D, etc.)

Each Practice Development Area has a point person, or product owner, in the Labs team and its own backlog, housed in an online spreadsheet in Central Desktop. Every Monday, Scott holds a meeting where every Practice Development backlog is reviewed, prioritized, and clarified.

Labs team members now wear their Product Owner hats when preparing the Practice Development backlogs, speaking with the senior point people about the Portfolio company backlogs, and in the Monday Practice Development and Investment Committee meetings. They wear their Scrum team hats the rest of the sprint.

#### *D. Stories*

Stories are now relatively clear. Less communication during the sprint is required between the Labs team and the portfolio companies and the senior point people / product owners.

Definition of done is now specified and the same for every story. The story is done when the deliverable (whether an analysis or notes from a call or meeting) is posted to Central Desktop and all the stakeholders (the point people/product owners) are notified. Next steps are specified. This definition of done was created in response to the Labs team completing stories but not communicating effectively back to the point people what had been done and what remained to be done in the initiative.

#### *E. Sizing and Planning*

Sizing and Planning is now done in the Scrum Room, whose walls have been turned into product backlogs and a Scrum board. While all stories start in an online spreadsheet on Central Desktop, they're written on post-it notes and end up on the walls.

After experimenting with having a Sizing meeting on Thursdays, the Labs team settles back on a Sizing meeting every Monday primarily because the backlog changed too much between Thursday and Monday. Sizing continues to be done in the same fashion as before, using fingers on hands instead of planning cards.

For a period of time, the Labs team tries to transition to sizing stories in Story Points rather than perfect hours. After a while the meaning of a Story Point becomes unclear, and it seems like they are just another time for perfect hours. The team goes back to using perfect hours.

In discussions with Jeff Sutherland, the reasons behind the confusion become clear:

- Many Labs stories are small enough to be Tasks, which in Scrum are sized in hours.
- Many stories are time-based. For example, "A one hour call with Jim to discuss marketing plan." Many research type stories are also time boxed.

As a result, it's difficult to size a large part of the backlog in hours while sizing another part in relative story points. No changes are made to the way prioritization and planning are done.

#### *F. Daily Scrums*

Daily Scrums now last only 15 minutes, with the same three questions asked.

#### *G. Retrospective*

Plenty of impediments still surface in the weekly Retrospective. There is now an impediment backlog if the impediment cannot be removed immediately.

#### *H. Velocity*

Velocity for a sprint is now set by the team's velocity for the previous sprint. After many failed sprints, the team is now landing almost every sprint.

Through trial and error, the Labs team has discovered that the more it takes on beyond the previous sprint's velocity, the less it actually gets done during the current sprint and velocity falls. When the team takes on less, it gets more done and pulls forward.

While calculating velocity in perfect hours instead of story points makes it difficult to use velocity as the sole gauge of improved output productivity, combining perfect hours with other metrics and a few conservative assumptions demonstrates significant productivity improvement in comparison to early 2008.

Velocity in perfect hours has risen from approximately 80 for four team members each working 50-70 hours per week in January of 2008 to approximately 190-200 for nine team members each working 40-50 hours per week in October of 2008, representing a 40-70% increase. This is primarily an improved focus factor. If standard reference stories were used to calculate story points a much bigger improvement in velocity could be demonstrated.

On the chief Product Owner's request, the Scrum team does not hold sizes of the same stories constant over time but reduces the size as the team has gotten faster at doing certain stories or has automated their execution. Some stories go from a size of 1 story point to 0. Others go from 5 to 1, and so on. The new definition of done is also generally not accounted for in the story sizing.

If one assumes that reducing the sizes of the same stories over time due to process improvements has had an impact of at least 20%, and the definition of done has increased the actual size of stories by at least 10%, one can assume a productivity improvement of approximately 80-100% from late January to October of 2008.

This does not account for the total business value generated by the Scrum teams. The Chief Product Owner feels this has gone up significantly due to the elimination of low priority stories and focus on high priority ones. Scott

Maxwell says output in terms of value has gone up by “150%, bare minimum”. The Scrum teams believe they are doing much higher value work as well, and morale is higher.

### *I. Benefits*

Now that the team is utilizing Scrum to surface and remove impediments, there is more transparency within the Labs team and between the Labs team and the portfolio company point people / product owners. Every team member now knows exactly what they need to do and why at the beginning of every week, allowing everyone to focus on execution. Team members are now getting more done working less hours. Late nights and weekend work are now frowned upon. Sustainable pace is the goal.

The Labs team has gone from working with six portfolio companies to working with ten without getting overwhelmed or drop-off in quality. Improving OpenView’s value-add capabilities is now receiving significant attention, and there are now ten practice development backlogs.

New team members are integrated into the team and workflow extremely quickly via Scrum. They carry a full workload within three weeks of starting without extensive training. Team collaboration and cross-training increase.

Labs Scrum team is executing more stories across more portfolio companies and practice development areas while working less, producing higher quality, more value, and requiring less outside management.

### *J. Challenges*

One Labs team member cannot work within the team. He is a strong individualist and prefers answering to one clear manager rather than a team of peers. He leaves the Labs team and now works for OpenView outside the Scrum process.

