Hitting the Wall: What to Do When High Performing Scrum Teams Overwhelm Operations and Infrastructure

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

Robert Frohman
Pegasystems
Cambridge, MA USA
rob.frohman@gmail.com

Abstract

All-at-once Scrum implementations require total commitment to change, high level management support and aggressive removal of impediments. Several company-wide implementations are known to the authors but none of them had to deal with the complexity of a large, mission-critical, enterprise software product. Pegasystems develops software to manage, automate and optimize a broad array of business processes. In July of 2009 the company deployed over 20 Scrum teams in the U.S. and India within two months.

Complexity of languages, frameworks, and tools led to an architecture where continuous integration support for software development teams was impossible without a major upgrade in infrastructure and operations. A virtual Scrum team was formed to avoid "hitting the wall" before this impediment impacted the first Scrum release of the software. Here we describe how a Scrum team engineered a continuous integration environment for hundreds of software developers on two continents within a few weeks.

1. Introduction

All-at-once Scrum implementations known to the authors are Wildcard (27 Scrum teams the first day leading to total domination of their market in six months), OpenView Venture Partners [1] (companywide Scrum led to reorganization every four months for rapid response to emerging requirements), expansion stage startups Balihoo and Intronis (senior management and departmental team Scrums). PatientKeeper had company wide hyperproductive Scrum teams for many years [2]. None of these implementations had to deal with the architectural complexity or rapid deployment of a large, mission-critical, enterprise software product combined with dozens of teams in the United Sates and India.

Pegasystems develops software to manage, automate and optimize a broad array of business processes. The BPM and SOA engine markets, at $1.8 billion in 2008, are expected to grow to $6.2 billion by 2015 [3]. As a growing company, Pegasystems is well positioned to acquire market share. However, in this highly competitive market, it faces challenges from traditional stack vendors like Oracle and IBM. As well, there is the potential for disruption from smaller vendors coming up the stack. In addition, Pegasystems is challenged to acquire top engineering talent in the MIT area and must compete with neighboring Google, Yahoo, and Microsoft R&D labs. Innovation is critical to success in order to capitalize on and extend the competitive gains already seen. An innovative environment is essential to grow and retain talented staff.

Pegasystems made a company-wide commitment to Scrum in July of 2009 after previous experience with a small pilot project. They consulted with one of the authors on a strategy for rapid rollout of Scrum in July 2009.

2. Introducing Scrum

In August 2009 there were almost a 100 engineers in the U.S. and 100 in India using traditional project management. The replacement of that process with Scrum at Pegasystems occurred quickly and in several steps:

1. A half day was spent with the management team to go over the basics of Scrum, expected productivity gains, requirement for management involvement in removing impediments, and steps to organizational transformation from waterfall to a Scrum development environment.

2. Product backlog needs to be ready before starting Scrum with the engineering teams. Company objectives were optimized for the next release to a

few significant goals. The Product Marketing team was trained as [redacted] Product Owners along with much of the management and senior members of the engineering teams. Workshops were facilitated to help transform waterfall requirements to a prioritized Product Backlog for about six teams.

3. An enterprise transition team was formed at the senior management level to surface and prioritize company impediments and develop implementation plans for removal. This team needs a senior management sponsor, an Agile company coach, a manager who can work on agility issues with the management team, a process leader who will drive process change across the company and HR support for scheduling, consulting, and training.

4. The Scrum Foundation provided a consulting team of six senior Agile experts for [redacted] Product Owner Training and [redacted] Scrum Master training for development, management, and product marketing. This team then did onsite coaching in the U.S. and India to facilitate startup of teams. In critical cases, the Agile coaches assumed the role of Scrum Master for a team and developed and trained an internal Scrum Master.

5. The teams in the companies were reorganized to maximize impact on delivering the product backlog for the next release as soon as the Product Owners clarified the backlog and began to get it in a "ready" state.

