

Values, Power, and Agility

By: Dave Hoover, dhoover@thoughtworks.com

When a software development team adopts agile principles, [MANIFESTO] it is not uncommon for the team to experience conflicts between its newly adopted principles and the values of its parent organization. When teams and organizations have opposing values, power struggles will inevitably erupt. The purpose of this article is to generate a greater awareness of power distributions and value conflicts among agile teams and their parent organizations. It is my belief that an increased awareness of these conflicts will help these groups collaborate more successfully.

As teams transition toward agility, power is necessarily redistributed. The general theme of this power redistribution is that of empowerment of the people on the front line:

- Developers provide estimates for their work and have authority to evolve any aspect of the system.
- Architects are in the trenches, actively tending to the system design.
- Testers provide automated tests that drive the development process from the outset of the project.
- Customers possess the ability to steer the project, allowing them to adapt their requirements as their business needs change.

Agile teams experience the powerful phenomenon of collective mind. [WEICK] This phenomenon is the result of a richer interrelating among teammates and the inclusion of experienced and novice people together on the same team.

To maintain agility, it is critical that an agile team keeps a watchful eye on how its values are interacting with the values of its parent organization. This article is peppered with questions to help raise the reader's awareness of the power struggles within which these value conflicts hide.

- *What are the explicit values of your team? What are the implicit values of your team?*
- *What are the explicit values of your parent organization? What are the implicit values of your parent organization?*
- *Does your team possess the authority to direct itself? What stands in the way of your team becoming self-directed?*

Merriam-Webster's web site defines "power" as the following...

power (*noun*): 1) the ability to act or produce an effect, 2) possession of control, authority or influence over others.

For the remainder of this section, I will refer to the above definitions as *first-order power* (the ability to act or produce an effect) and *second-order power* (possession of control, authority or influence over others). In the context of software development, my understanding of first-order power is the power to directly act or affect the system under

development, such as writing code. I view second-order power in this context as the power to influence the direction of the team.

- *What sorts of power do you possess in your everyday work?*
- *How have the values of your team influenced the types of power you possess?*

Developers

On a traditional team, a developer has first-order power. She has little authority over anyone or anything other than the functionality she is currently developing. Even the small amount of authority she has might be subjected to the will of the architect, depending on the precision of the design specifications.

Developers, as a whole, gain more power than any other role on an agile team. While they retain first-order power over the code they are currently developing, they are explicitly given second-order power to evolve the best possible design in continual communion with their fellow developers. It is in this communion that individual developers seemingly lose some of their first-order power. Rather than retaining exclusive rights to specific components, their authority is decentralized into the collective of developers. [CCO-WIKI]

This decentralization of authority into the development team enables a phenomenon known as collective mind. [WEICK] While the first-order power of the individual developer appears to diminish, the power of the collective is greater than the sum of its parts. In their study of flight deck teams on aircraft carriers, Karl Weick and Karlene Roberts found that

“No matter how visionary or smart or forward-looking or aggressive one brain may be, it is no match for conditions of interactive complexity. Cooperation is imperative for the development of [collective] mind.”
[WEICK:378]

This finding is congruent with the experiences of successful self-organizing teams. While variations in ability exist among the teammates, there is a consensus that no one member could have designed as good a system on her own. Thus, via collective mind, an agile developers’ first-order power has increased in breadth, if not depth.

Agile developers are granted second-order power to estimate their own development tasks. This is a critical point of change for many teams who traditionally had team leads, architects, and/or management handing down fixed-length, fixed-scope schedules to developers. Second-order power, in this circumstance, does not necessarily mean that developers have power and authority over project managers and architects, but that the power between these roles has been balanced.

