Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership

Ken H. Judy
Oxygen Media
kjudy@computer.org
CSP, CSDP

Ilio Krumins-Beens
Oxygen Media
ikruminsbeens@yahoo.com
CSM, PMP

Abstract

Scrum describes a separation of roles; the product owner is accountable for achieving business objectives and the team for technical execution. A pragmatic and collegial relationship between a product owner and team can satisfy the definition of collaboration and honor roles while barely tapping or actually working against the potential of a project and its participants. This paper surveys literature to describe different forms of collaboration, to establish that deep, unbounded collaboration is at the heart of agile values, and that partnerships of high trust and shared risk lead to value and innovation. Finally, this paper incorporates a real-world example of a product owner who, while remaining accountable to the outcome, shared ownership over vision, priorities and execution with her Scrum/XP development team.

1. Roles in Scrum

Table 1. Three roles in Scrum [1]

Role	Definition
Product Owner	The person responsible for managing the Product Backlog so as to maximize the value of the project. The Product Owner represents all stake holders in the project.
Team	A cross-functional group of people that is responsible for managing itself to develop software every sprint.
Scrum Master	The person responsible for the Scrum process, its correct implementation, and the maximization of its benefits.

"[T]he product owner is responsible for an emerging set of requirements on something called the product backlog. They are responsible for elucidating these requirements as needed, decomposing these requirements on an ongoing basis, changing them to optimize return on investment, and even meeting with development teams frequently to tell them what is needed and to review what they have done. The product owner has gone from someone who could blame development if a project failed to someone who is responsible for the success or failure of the project. At Yahoo, this person is called the "single wringable neck." [2]

The need for this role arises from what was described in 1969 as the "software crisis" [3], an unacceptable failure rate in software development projects. In R. Charette's 2005 IEEE Spectrum article, "Why Software Fails", eight of the twelve most common factors for project failure lie the relationship of the businesses to development: unrealistic or unarticulated project goals, poor reporting of the project's status, unmanaged risks, poor communication among customers, developers, and users, poor project management, stakeholder politics, and commercial pressures. [4]

Agile attempts to resolve the "software crisis" by, "satisfy(ing) the customer through early and continuous delivery of valuable software." It addresses the challenge of aligning development to business value by insisting "business people and developers must work together daily throughout the project." [5] Scrum elaborates on this by insisting the business be represented by one accountable, empowered individual.

"The Product Owner is one person, not a committee… The practice Scrum adds that only one person is responsible for maintaining and sustaining the content and priority of a single Product Backlog. Otherwise, multiple conflicting lists flourish and the Sprint teams don't know which list to listen to. Without a single Product Owner, foundering, spin, contention, and frustration result." [6]

One interpretation of this call for a single Product Owner is that the development team itself should play little part in shaping the vision and value priorities of a product backlog, focusing instead on efficient delivery of those priorities. This is a suitable defensive posture for teams new to agile and in environments with barriers of trust and communication between the business and development. It is fighting a battle to take a poorly

1530-1605/08 $25.00 © 2008 IEEE

functioning organization to a level of efficient performance.

The authors intend to show that in organizations with an established scrum practice and the right conditions for meaningful cross-departmental collaboration, a capable Scrum Product Owner and performing team can build a spirit of collective ownership over all aspects of a product lifecycle while still fulfilling the Scrum roles. This collective product ownership is close to the lean product development origins of agile which values the insights into products held by those who build them and positions an organization for sustained excellence and product innovation.

2. Collaboration beyond defined roles

Scrum itself is derived from research into other disciplines. [7] In that spirit, it is worth looking for insights into collaboration within Scrum beyond research specific to software development. Over a seven year span, Andy Hargreaves and Michael Fullan authored the *What's Worth Fighting For* series of monographs to help teachers and principals fight for fundamentally positive changes in education. Hargreaves and Fullan provide concise and coherent descriptions of pitfalls that prevent collaboration from driving organizational change. Since both collaboration and continuous improvement are at the heart of Scrum, these pitfalls should be of particular interest to Scrum practitioners.