6. The Scrum teams began work at the beginning of August 2009 with a focus on coaching half a dozen teams to successfully complete sprints with working code tested at the feature level before the end of sprints.

The enthusiasm and focus of the Pegasystems organization caused them to more aggressively implement Scrum than anticipated and within two months they had deployed over 20 Scrum teams in the U.S. and India. Complexity of software languages, frameworks, and tools created an architecture where continuous integration support for software development teams was impossible without a major upgrade in infrastructure and operations. The acceleration of software output by the new Scrum teams caused continuous integration to be identified as the highest priority in the company.

A virtual Scrum team was formed with leading experts in the company to avoid "hitting the wall" before this impediment impacted the first Scrum release of the software. This paper describes how a Scrum SWAT team engineered a continuous integration environment for hundreds of software developers on two continents within a few weeks.

3. Hitting the wall

In spring of 2009, with several successful releases and increasing complexity in the product, test cycles began to lengthen. Engineering staff had grown to over 175 employees working on many concurrent projects. Aggressive hiring plans continued to accelerate the increase in staff. Despite the growth in product development, there had been little investment put into the infrastructure and operations as required foundational support. Pegasystems took the usual approach of patching with more resources when issues arose. By May of 2009, the development environment consisted of several application servers accessing a common database, concurrently developing both the Java and Rule aspects of the product. Complicating matters further, a new engineering office in Hyderabad had opened and was a focal point of engineering growth. It was also leveraging the hardware configurations in the Cambridge, MA office.

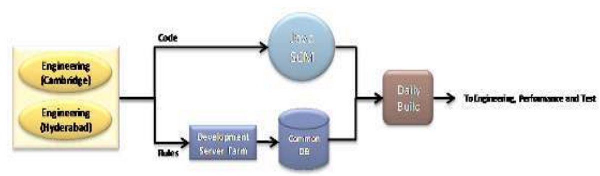

Figure 1: Development infrastructure before July 2009

In July of 2009, the company began implementing Scrum and by September teams were fully deployed in the U.S. and India. The adoption of Scrum resulted in teams being formed around functional areas of the product, effectively breaking up the product development effort into functional areas. As a result, there was a need for additional hardware and support resources as well as for more sophisticated build processing to address integration issues. By September 2009, velocity of production had increased by 2x, quickly overwhelming product development operations infrastructure and release engineering.

The enterprise transition team developed a prioritized backlog of institutional impediments and put continuous integration [4] as the top priority. It was clear that being stopped by plumbing could result in losing the war for the hearts and minds of the people. Achieving one of the key objectives of Scrum, hyperproductive teams, was impossible without a continuous integration environment and automated tests run on every build.

4. Root cause analysis

The desire to be successful with Scrum led to an organizational focus on process, training of teams, and migration from existing methodologies. In addition, there was excitement and energy about getting started which left few resources to focus on supporting these efforts. With the bulk of Engineering forming into Scrum teams and becoming highly productive, the Infrastructure and Operations Teams remained operating as usual but continued to fall further behind as there was not enough time to focus on improving process efficiency. A significant portion of activity was spent on short-term low-gain fire-fighting which only reduced the resources required to maintain and significantly improve the infrastructure.

The Infrastructure and Operations Teams had developed a strategy to solve the problem of regular global builds of all software for integration testing, but did not have the resources or the institutional commitment and focus to execute on the plan. It was only through the Scrum process of identifying and prioritizing impediments that the company identified continuous integration as the top priority impediment in the correct timing of their Scrum rollout. Even then, there was no way to act on this impediment without senior management action to organize resources across the company.

The engineering leadership and Scrum consulting team met with the CEO of Pegasystems and presented the problem. Only with a decision from the top could aggressive action be taken.

5. Taking action

Embracing Scrum without addressing the infrastructure and operations caused a serious situation. Teams were becoming high-performing and the infrastructure was unable to support the demand. Anticipating the impact, Product Operations teams quickly came up to speed on the methodology and included some of the first [redacted] Scrum Masters in the organization. Over the summer some planning was done within Product Operations, (Figure 2 – Aspirational Infrastructure), but with the primary focus on getting Scrum teams operating, these plans were not implemented and operational issues were addressed ad hoc.