- *With the adoption of agile principles, how has power shifted within your development team? Where have power struggles emerged?*
- *Would you say that your development team possesses a collective mind?*

Architects

“Exceptional designers exercise leadership through their superior knowledge rather than bestowed authority.” [POPPENDIECK:112]

On an agile team, an architect's first-order power has increased. Where a traditional architect might deliver blueprints to a team of developers, an agile architect acts as a guide, [FOWLER] rubbing elbows with developers and getting his hands dirty in the implementation of the system. He has immediate and ongoing access to the state of the system's design, whereas in a traditional environment, he might be hierarchically removed from the implementation.

An agile architect's second-order power remains unchanged, though it is manifested differently. While a traditional architect's second-order power is often imposed through hierarchical channels, an agile architect's second-order power is exercised through face-to-face collaboration with his teammates. An agile architect's authority is less likely to be experienced as coercive or authoritarian than a traditional software architect. His authority is now established by his abilities to communicate and contribute rather than through simply holding a position of power within the organization.

The need for agile architects to descend into the trenches can become a stumbling block for organizations in which architects are inexperienced in system implementation. When architects are ignorant to how to implement their own ideas, their value on agile teams is diminished. With a decreased ability to exercise second-order power via hierarchy and the inability to exercise first-order power at all, these architects will likely resist a move toward agile principles.

Mary and Tom Poppendieck use the term *master developer* to label the role that people traditionally think of as architect:

“Master developers are part of the team, enmeshed in the details of the work. They provide the leadership necessary for the team to make good decisions, make rapid progress, and develop high-quality software.” [POPPENDIECK:113]

Clearly, the common thread for the agile architect is his inclusion as a member of the development team. His inclusion is a critical enabler of collective mind and the steady improvement of the team. Weick and Roberts found that “[a]s seasoned people become more peripheral to socialization, there should be a higher incidence of serious accidents.” [WEICK:368] There are few better ways for novice developers to improve than to work closely with architects. [COCKBURN-WIKI] Interestingly enough, architects will find that they too will benefit from these interactions: “Comprehension can be increased if more levels of experience are connected, as when newcomers who take nothing for granted interrelate more often with old-timers who think they have seen it all.” [WEICK:366]

- *How do architects in your development organization interrelate with developers?*

- Would these architects fit the description of guide, mentor, or master developer?*
- *Do your organization's architects lack first-order power? How does this lack of first-order power affect the interactions between developers and architects?*
 - *What does “architect” mean in your organization?*

Testers

The relationship between developers and testers has traditionally been antagonistic. Developers work to produce functionality while testers work diligently to uncover flaws in this functionality. It is not uncommon for testers to discover defects that developers feel are simply misunderstood requirements. This can then develop into a second-order power struggle over who holds the authority to interpret requirements. This power struggle will almost inevitably bubble up to the customer, diminishing his confidence in the development process.

Agile principles call for software to be released frequently, preferably, every couple weeks. [MANIFESTO] To achieve this feat, an agile tester must test functionality in frequent bite-sized chunks rather than in huge batches. The ability to embrace changing requirements and complete iterative test cycles requires the agile tester, like the agile developer, to collaborate continually with the customer. Agile testers interpret the customer's requirements and develop tests to drive the development effort.

Robert “Uncle Bob” Martin wrote of the power shift between developers and testers:

“[Agility] completely changes the power structure between development and QA... QA finds itself in a specification role as it writes the acceptance tests that define the features. Development can no longer dump a pile of crap on QA. Instead QA dumps requirements on development.” [UNCLEBOB]

Thus, a tester, traditionally a person with only a small amount of indirect first-order power, now finds himself to be the cornerstone of the agile software development effort. While his first-order power remains indirect, this power is far more focused and prescriptive, rather than delayed and reactive. Because the agile tester is the primary supporter of the customer's requirements, his second-order power is equal to that of the customer. The agile community has made significant progress toward integrating testers into the agile team and I believe testers will find themselves with far greater power because of these efforts.