The disempowering climate faced by teachers is a direct parallel to that faced by software developers disengaged from the business vision and value priorities of their work. "… [W]e have collectively 45 years of teaching experience and nobody has ever asked us our opinion about anything where it would actually be put into action. And yet I've got to have more experience with junior [children] than a lot of the people who are telling me what I should be doing with them. And I think that is very frustrating… I think I could help bring a lot to it and nobody ever asks, no one ever asks what we think. They just go ahead and proclaim and we have to follow." [8]

The two disciplines have more in common. Education is focused on directed learning shaped by a larger ecosystem of stakeholders. Teaching is both a profession and a creative act dependent upon human interactions. Software development is invention and directed problem solving with a similar ecosystem of stakeholders bound by a set of ethical concerns, requiring creativity and dependent on human interaction. Both teachers and software developers face criticisms for a high failure rate.

Problems faced by teachers that limit the potential of schools and therefore student outcomes are: overload, isolation, "groupthink", untapped competence, narrowness of the teacher's role, poor solutions, and failed reform. [9]

Agile values of sustainable pace, self-organization and practices such as retrospectives, pairing, and the daily standup, among others are meant to address concerns of overload, "groupthink" and individual isolation. The Scrum Master role attempts to protect the team from management solutions and reforms that however well-intentioned subvert the productivity and cohesion of the team.

However, it is in the relationship to the Product Owner that we need to look for concerns of untapped competence and narrowness of the software developer's role.

2.1. Expand the team's role to promote organizational efficiency

"(F)or classrooms to be effective, schools must be effective. Teachers are a big part of the school. As individuals and groups of individuals, they must therefore take responsibility for improving the whole school, or it will not improve." [10]

By analogy - for development projects to be effective, the businesses or institutions that surround them must be effective. As individuals and teams, developers must take responsibility for improving the whole organization or it will not improve.

"Scrum is the very simple mechanism that helps an organization be more effective in accomplishing its goals.... seeing what is going on in their organization and going through the change process to become effective [a]nd learning how to continually inspect and adapt to keep their organization's practices optimal." [11]

2.2. Expand the team's role to promote innovation

Japanese lean manufacturing principles are the acknowledged origins of lean software development and Scrum and a significant contributor to the larger body of agile practices and values [12].

Consumer product companies with sustained innovation programs like Sony, Toyota and Canon engage the imaginations, tacit understanding and problem solving capabilities of front-line staff, middle-management, and leadership in a way that flows knowledge up, down and across the organization. As described by Nonaka, Takeuchi and others, this lean manufacturing approach bases success upon responsiveness, customer focus, high quality, and repeated innovation.

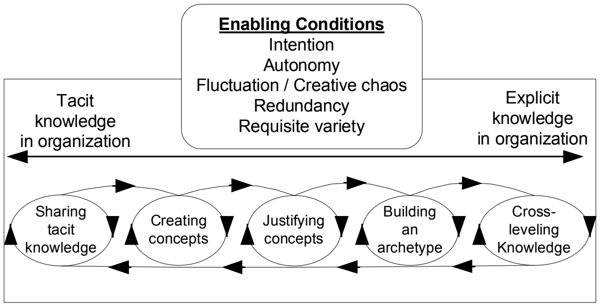


Figure 1. Five-phase model for the organizational knowledge creation process [13]

“Collaboration should mean creating the vision together, not complying with the principal’s own… The articulation of different voices may create initial conflict but this should be confronted and worked through. It is part of the collaborative process.” [14]

Developers need to engage with Product Owners and executives in a running dialog on “why” and “what” not just “how” and “when”. In an industry where value derives from modeling solutions, the knowledge worker is a core asset not just for their labor but for their understanding of the core opportunities software is meant to address.

3. Collaboration in agile practice

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

i. Individuals and interactions over processes and tools
ii. Working software over comprehensive documentation
iii. Customer collaboration over contract negotiation
iv. Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.” [15]

The *Agile Manifesto* and the principles behind it, emphasize collaboration among team members and project sponsors. Jean Tabaka suggests the original signers of the *Agile Manifesto* “explicitly declared collaboration and communication as fundamental practices for successful software development, particularly in the first and third declarations” [16].

“In collaboration cultures, the success of the organization hinges on how teams formulate, organize, decide, and deliver. In such organizations, teamwork is prized, with an emphasis on how individuals share information, process it, and converge on the best thinking. Decisions are either team-driven or manager-driven with team consultation. In addition, consensus plays an important role in creating sustainable agreements about the solutions that emerge through the great wisdom of the group.” [17]

Also, three principles behind the *Agile Manifesto*, speak directly to collaboration:

“Business people and developers must work together daily throughout the project.”