While overall product quality is higher, the Labs team oscillates between focusing on velocity and focusing on quality, with spikes in velocity leading to reduced quality and focus on quality leading to reduced velocity.

While the impact of most stories is now clearer, the big picture context is still lacking for some team members. Some team members become too focused on getting ‘perfect hours’ done and driving velocity rather than on the ultimate impact of the story. While team collaboration has grown significantly, a good number of team members still work as individuals within the team.

Insufficient cross-training among the Scrum team members is still a bottleneck, especially with three brand new team members.

Long-term projects drag out over long periods of time because too many projects are being worked on simultaneously.

## IV. WHERE WE ARE IN NOVEMBER 2008

OpenView has twenty-two full-time employees, actively works with ten portfolio companies, and is working on ten practice development areas. The Labs consists of two Scrum teams, both with four people, with a velocity of approximately 180. The Labs Scrum teams have high morale and have embraced continuous improvement.

The current main initiatives include better focus and more team collaboration and cross-training:

- Focusing on higher impact stories rather than tasks.
- Focusing on FEWER goals and long-term projects at a given time and working to complete them more quickly.
- Weekly lunch-and-learn sessions allow senior team members to educate the newer ones on topics chosen by the new team members.
- Each Friday, all the portfolio company point people / product owners attend a debriefing meeting with the Labs teams, discuss the results of the sprint’s stories, and get sharper on next sprint’s goals and stories. This helps remove impediments resulting from lack of clarity between the point people and the Labs Scrum teams.

OpenView implemented the Scrum tool VersionOne ([www.versionone.com](http://www.versionone.com)) to meet the need for a more sophisticated backlog/sprint management vehicle instead of using Central Desktop online spreadsheets. The firm still uses Central Desktop for online content management, collaboration, and for organizing meetings.

## V. KEY LESSONS LEARNED

Part of successfully implementing Scrum is realizing that there are four key components that need focus.

- Direction (the backlog)
- Speed
- Quality
- Sustainability/Predictability

Major challenges emerged when the team focused on just a single area.

For Scrum to work, you must have trust, openness to conflict, commitment, accountability, and attention to results on a team [4].

Scrum is very good at revealing areas in which a team needs to improve. The self-organized team nature of Scrum immediately surfaces negative outcomes from lack of trust, fear of conflict, lack of commitment and accountability, and inattention to results. The Scrum Master must identify and clarify these impediments and then work with the team and Management to remove them.

Aggressive removal of impediments without doing root cause analysis leads to extra work. Impediments come back in the same or modified form until root cause is eliminated. Study of the Shewhart/Deming cycle and implementation of the Toyota A3 Process will be used to solve this problem [5].

Scrum is not for everyone. Some very capable people are so individualistic that Scrum kills their productivity and causes them to hurt the productivity of the Scrum team. If they cannot change, it is best to remove them from the Scrum team.

## VI. CONCLUSIONS

Repeatedly and systematically removing impediments leads to refactoring of the organization. This paper has reviewed an initial and transformed implementation of Scrum at OpenView Venture Partners. The Phase II implementation of Scrum has recently been radically

transformed into a Phase III Scrum organization that will be the topic of future study.

OpenView has gone through a classical Scrum pattern. The Phase I implementation surfaced problems with the definition of DONE. This was fixed by the Phase II implementation and velocity doubled. Increased velocity puts pressure on the Product Owner to create more and better stories. A key feature of the new Phase III organization is creation of a Product Owner team with more rigorous attention to story formation, prioritization, and READY state of user stories before allowing them into a sprint. There are already indications that the team is on their way to the second doubling of productivity using this strategy. This Scrum pattern of repeated doubling of productivity has been carefully researched at Systematic Software Engineering [6] for software projects. Here we see the same phenomenon in non-software Scrum.

It is possible to utilize Scrum for non-software project management and execution and to achieve significant productivity gains. This approach has gained traction with OpenView portfolio companies and at least three have begun to implement variations of Scrum to run multiple functions outside of Software Development. One is emulating OpenView Venture Partners by implementing Scrum in every area of the business.

#### VII. ABOUT OPENVIEW LABS

OpenView Labs' mission is to gather, create, store, and disseminate best practices and expertise for the benefit of

OpenView Venture Partners, its prospects, portfolio companies, and its community and provide high impact execution assistance to the portfolio companies. They execute their mission through a combination of best practice documents, online and in-person forums, and on-site consulting engagements using both full time operational staff and a growing network of functional experts. Their goal is to achieve significantly higher investment returns using industry best practices.

#### VIII. REFERENCES

- [1] J. Sutherland and K. Schwaber, *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Method*. Boston: Scrum, Inc., 2007.
- [2] M. Cohn, *Agile Estimation and Planning*: Addison-Wesley, 2005.
- [3] J. O. Coplien, "Borland Software Craftsmanship: A New Look at Process, Quality and Productivity," in *5th Annual Borland International Conference*, Orlando, FL, 1994.
- [4] P. M. Lencioni, *The Five Dysfunctions of a Team: A Leadership Fable*: Jossey-Bass, 2002.
- [5] D. K. Sobek and A. Smalley, *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*: CRC Press, 2008.
- [6] 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.