- *How are testers perceived in your development organization? How are testers perceived on your team? Are testers considered part of the development team?*
- *At what point and to what degree are testers involved in your development process? How do these factors impact the power of the testers? How do these factors impact the developers' relationships with the testers?*
- *Is there a distinction between developer and tester on your team?*

Project Managers

A project manager rarely finds herself in a position of first-order power. Her role tends to prevent her from contributing directly to the project. While an agile project manager is no more likely to write code than any other project manager, she does tend to focus herself more on removing obstacles that stifle the team's ability to make progress, [CCPACE] a sort of indirect first-order power.

The agile project manager's focus on obstacle removal does not lessen her second-order power, though it does tend to deemphasize it. In many organizations, she retains her traditional second-order power, but exercises it sparingly. The agile project manager tends to defer her second-order power to the development team, recognizing that, "the more the team relies on outsiders to make its decisions, the less control it has over its commitments." [SCRUM:45] Thus, an agile project manager recognizes that she is an outsider, and takes on the roles of development facilitator, organizational liaison, and visionary leader. [CCPACE]

- *As the liaison between your team and the organization, what value conflicts does your team's project manager experience on a regular basis?*
- *Do the organization's expectations of your team's project manager conflict with the agile principles?*

Customers

The customer gains significant second-order power on an agile team. Rather than being kept at arm's length in a traditionally antagonistic relationship with the development team, the agile team depends on frequent face-to-face collaboration with its customer. While the customer does not have absolute authority to dictate what should be delivered when, he regularly exercises his power to prioritize what should be delivered next.

The agile principles have introduced a fundamental shift in perspective on changing requirements, a historically favorite complaint of the software development industry. Agile teams embrace late changing requirements, viewing them as opportunities to increase the value of the system. [MANIFESTO] Thus, agility increases the customer's second-order power not only in depth, but also in breadth.

- *Would your team's customers agree that their power has increased since your transition to agility? What was it that facilitated this increased power for your team's customer?*

The Power of the Familiar

"The power of the familiar is very strong, often stronger than the wish to change. Strong interventions, lots of patience, and continual awareness help us challenge the power of the familiar." [SATIR:212]

One of the chief frustrations with adopting agile methodologies is that the values set forth

by an agile approach often stand in direct opposition to the implicit values of the larger organization. Even if a team is successful at integrating the principles of agility, the values and practices of the larger organization will act as a consistent and powerful force, pulling the team back toward the values of the organizational system.

A development team in an organization is not unlike a child in a family. While a child may choose to adopt a lifestyle that contradicts the traditions of the family, it is unlikely that the family is going to suddenly embrace that lifestyle as its own. It is more likely that the child will gradually revert to her family's traditions. This likelihood has much to do with Virginia Satir's notion of the power of the familiar:

“Familiarity exerts a powerful pull. What we have observed and experienced day after day exerts a powerful influence. Most people will choose the familiar, even though uncomfortable, over the unfamiliar, because of that power.” [SATIR:144-145]

Beyond the power of the familiar lies the child's own power within the system. As a child grows physically and matures psychologically, her power will necessarily increase. Eventually, the power of the adult child will rival that of her parent. Yet, as the child grows in power, the power of the familiar grows with her, making it increasingly difficult for her to make fundamental changes in her lifestyle.

Similarly, a software development team that chooses to adopt a process that defies the values of its organizational system will not only face resistance from the power of the familiar, but also from the power of the organization itself. Like an adult child in a family, a team that holds a position of power within the organization will likely see its newly adopted agile principles spread into the organization. Conversely, if the organization is very large or the team immature, the team will likely struggle to maintain its newly adopted principles and gradually revert to the principles of the organization.

“A single new member, or even a single new group within so large a group, really has no chance of converting the social system, even if he is firmly convinced of the correctness of his way of doing things.” [WEINBERG:63]

An exception to this idea occurs when a team has an executive sponsor, a champion that shields them from organizational pressure. Imagine what a difference it would make to have a well-respected uncle stand up for a child in the midst of a family conflict. This high-level support can be a critical enabler of the adoption and maintenance of agile values within a traditional organization.