“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.” [18]

Scrum and XP emphasize collaboration in their values. Scrum values include commitment, focus, openness, respect and courage [19]. XP values communication, simplicity, feedback, courage, and respect. [20] Kent Beck’s 2001 “One Team” paper called for open collaboration between customer and team. The daily standup and XP’s onsite customer are specific collaboration practices [21].

Jim Highsmith states, “Balancing at the edge of chaos between flexibility and stability requires people who are good improvisers—who have the ability to deal effectively with the ambiguity, and the paradox, of pursuing to seemingly dissimilar goals at once. Organizations that support these improvisers have three key traits:

- An adaptive culture that embraces change.
- Minimal rules that encourage self-organization, combined with the self-discipline to closely adhere to those rules
- Intense collaboration and interaction among the project community” [22]

4. What passes for collaboration

Collaborate:

1: to work jointly with others or together especially in an intellectual endeavor…
3: to cooperate with an agency or instrumentality with which one is not immediately connected [23]

All attempts to work together are not equal. J.W. Little identifies four different kinds of collegial relations. Of them, only one is strong enough to contribute to a collaborative culture of enduring benefit. [24]

Table 2. Forms of collegial relations [25]

Benefit	Form	Activities
Weak	Scanning and storytelling	Anecdotes without connecting to each other experience.
	Help and assistance	Help giving only when asked.
	Sharing	Pooling of existing ideas without examining and extending them.
Strong	Joint Work	Teaming, planning, observation, action research, sustained peer coaching, mentoring, etc.

In fact, Hargreaves and Fullan identify three patterns of collaboration that entrench status quo rather than move an organization forward.

4.1. Balkanization

Small collaborative groups can exist in isolation or in competition with other groups within an organization. When small teams are allowed to be autonomous in intentional but directed competition this can lead to creative problem solving such as in Nonaka and Takeuchi's concept of "bounded cohabitation" where teams are set in productive competition with each team pursuing a different set of premises and value propositions all geared toward the same outcome. [26].

However, if this fragmentation divides on disciplines, is entrenched or exists in an environment of low trust, balkanization can stifle organizational improvement. Individuals do not work jointly across groups. Valuable learning in one group is not communicated or is ignored or undermined by others.

Agile is often introduced bottom up with little executive sponsorship in less than optimal organizational cultures. In this context, development teams exist in isolation to the larger business unable to share their learning and separated from decisions about opportunities, product portfolios, potential revenues, and product features. This is both a fragile place for the teams and also diminishes opportunity for the company.

"Effective collaborations operate in a world of ideas, examining existing practices critically, seeking better alternatives and working hard together at bringing about improvements and assessing their worth" [27]

4.2. Contrived collegiality

"The unpredictable nature of collaborative cultures can lead administrators towards forms of collegiality which they can control, regulate, and tame." [28]

14% of respondents to a recent survey on agile adoption listed loss of management control as one of their organizations greatest concerns in adopting agile practices. 13% cited loss of predictability. 20% cited lack of up-front planning. [29]

In an attempt to exercise control or fix perceived weaknesses in agile practice, management or customers may impose communication overhead that, in fact, only produces a surface veneer of visibility and collaboration while stultifying relationships and eroding goodwill with the team.

Granted, there are times when leaders need to introduce collaborative practices into an environment. But the harmful effects of contrived collegiality present a real danger to top-down efforts to impose agile adoption. If team members or stakeholders don't buy into the values underlying agile practice, then a project may run as a "scrum" while not achieving meaningful change or returned value.

4.3. Bounded collaboration

Collaboration can be limited in context and substance. For example, a product owner and team may never work together except in the context of conversations during formal inspection points and when the team has specific questions. Topics may be limited to the immediate needs of the project and not range to larger questions and concerns.

"Bounded collaboration rarely reaches deep down to the grounds, the principles or the ethics of practice. It can get stuck with the more comfortable business of advice giving, trick trading and material sharing of a more immediate, specific and technical nature. Such collaboration does not extend beyond particular units of work or subjects of study to the wider purpose and value of what is taught and how. It is collaboration, which focuses on the immediate, the short-term and the practical to the exclusion of longer term planning concern." [30]

In bounded collaboration developers do not invest themselves in the business outcome of a project. This can occur through no fault of their own if the plan is vague, not shared with them, or if their input is not invited or listened to. Some managers simply don't consider it important that technical staff buy into the vision or features of the software they're building.

One might characterize the resulting attitude as, "I'm not personally invested in the business plan or priorities but I will execute on them *as you (product owner) define it*. Since I have no authority or meaningful influence on the plan or priorities, as long as I produce reasonably error-free code I refuse to be judged by how the product fairs in the marketplace."