In October 2009, a consultant was brought in to address the shortfall and to create a long-term strategy to proactively address the need. In parallel, a Scrum team was formed for Product Operations with the SVP of Engineering as Product Owner.

The Product Operations team had several responsibilities including the patch process for delivering fixes for products already in the field. The focus of the first sprint was to address two questions:

1. What will keep the trains running?
2. What can a small Scrum team do to prepare infrastructure and operations for hyper-productive software delivery?

6. Goal/target condition

The ideal target would be to extend the infrastructure and operations to support highly productive Scrum teams with minimal impact on those Scrum teams and without negatively impacting the other Product Operations responsibilities. In the end, the desired state included the following, with the expectation that normal product operations day-to-day activities would not be significantly impacted.

- Upgraded build process that supports continuous integration
- Hardware/software infrastructure that supports Agile development
    - Standard image that every team develops on – can be deconstructed and renewed in less than 24 hours
    - Image must evolve as product grows
    - Keep current operations running until new environment available
    - Consistently measure progress with quantitative metrics
- Apply continuous improvement process in infrastructure and operations in order to grow and capture talent of best engineers

7. Countermeasures (experiments)

As September and October unfolded, it was apparent that the engineering teams had outpaced the existing infrastructure even as the Product Operations teams had attempted to implement improvements. The renewed focus on Product Operations at the end of October allowed the team to work on long-term solutions as opposed to applying a steady stream of quick-fixes. Further enabling this activity, the Product Operations teams were divided into two independent teams. One team was established to "keep the trains running". It was essential that day to day operations continue while the infrastructure was improved. The second team, with the SVP of Engineering as Product Owner, was to focus on improving the infrastructure and operations over 3 two-week sprints in November and December 2009. The approach taken included,

1. Communicate with Engineering about the changes
   a. SVP Engineering addresses Engineering about expectations
   b. Establish a Scrum Masters meeting where infrastructure and operations information can be conveyed.
2. Developed, document and disseminate processes
   a. Source Control
   b. Build
   c. Performance and Test
3. Develop tools and reporting

The first task was to communicate with engineering about the expectations and this was accomplished in two ways. The Product Owner (SVP of Engineering) communicated directly to the engineering leadership that Product Operations changes were to be made and that there may be a short-term negative impact to productivity. Communications were bilateral, enabling leadership to communicate directly with the Product Owner.

In parallel with the leadership communications, the Scrum Masters formed a Scrum Master meeting which was to take place for 1 hour per week. In this meeting, changes to infrastructure and operations were disseminated including tutorials on the appropriate use of new technology introduced. In addition, expectations were set that this meeting would also provide a forum for integration issues to be discussed across Scrum teams.

With communications opening up across the organization, the next step for the Product Operations team was to identify the current state. The key areas of focus were the development environment and the build process. With the processes mapped, impediments and shortcomings could be identified, prioritized and rectified as backlog items.

A key impediment was the distributed development model in use by the Scrum teams. Originally, before Scrum adoption, all teams worked together in a common environment. There were many integration issues as a result and no solid process to rectify them. So with the initial adoption of Scrum in summer 2009 it was decided that each individual Scrum team should have a dedicated and private server with integration to take place after the build. In retrospect in the fall of 2009, the downside of the decision to break up the environment was being felt as integration issues were being discovered very late, as the build process was the first point of integration. To overcome this impediment the decision was made to re-integrate all teams onto a single server environment again, so that integration issues could be rectified before they left development. Further, the original environment was inconsistent and an effort was made to standardize the development server images and be ready for additional hardware as the organization scaled. Taking into consideration the desired Continuous Integration model, the aspirational development environment was identified to include such features as collocation, scalability, and promotion models.