In most cases, the success of a software development team's transition toward agility will be dependent on transitioning the parent organization concurrently. This may mean that the team will have to persistently engage its parent organization in order to spark an organizational transition. Alternatively, whatever force initially triggered the transition in the team may need to work concurrently in the organization. Either way, it is vital that both the team and the organization stay aware of how power is shifting as hierarchies

collapse and agile values emerge.

- *During your team's transition toward agility, when was it most difficult to resist the power of the familiar? How was the team able to establish new principles and practices in the face of the power of the familiar?*
- *How much power does your team hold within the organization?*
- *Using the metaphor of the child and the family, how would you describe your team?*

The Power of Awareness

“In habitual action, each performance is a replica of its predecessor, whereas in heedful performance, each action is modified by its predecessor.”
[WEICK:362]

Families that struggle with change and conflict sometimes employ family therapists to help the family improve itself. Similarly, organizations struggling with conflicting values or the inability to adapt can benefit from hiring outsiders to facilitate change.

Agile teams meet at regular intervals to reflect on how to become more effective.
[MANIFESTO] A retrospective [KERTH] is a common agile practice that is often facilitated by someone outside the system, though internal consultants or coaches can facilitate them as well. While an entire organization does not usually attend a retrospective, the process includes anyone that the team interacts with on a regular basis.

Retrospectives, when done regularly and successfully, can significantly increase an agile team's awareness of how its values are integrating into the organizational system. Furthermore, by reflecting on its own experiences, both the power of the familiar and the distributions of power become more visible to the team. This increased awareness provides an agile team with the ability to adapt and evolve itself, along with engaging its parent organization in intelligent dialogues about problematic value conflicts and power struggles.

- *How well do your team's values integrate with the values of the agile manifesto?*
- *How well do the values of your parent organization integrate with the values of your team?*
- *What value conflicts lie beneath the surface of the power struggles your team experiences?*

Conclusion

The power of agility is seen in the empowerment of the front line workers by releasing authority from hierarchical structures.

“The people on the front line combine the knowledge of minute details with the power of many minds. When equipped with necessary expertise and

guided by a leader, they will make better technical decisions and better process decisions than anyone can make for them.” [POPPENDIECK::xxvi-xxvii]

When value conflicts exist, power struggles will surface and the power of the agile principles will be diminished. To maintain agility, it is critical that an agile team keeps a watchful eye on how its values are interacting with the values of its parent organization.

Acknowledgements

Patrick Morrison and Laurent Bossavit provided excellent guidance and feedback on this article. Rebecca Parsons and Boris Gloger brought up some important issues that I had missed. Thanks for your time everyone.

References

- [CCO-WIKI] <http://c2.com/cgi/wiki?CollectiveCodeOwnership>
- [CCPACE] Sanjiv Augustine and Susan Woodcock, *Agile Project Management*, <http://ccpace.com/Resources/documents/AgileProjectManagement.pdf>
- [COCKBURN-WIKI] Alistair Cockburn, <http://c2.com/cgi/wiki?ExpertInEarshot>
- [FOWLER] Martin Fowler, *Who Needs an Architect?* <http://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>
- [KERTH] Norm Kerth, *Project Retrospectives*
- [MANIFESTO] *Principles behind the Agile Manifesto*, <http://agilemanifesto.org/principles.html>
- [POPPENDIECK] Mary and Tom Poppendieck, *Lean Software Development*
- [SATIR] Virginia Satir, *The New Peoplemaking*
- [SCRUM] Ken Schwaber and Mike Beedle, *Agile Software Development with Scrum*
- [UNCLEBOB] Robert Martin, March 12, 2004, <http://groups.yahoo.com/group/chicago-agile-dev/message/1546>
- [WEICK] Karl Weick and Karlene Roberts, “Collective Mind in Organizations: Heedful Interrelating on Flight Decks,” *Administrative Science Quarterly*, 38 (1993): 357-381
- [WEINBERG] Gerald M. Weinberg, *The Psychology of Computer Programming*