In such an environment, the success of a product beyond the reliability, supportability, scalability and time to market of its implementation rests entirely on decisions of the Product Owner. It's fair to claim a visionary Product Owner with great insight into her customers can find success supported by capable though uninspired craftsmanship from the team. However, this isn't a model for sustained organization knowledge creation.

5. What is successful collaboration?

"**Collaboration** is a mutually beneficial and well-defined relationship entered into by two or more organizations to achieve common goals. The relationship includes a commitment to mutual relationships and goals; a jointly developed structure and shared responsibility; mutual authority and accountability for success; and sharing of resources and rewards." [31]

6. Factors influencing collaborations

The Wilder Center published a review literature on factors influencing successful collaborations: [32]

6.1. Environmental factors

- History of collaboration or cooperation in the community
- Collaborative group seen as a legitimate leader in the community
- Favorable political and social climate

6.2. Membership characteristics

- Mutual respect, understanding and trust
- Appropriate cross section of members
- Members see collaboration as in their self-interest
- Ability to compromise

6.3. Process and structure

- Members share a stake in both process and outcome
- Multiple layers of participation
- Flexibility
- Development of clear roles, rights and responsibilities
- Adaptability
- Appropriate pace of development

6.4. Communication

- Open and frequent communication
- Establish informal relationships and communication links

6.5. Purpose

- Concrete, attainable goals and objectives
- Shared vision
- Unique purpose

6.6. Resources

- Sufficient funds, staff, materials and time
- Skilled leadership

7. Collaboration at Oxygen Media

Oxygen Media is the only cable network owned and operated by women. The company launched on February 2, 2000 as a converged television and internet company. As the internet bubble burst, the company focused on its cable network where it achieved profitability in 2005 and has outperformed its business plan for several years running. Oxygen is carried in over 70 million homes.

Oxygen's CEO, Geraldine (Gerry) Laybourne, consistently works to invest her employees in a shared sense of purpose and a collaborative style of work:

"It would have been the easiest thing in the world for me to sit down and write a mission statement. But that isn't what I did. I put the general counsel in charge of drawing up the mission statement. He's the fairest person I know – and he has patience. He involved everyone in the process. Everyone here was able to make a contribution, everyone was listened to and everyone became part of the process." [33]

8. Unbounded collaboration on the Ript™ project

In January 2006, Oxygen began a process of cross-departmental brainstorming that eventually led to a new mission to build playful and purposeful consumer software based on the observed needs and interests of women.

In June 2006, Oxygen began its first official Sprint for the first of these products, a tool for visual planning and sharing called Ript™.

Gerry Laybourne is the Product Owner with assistance from a proxy within the development team. From the outset, this project has been an unbounded collaboration on all aspects of the product between the Product Owner and the development team. The team contributed to the brainstorming of the product, the vision, roadmap, features and priorities.

The Product Owner has allowed the team to fall in love with the project by leading while listening. She's given the team a gift of high expectations demanding that the product be original, useful, and fun. She's treated the team as peers despite her much higher authority. She's shared her excitement for the product while sharing credit for it. She's championed her priorities while allowing the team to question any and all aspects of the product. Both had the pleasant experience of disagreeing and realizing the other was right.

9. Collective product ownership

In collective product ownership, the product owner does not dominate the development team nor is there a strict boundary between product owner and development team. It is a collaboration built upon high-performance, mutual respect and deep trust. The product owner walks a tight rope, engaging the team in an evolving product and business plan while guiding the project toward her vision and high-level goals. The team is passionate about the product they are building and feel personally accountable to the product's success.

Peter Senge writes, "The first step in mastering the discipline of building shared visions is to give up traditional notions that visions are always announced from 'on high' or come from the organizational institutional planning processes." [34]

Of her role as Product Owner, Gerry Laybourne said, "Most of the time in a creative project, I might be involved at the beginning and the very end, but never in this kind of on-going role... I am on the team with you. I love it. I own it. I worry about it. Just like you guys do. That's why it works."