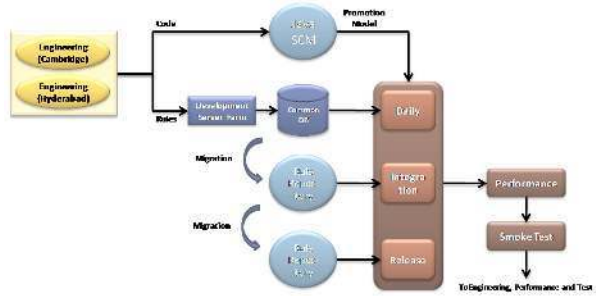

Figure 2: Aspirational development infrastructure

Another impediment was identifying what would constitute a good build. Release Engineering saw it as successfully producing an image whereas the Performance team saw it as passing performance tests. Further, Product Management might qualify it by the availability of key features. As a result, when claims were made that a build was broken, much activity was spent trying to identify the root cause.

In the first sprint the Product Owner identified a build process map as a high priority backlog item. The resulting map described the build process as starting with the source control, moving through Release Engineering, Performance, basic Test Engineering smoke testing. In addition, a web portal was established which demonstrated the build as it made it past the various acceptance gates. With a well defined build process and acceptance criteria established, the conversation moved from "the build is broken!" to "the build was not accepted because". This had dramatic effect on reducing the churn associated with a broken build and resulted in a faster response to issues when they were raised.

8. Confirmation (results)

Teams began to achieve proficiency with Scrum and the infrastructure and operations was able to support it. The Shared Development Environment was very stable with few outages or issues encountered. Also, the build environment was demonstrating versatility and capacity to scale. In November, Scrum teams had targeted feature complete for February 1, 2010. With the Scrum teams functioning and the infrastructure and operations capable of supporting them, this target was met.

Achieving this result, the emergency Scrum team was disbanded and Product Operations teams reverted to their original charter.

9. Follow-up (actions)

In February 2010, with the bulk of the original issues resolved and Scrum teams no longer being impeded by the operational issues, the Release Engineering team began to evolve the build process to one of a continuous build. This was an early step in moving the organization to a Continuous Integration model.

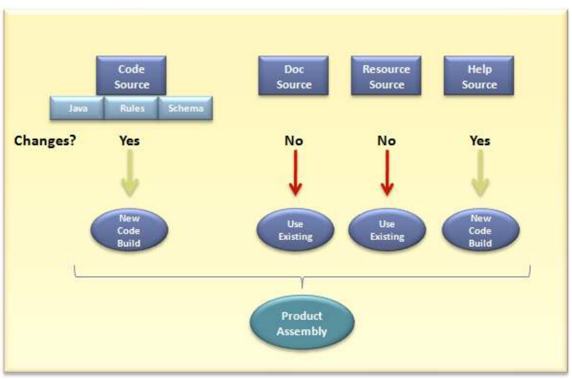

Figure 3: Continuous build process

In keeping with the desire for continuous process improvement, the next step in this build model was to break up the Java, Rules, and Schema aspects of the Code Source into their own build environments, picking up only those elements that changed when assembling the product.

With the operations issues largely resolved the next key impediment was identified as stemming from how bugs are addressed. Without a continuous integration model and lacking test-driven development, Technical Debt began to accrue on Scrum Team backlogs. While the feature freeze date of February 1, 2010 was met the amount of technical debt accrued required a bug-fix phase that lasted through February and March. The next challenge to the Engineering organization was to address the proliferation of defects caused by lack of continuous integration in the early phases of Scrum deployment.

10. Conclusions

It is very important to take into consideration the underlying operations and infrastructure when embracing a methodology like Scrum. Teams will quickly see an increase in productivity and planning for success is a critical element. As a result it is important to invest in the infrastructure.

1. Form a Scrum team to focus on infrastructure and operations. Communicate team expectations to the organization as a whole.
2. Evaluate hardware needs carefully.
3. Document existing infrastructure and operational processes. Evaluate for impediments and develop a backlog.
4. Implement a simple continuous build process.
5. Identify needs of infrastructure operations in support of a Continuous Integration vision.

