VALVE'S ORGANIZATION

OPPORTUNITIES AND OPEN QUESTIONS

GEORG VON KROGH • NINA GEILINGER

Valve is a very interesting case study! The company shares many features with open source software projects. In Valve, as in the open source world, the focus is on creating advanced technologies and letting developers self-select projects and tasks. Self-selection seems to work particularly well in an environment where technology development itself has a coordinating function. In software development, people have a strong bond of common knowledge based on design philosophies and approaches, language and terminology, and engineering tasks. In most cases, they also share a similar educational background and/or development experience that helps them implement creative ideas in physical machines and graphic environments. Developers rarely rely on detailed instructions from higher-level managers because they already have the implicit knowledge of what needs to be done. When inconsistent views about development come to the forefront, they are best reconciled where the knowledge resides: with the experts. Moreover, it appears that Valve has uncovered how developers' self-selection of projects and tasks can be a powerful motivator to unleash their creativity.

How does Valve's flat hierarchy impact other aspects of its organization? The media have claimed that an uncontrollable "hidden layer of powerful management" arose in Valve to fill the power gap created by its absence of formal management and that this hidden management led to the formation of cliques and the troubling discontinuation of more radical projects. We suspect, however, that informal hierarchies form and influence decisions wherever people come together and that the flat hierarchy at Valve was not the sole reason why certain projects were abandoned. While we do not have hard evidence, we suspect that the tolerance of conflict and discussion in flat hierarchies enabled, rather than prevented, radical innovation at Valve. Valve's recent development away from a pure video game developer toward a hardware company and its reinvention of the entire video game market with its platform Steam, indicate that Valve's non-standard form of organizing may not have hampered radical innovation in products, technologies, and business models. In fact, it may have facilitated such innovation.

We suggest that the key question to ask in the case of Valve is how leadership fills the void left by the absence of many of the functions and tasks that exist within a traditional hierarchy. For example, it may be that at Valve, developers exercise distributed leadership rather than authority-based, top-down leadership. Developers in teams share leadership roles and responsibilities, with fluid authority based on tasks, knowledge, and interests (von Krogh, Nonaka, & Rechsteiner, 2012). Such self-driven leadership means that each team member must have the capacity to temporarily act as a leader. Over time, team members get to know each other, and relationships evolve and ease the rapid succession of authority. Yet, to become leaders and gain authority, team members must share their knowledge and reveal their interests. By doing so, team members can take on leadership activities in ways that are both comprehensible and legitimate.

Furthermore, the case of Valve holds an important lesson for us as management scholars: be sensitive to the role of technology in organizing. We need to determine which coordination technologies (e.g., project support tools, wikis, engineering tools) Valve's employees use and how such technologies reinforce distributed leadership within Valve's flat hierarchy. We must understand how often simple and lightweight tools replace formal communication channels. Such findings will help us strengthen the link between organization design and technology. Technology clearly cannot replace face-to-face coordination completely. In Valve's offices,

Journal of Organization Design
JOD, 4(2): 18-19 (2015)
DOI: 10.7146/jod.20159

new project team members still roll their desks together. This physical presence may in fact be what enables distributed leadership.

We think that empowerment enhances creativity and experimentation, and that consensus building through distributed leadership could be an ideal approach for resolving conflict. However, such practices also carry risks. Consensus building is time consuming, and organizations with distributed leadership may appear less agile. They may even appear to lack resolve or consistency toward external stakeholders. Consider a recent example on Valve's platform, Steam. Valve removed a game called Hatred, which was created by another video game development company, from Greenlight (Steam's community area, where users can vote on which new games should be sold in the main digital store). Valve later added the game again after its CEO reviewed the decision internally and concluded that the decision to remove the game was not a "good" one. In a recreational products market, such inconsistencies might not seriously endanger a business; at most, they may lead to a short-term uproar in the media and the user community. However, when an organization's resolve and consistency in decision-making are critical to its customers' welfare and the firm's legitimacy, image, and identity, some natural limits to distributed leadership and consensus building may emerge. Many software firms facing such constraints implement strict quality assurance, heavy legal compliance procedures, and the formal testing of products throughout the development cycle – elements that Valve seems not to have prioritized in its culture. As management scholars, we should ask how distributed leadership fares in firms building critical software for business organizations (e.g., transaction systems or secure trading in banks). Such firms may implement flat hierarchies and distributed leadership in some parts of their value chain (e.g., in R&D) but only in combination with more traditional designs and top-down leadership in other parts (e.g., in production, sales, and marketing).

Finally, the question about scalability remains an open one. Flat hierarchies may be rare in industries other than IT and in large corporations (well-known exceptions exist, such as Ricardo Semler's Semco or Pentagram). However, the story may be more intricate than one of structural design or technology. The open source phenomenon has repeatedly demonstrated that flat hierarchies can support massive and stable technology development that coordinates thousands of users and contributors. A cardinal principle of such organizations seems to be a role-model founder who walks the talk, stands strongly behind the principles of distributed leadership and empowerment, and imprints those principles upon the organization's identity (e.g., GitHub or Wikipedia). Such founders seem to infuse their organizations with distributed leadership and do so by practically demonstrating how such leadership can work. That is no small feat!

REFERENCES

von Krogh G, Nonaka I, Rechsteiner L. 2012. Leadership in organizational knowledge creation: A review and framework. *Journal of Management Studies* 49(1): 240–277.

GEORG VON KROGH

Professor
ETH Zurich
E-mail: gvkrogh@ethz.ch

NINA GEILINGER

Ph.D. Candidate
ETH Zurich
E-mail: ngeilinger@ethz.ch