"Honestly, if I were as know nothing as I am about software development and I was a command-and-control person who thought I knew best about everything, this would be a disaster. I have seen this happen with TV executives who have never made anything. I think that is why, especially with your group, I'm interested in having everybody's necks in the noose with me." [35]

Great product owners lead their teams rather than manage or control them. Mary and Tom Poppendieck observe that 3M and Toyota inspire technical teams through *respected leaders* who "fully understand that leveraging the talents of a large pool of experts is far more effective than trying to control the work." [36]

Sustained innovation involves an embrace of seeming contradictions. [37] An environment of collective product ownership requires a highly accountable product owner. The team must trust they are not out there alone ahead of the business. The product owner must stand in front with authority and responsibility for the ultimate outcome.

In this sense, collective product ownership is not the same as collective ownership of code as described in XP. In ideal collective code ownership, any developer can refactor any area of code in an application as long as it continues to meet the contract defined by interfaces and unit tests. The idea a single accountable authority on such decisions is anathema. [38]

Gerry Laybourne delegates a great deal of authority. She values differing perspectives and makes decisions in the face of strong competing arguments. However, there is no question that she is the ultimate authority on questions of customer value and ultimately accountable to the outcome. She has overruled the team on several instances.

"[I]n terms of my management of all creative projects, it is exactly the same, I try to avoid micromanaging. On television it is the same thing. I like an idea or I hate an idea. It is a yes or no with me. It is not a bunch of manhandling or woman handling. It is not a bunch of second guessing. "No, I don't like that. I don't want that. It is not right." Then there is a conversation, but I feel like everybody's voice is heard." [39]

In collective product ownership the development team is passionate about the product they are building, connected to business goals and empathetic to their end users. They feel on the hook if the product does not succeed.

The team is a united cohesive, performing unit. They challenge each other's assumptions but rally around decisions. They share recognition for ideas and do not compete to impress. They trust that everyone carries their own weight. Ideas coming out of the group are both more novel and rigorously debated than those of any single individual.

10. Practices to support collaborative product ownership

The Product Owner has obligations to her company's much larger television business, to other interactive initiatives and to the cable industry itself. For example, she co-hosted the 2007 National Cable Television Association conference. She cannot be a highly available onsite customer.

However, the team knew success required she stay engaged in the project. The product sprang from her vision and serves her mission for Oxygen. The product and the larger program of work still needed a business plan. While the software team had built Scrum/XP practices over four years of successful outcomes with their clients, agile principles were not widely understood at a senior level.

Therefore, a member of the team, himself an experienced Scrum Master assumed the role of Product Owner Proxy. The proxy sits with the team and is highly available to answer questions about priority and desired behavior. He maintains the backlog, author's user stories and works with the team on acceptance criteria. He accepts stories as they are completed, works with testing and the team to ensure product quality, and coordinates with research, marketing and other external units of the business. He also demonstrates the product for internal and external audiences.

11. Using the Product Owner's time well

The team has set up a structure where the Product Owner is able to fulfill her role with a minimum of time commitment. In return, the Product Owner has made a point to attend every meeting in person. In the past 27 biweekly sprint reviews she has rescheduled or participated remotely only three times. This commitment is yet another way she earns the team's trust and loyalty.

Table 3. Gerry's time commitment

Frequency	Purpose	Time
2 wks	Sprint Reviews	60 min.
2 wks	Mid-sprint check in (pre-staged sprint planning)	15+ min.
3-4 mos.	Release Planning	3x30 min.
At PO's discretion	Re: business plan, potential partnerships, marketing, and consumer research	

11.1. Sprint reviews

Developers rotate the responsibility of presenting features for each sprint review. The remaining time is used for business issues, risks and to confirm priorities for the upcoming sprint.

Team members feel free to raise concerns, suggest new features or question existing priorities. If a decision is pressing for the next sprint, the Product Owner will often canvas the room before weighing in but will make sure the team can move forward.

If an issue of long-term significance remains unresolved, the Product Owner will ask the team to work through their disagreement and come back to her with a single or specific set of recommendations at the next review.

After the review, the latest build is installed on the Product Owner's computers. She is an active and avid user of the application in development.

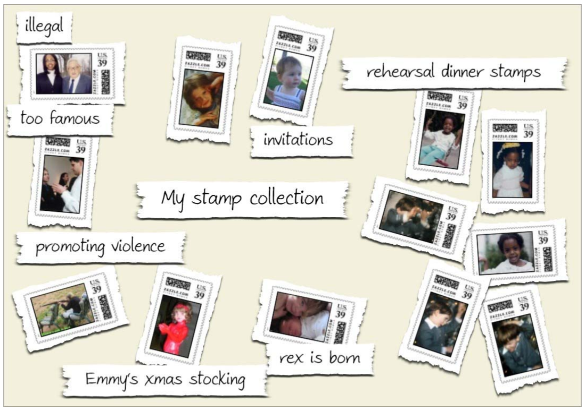


Figure 3. One of Gerry's Ript™ pages

At some point mid-sprint, the Product Owner meets with representatives of the team and dev management for an update on sprint health, issues related to the project, and to pre-stage priorities for the next sprint.