Before embarking on Scrum in development, establish and train a Scrum team to focus on the infrastructure. This team should adopt a Continuous Process Improvement approach to increasing capacity within the organization. As infrastructure and operations typically impact everyone, set the expectation that change will be coming. Be clear about the purpose and benefits, and identify the communications plan.

10.1 Differentiate resources required

With increased productivity it is likely that additional hardware will be required. Be cautious in applying new hardware and understand that it must be supported and maintained. As Brook’s law states that adding people to an already late project will only make it later, adding hardware to an overwhelmed infrastructure team will only decrease its capacity as it assimilates the support of that new hardware. So be careful to distinguish “manpower” and “CPU” needs.

10.2 Identify current state

Identify and map the existing infrastructure and procedures and begin to evaluate where there could be improvement. Keep in mind the overall vision of Continuous Integration and build upon what already exists. Consideration should be put to all aspects of code development, from source management through deployment to an integration server and to the customer.

10.3 Start building

Evaluate the current build processes and look for opportunities to evolve to a continuous build model. Ideally, each component is built only when it changes. In this way you can 'assemble' the various builds and only be concerned with variability in the component what really changed. Ideally, the Assembly is a customer-ready image.

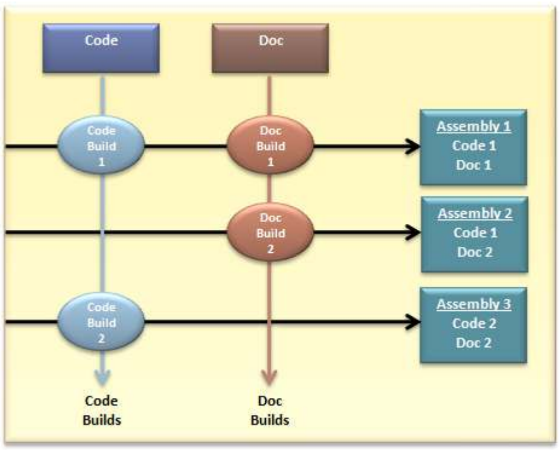

Figure 4: Assembly by continuous build

11. Future research and implementation

Continuous integration provides a platform for carefully prioritizing automated tests that systematically remove deployment problems and accelerate product release. While the rapid deployment of Scrum at Pegasystems allowed achieving the goal of feature complete at the desired date for a major release, the lack of continuous integration and automated testing with an aggressive focus on defect prevention and removal in the early stages resulted in several months of bug fixing required before deployment of the first Scrum release of the product.

Focus on shorting the time to fix defects found in an automated build process has cut the cost of projects in half and reduced defects by 40% companywide at Systematic, a CMMI Maturity Level 5 company [5]. More recently at an Openview Venture Partners company [1] wrote 130 tests in a few weeks after implementing continuous integration and cut deployment time after feature complete of high-reliability network storage server software from 4-6 weeks down to 2 weeks. With a similar focus on low defect tolerance, Pegasystems could cut the final hardening phase of major product releases from a few months down to a few weeks. It was possible at some Scrum companies [6] to cut this time period to zero with aggressive removal of all impediments to deployment.

12. References

[1] J. Sutherland and I. Altman, "Take No Prisoners: How a Venture Capital Group Does Scrum," in *Agile 2009*, Chicago, 2009.

[2] J. Sutherland and K. Schwaber, *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Method*. Boston: Scrum, Inc., 2007.

[3] WinterGreen-Research, "Worldwide Business Process Managerment (BPM) Market Opportunities, Strategies, and Forecasts, 2009 to 2015.," Research and Markets, 2009.

[4] P. M. Duvall, *Continuous Integration: Improving Software Quality and Reducing Risk*: Addison Wesley, 2007.

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

[6] J. Sutherland, "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects," in *AGILE 2005 Conference* Denver, CO: IEEE, 2005.