11.2. Release planning

Leading up to release planning, the Product Owner and her proxy meet for 30 minutes to discuss priorities at the theme level. The proxy provides a straw man release backlog as a context for this meeting. After that subsets of the team review themes and provide rough relative sizing.

The Product Owner and team spend 30 minutes at the beginning of each release planning meeting discussing release targets and their value to the business. The team then spends two hours on relative sizing of themes. They propose new themes and question others. The proxy either responds or flags it for later conversation. The proxy and team chunk the themes into sprints. The team votes on their confidence in the resulting plan.

The Product Owner returns for 30 minutes at the end to discuss proposed changes to release content or priorities address any open concerns and approve the team's release commitment.

12. Keeping the project on course

At times, the Product Owner and team have been unable to come to agreement on a significant decision. Generally, that has been a bad smell of non-essential complexity or a concept with an unclear value proposition. The product has generally benefited from deferring or dropping the work related to such decisions.

Based on her instincts and judgment, the Product Owner has pushed the team to action on specific issues despite disagreement in the team. Her judgment on when she chooses to do this, how she does it and how those decisions play out has deepened trust between the Product Owner and the team.

Early in development, the team proposed the application flow as a sequential progression of three modes. Each mode had a corresponding user interface. The Product Owner objected to a transitional step between the two activities she most valued in the application. The team largely disagreed; concerned the user experience would be confusing.

The Product Owner challenged the team to find a different approach. In response, the developers went through two exercises. They documented how their family did related tasks with real world objects. Using fellow employees, they also organized an exercise using physical tools and materials to create the same work they would in the product.

The Product Owner was right. The team devised a user interface that easily transitioned between the two most compelling user activities. Further usability testing has validated this approach.

In another case, the Product Owner asked the team to explore alternatives for an organizing metaphor for work created in application. The team spent several days working out alternatives but could not find any one solution to rally around. Concerned about introducing complexity in lower value features, the team proposed making simple modifications and deferring a decision on the larger organizing metaphor.

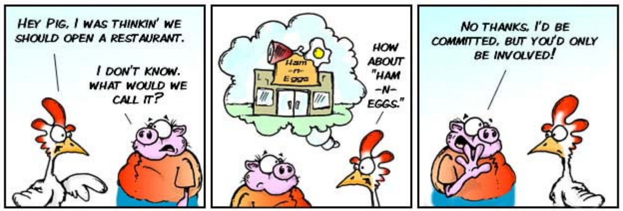
The Product Owner was unconvinced but decided to revisit the conversation at the next sprint review. By the next review, she concluded the team was right. She had shown the application to a potential distribution partner and they had given her similar advice.

In the first example, the Product Owner was in front of the team. In the latter, she acknowledged the team was in front of her. Reacting with humility to the sometimes surprising realization that the other person is right builds trust.

13. Collective product ownership is hard

Any team working towards collective product ownership should beware of the challenges.

Some disagreements cannot be resolved in the moment. The team may need more information; the situation may need to play out over time or not surrender to rational debate. The developers need to buy into decisions without necessarily achieving consensus. However, relying on majority rules results in safe decisions and a mediocre product. Sometimes the best answer is to let the individual win over the majority. On the Ript™ project, natural thought leaders emerged in specific areas. The team also relies on co-located product owner roles: UX director and product owner proxy.


HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT.
I DON'T KNOW. WHAT WOULD WE CALL IT?
Ham -n- Eggs
HOW ABOUT "HAM -N- EGGS."
NO THANKS, I'D BE COMMITTED, BUT YOU'D ONLY BE INVOLVED!

By Clark & Vizdos

© 2006 implementingscrum.com

Figure 6. The tale of the pig and the chicken [40]

Collective product ownership will only emerge in an environment where people responsible to the product outcome (pigs) are given the authority and powerful stakeholders who want to give input but are ultimately unaccountable (chickens) understand their limited role.

When unaccountable stakeholders drive decisions, they diminish the team's influence and wreck shared investment in aspects outside developments direct area of control. The Ript™ project exists in a magic bubble because the product owner has ultimate authority in the organization. The challenge for Oxygen is to expand that bubble without bursting it.

Even after the Ript project, the Oxygen team has had difficulty replicating the relationship established between the team and CEO with other product owners.

On one project, the product owner started being over-ruled by another executive mid-release breaking the integrity of sprint and clouding the release plan. The team had to "stop the line" and re-assign product ownership to the other executive, re-set the release backlog and resume work.

On another project, a short pilot arising from within the development team, the developers launched into the project without a product owner. This misunderstanding of collective product ownership broke the rule of "single wringable neck" and quickly got the project into trouble within the first sprint. Since decisions were being made without ownership by any single individual there was no one able to coherently defend the vision of the product. Quickly, one developer who had initiated the pilot was named product owner. Even when a decision is made by consensus of the team, there needs to be a single authority accountable to the outcome who says, "It's my decision to try it the team's way."

Finally, while Oxygen has an agile development team and a visionary leader, it is not yet an agile organization. As a result, the organization faces the challenge of aligning resources at a project portfolio level and of leveraging strengths of its on-air business units towards its interactive business. There is lack of consensus in the organization over the role of product ownership, the broader applicability of Scrum, and the concept of collective ownership presented in this paper. In an attempt to resolve this, Oxygen is in the process of introducing agile practices at a senior executive level and of piloting Scrum into business units close to the software development team.

"There are many reasons why your enterprise can't develop and deploy products and systems as rapidly, inexpensively, and with the quality you would like. You and your staff probably can already list many of them. Scrum won't solve them. Scrum is simply a tool that will relentlessly and ruthlessly expose them." [41]

The hard and necessary work in achieving sustained collective ownership at Oxygen has only begun. In the spirit of agile, the company needs to learn the correct lessons from its successes and its failures, to build self-directed and cross-functional teams and to methodically remove obstacles to performance as identified by those teams.

14. Conclusion

Collaboration is a common word with many definitions and many more physical embodiments.

Some collegial relationships imitate the surfaces of a more significant collaboration without altering the outcome or setting the organization up for continued improvement. Some forms of collaboration, particularly when imposed actually damage a group's ability to work jointly by distracting and de-motivating the participants or by actively hiding underlying concerns.

However in an organization containing factors for successful collaboration, joint work allows participants to engage each other at a deep level, learning from and teaching each other, challenging each other to self-improvement and the betterment of their output. This unbounded collaboration is in line with the values of Scrum and agile development.

Such learning spirals are the core competence of knowledge creating companies. Such companies have a gift for sustaining innovation, high quality, and high levels of customer satisfaction.

Oxygen Media, while not yet achieving such consistent high performance provides with the Ript™ project a tangible example of collective product ownership. In this project, the Product Owner has retained accountability while sharing authority over vision, priorities and execution. Collective product ownership emerged in the Ript™ project as a result of circumstance, the unique qualities of the Product Owner and the capacities and practices of her agile team.

XP/Scrum practice aligns authority with responsibility and allows the Product Owner to engage with the team face to face and at a level of detail.

The experience has been a joyous experience for both the team and Product Owner.

"I am a cable pioneer…an entrepreneur who has gotten to do a lot of cool things over the years. But I have to say being on the team (yes, I was welcomed as a true member of the team, not just as a CEO) that invented Ript™ is one of the coolest of all. To get to learn and try and change and innovate in a brand new medium with a group of reliable (yes they keep their word), endlessly creative and resourceful folks has been exciting for me. And the end result is as good as the process!" – G. Laybourne 12/06

15. References

[1] Schwaber, K. *Agile Project Management with Scrum.* Redmond : Microsoft Press, 2004, pp 142-143.

[2] Schwaber, K., *Scrum Release 2.0?,* Scrum Alliance, [Online] [Cited September 14, 2007], http://www.scrumalliance.org/articles/12-scrum-release-

[3] Dagstuhl-Seminar 9635: "History of Software Engineering" Schloss Dagstuhl, August 26 - 30, 1996

[4] Charette, R., *Why Software Fails*, IEEE Spectrum, [Online] [Sited September 1, 2007] http://www.spectrum.ieee.org/sep05/1685

[5] Beck, K., Beedle, M., et, al.,. Principles Behind the Agile Manifesto. [Online] [Cited: June 6, 2007] http://agilemanifesto.org/principles.html.

[6] Schwaber, K. and Beedle, M. *Agile Software Development With Scrum.* Upper Saddle River, NJ : Prentice Hall, 2002, pg 34.

[7] Sutherland, J., *The Origins of Scrum*, [Online] [Cited: September 5, 2007] http://jeffsutherland.com/scrum/2007/07/origins-of-scrum.html

[8] Hargreaves A. and Fullan M., *What's Worth Fighting for in Your School?,* Teacher's College Press, New York, 1991, pg 1.

[9] Ibid, pp 4-13.

[10] Ibid. pg 11.

[11] Schwaber, K., Yahoo Scrum Development List 3/06

[12] Poppedieck, M. and Poppedieck, T., *Lean Software Development* Addison Wesley, Boston 2003.

[13] Nonaka, I. and Takeuchi, H. , *Hitotsubashi on Knowledge Management*, John Wiley and Sons, Asia, 2004, pg 81 source Kao Corp.

[14] Hargreaves A. and Fullan M *What's Worth Fighting for in Your School?,* Teacher's College Press, New York, 1991, pg 93.

[15] Beck, K., Beedle, M., et, al.,. Principles Behind the Agile Manifesto. [Online] [Cited: June 6, 2007] http://agilemanifesto.org/principles.html.

[16] Tabaka, Jean. *Collaboration Explained: Facilitation Skills for Software Project Leaders.* Upper Saddle River, NJ : Addison-Wesley, 2006, pp 4-5.

[17] Tabaka, J., *Collaboration Explained: Facilitation Skills for Software Project Leaders.* Upper Saddle River, NJ : Addison-Wesley, 2006, pg 11.

[18] Beck, K., Beedle, M., et, al.,. Principles Behind the Agile Manifesto. [Online] [Cited: June 6, 2007.] http://agilemanifesto.org/principles.html.

[19] Schwaber, K. and Beedle, M. *Agile Software Development With Scrum.* Upper Saddle River, NJ : Prentice Hall, 2002.

[20] Schwaber, K. *Agile Project Management with Scrum.* Redmond : Microsoft Press, 2004.

[21] Larman, C. *Agile & Iterative Development: A Manager's Guide.* Boston : Addison-Wesley, 2004, pg 30.

[22] Highsmith, J. *Agile Project Management.* Boston : Addison-Wesley, 2004, pg 8.

[23] Mirriam Webster – Online, *Collaboration,* [Online] [Cited June 10, 2007], http://www.m-w.com/dictionary/collaboration

[24] Hargreaves A. and Fullan M., *What's Worth Fighting for in Your School?,* Teacher's College Press, New York, 1991, pg 47.

[25] Little, J.W., *The persistence of privacy: Autonomy and initiative in teacher's professional relations,* Teacher's College Record 91(4), 1990, pp 509-36

[26] Nonaka, I. and Takeuchi, H., *Hitotsubashi on Knowledge Management*, John Wiley and Sons, Asia, 2004, pp 167-181.

[27] Hargreaves A. and Fullan M., *What's Worth Fighting for in Your School?,* Teacher's College Press, New York, 1991, pg 55.

[28] Ibid, pg 58.

[29] *Survey: The State of Agile Development,* VersionOne LLC, 2007, pg 3.

[30] Hargreaves A. and Fullan M., *What's Worth Fighting for in Your School?,* Teacher's College Press, New York, 1991, pg 55.

[31] Mattessich, P., Murray-Close, M., et. al, *Collaboration: What Makes It Work 2nd Ed.,* Fieldstone Alliance, 2007, pg 59.

[32] Ibid, pp 7-27.

[33] Heffernan, M., *How She Does It*, Viking, New York 2007, pg 106

[34] Senge, P., *The Fifth Discipline*, Currency, New York, 1990, pg 198.

[35] Laybourne, G., (Private Communication), Apr. 7, 2007

[36] Poppedieck, M. and Poppedieck, T., *Lean Software Development* Addison Wesley, Boston 2003. pg 112

[37] Nonaka, I. and Takeuchi, H., *The Knowledge-Creating Company*, Oxford University, New York, 1995, pg 79.

[38] D. Wells, "Collective Code Ownership", 1999, www.extremeprogramming.org/rules/

[39] Laybourne, G., (Private Communication), Apr. 7, 2007

[40] Clark and Vizdos, Implementing Scrum, May 11, 2006, www.implementingscrum.com/cartoons/implementingscrum-20060911.html

[41] Schwaber, K., *The Enterprise and Scrum,* Microsoft Press, Redmond, WA, 2007, pg xi