Journal of Organization Design
a SpringerOpen Journal

ORGANIZATION ZOO Open Access

GitHub: exploring the space between boss-less and hierarchical forms of organizing

Richard M. Burton[1], Dorthe Døjbak Håkonsson[2], Jackson Nickerson[3], Phanish Puranam[4], Maciej Workiewicz[5*] and Todd Zenger[6]

* Correspondence:
workiewicz@essec.edu
The authors are in alphabetical order.
[5]ESSEC Business School, 3 Ave Bernard Hirsch, 95000 Cergy, France
Full list of author information is available at the end of the article

Abstract

In this edition of the organizational zoo series, we take a closer look at an interesting organization design case—GitHub, a software company from California. Similar to Valve, the subject of the previous article in the series (Puranam and Håkonsson, J Organ Design 4: 2–4, 2015) GitHub is used to delegate the choice of projects and project allocation to its workers, fitting the recent trend in running organizations without bosses. The interesting fact about GitHub is that after years of praising its own unorthodox organizational structure, the company suddenly decided to abandon it for something much more traditional. We asked several renowned organization scientists to share their thoughts on this interesting case and discuss what we can learn from it.

Keywords: New forms of organizing, Organizational design, Boss-less organization, Self-allocation, Organizational change

Introduction

Dorthe Døjbak Håkonsson–Maciej Workiewicz

GitHub is an American software company located in San Francisco. Founded in 2007 by Tom Preston-Werner and Chris Wanstrath, the company has experienced an unprecedented growth in sales and subsequently in number of employees. The company doubled in size from 300 employees in the beginning of 2015 to 600 in November 2016. The company's product, also called GitHub, allows companies and individual developers to manage their computer code and collaborate with others on software projects. GitHub has been used by some of the biggest software companies like Google, Adobe, Twitter, Paypal, Yahoo, Facebook, and Dropbox. The company's free service became a popular tool among the members of the open source software community, students, and hobbyists. As of this writing, 14 million of users were collaborating on over 47 million projects using the company's product in its second round.[1] In July 2015, the company raised 250 million dollars from private investors, putting the overall value of the startup at two billion dollars.[2]

After 7 years of being run without formal hierarchy, in 2014, the company initiated a substantial reorganization, effectively dropping the boss-less form. From an organizational design perspective, this abrupt change raises some interesting questions regarding the boundary conditions of boss-less forms of organizing, which has been gaining significant attention in the academic and managerial circles.

Before the sudden change in structure, GitHub 1.0 (pre-2014) was a poster child of a boss-less organization. The company was openly proud of its approach to running the business. In fact, their organizing principles were similar in key values to how users of its product were collaborating on open source projects. In that sense, GitHub, the company, was a reflection of the values around which GitHub, the product, was built.

Open allocation

At GitHub 1.0, employees could start and join projects where they felt they could contribute the most. The projects were not hierarchically assigned, but employees could choose whatever they wanted to work on without any formal interference. The minimum condition was that there were at least two people on the team (the rule of two). Zach Holman, one of the employees summed it up as follows: “Teams are all informal. If you want to work on something, just go work on it. If you need a designer, grab a designer. Boom, a two-person team.”[3] He also remarked elsewhere that “We have teams, but the teams are set up so that they are easy to move around. And that’s the problem with teams if they are very structured so that you can’t get out of them.”[4]

No formal hierarchy

At GitHub 1.0, employees did not have official titles, instead many came up with whimsical ones, like “Developer Superhero,” “Bleeding Edge Cowboy,” and “Happiness Responsibilibuddy, Magician.” Coders could design, and designers could (and were encouraged to) code. There were no middle managers and no product, departmental, or project managers. Scott Chacon, one of the first employees, said: “We try to move decision-making capabilities to people closest to the problem. [Projects have] a ‘primary responsible person,’ or PRP, in open allocation lingo.”[5]

Maximizing employee happiness

Famously, the key objective at GitHub was to optimize employee happiness. Employees were free to choose their location, workdays, work hours, and how to dress to work. The belief was that happy employees are more productive and thus the company more successful. As one GitHub employee said: “There are a number of companies that have this 20% rule where employees are encouraged to work on stuff they are personally excited about outside of their main responsibility 20% of the time. I think in some ways you can say GitHub has a 100% rule, which is basically always be working on stuff you are personally excited about or that you find fulfilling.”[6]

Distributed workforce

The labor force was very much distributed with over half of the employees working remotely (the company’s only office until August 2014 was in San Francisco). To keep people up to date on new hires and recent developments, the company had its own film crew Streaming Eagle, which produced and disseminated video profiles of new employees and recordings of company meetings.

Conflict resolution through consensus

Conflicts were resolved by discussion, without intervention of senior managers. Other employees with high reputation and respect could be consulted on a specific matter, but the decision rested with the team. Employees used the company's own product to create requests for changes to a given problem and lobby other employees for support to have their version adopted.

Informal hiring and performance evaluation process

At GitHub, prior to 2014, there were no HR people. The pay was fairly uniform. The company boasted that people came to work to GitHub because they admired the product and culture. The salary was believed not to be the prime motivator.[7]

Organizational change—GitHub 2.0

Things began to change in early 2014, when the new CEO, and fellow co-founder, Chris Wanstrath, took over from Tom Preston-Werner. The biggest change for GitHub was the introduction of rules and processes to increase coordination—mostly through improved communication. According to Chris Wanstrath, "Before, we all had to share the road maps in our head. I might be building something here, not even knowing that Paul is building something over here. That worked out fine until we got more people and decided we needed a lot more coordination, a lot more communication."[8]

GitHub in 2015 is quite a different place from what it used to be. It still offers the same innovative product and has the same White House Oval Room replica reception, but the organizational structure is much more traditional. The company now has VPs, a robust HR function, product managers, technical leads, and a one-to-one reporting structure. The projects are now allocated at special meetings, and the employees undergo official performance reviews. The company is also starting to build a physical presence. In August 2014, GitHub opened its second office (after San Francisco) in Boulder, CO, and the first non-US office in Tokyo in June 2015.[9]

While some may perceive it as quite a departure from the old philosophy, the company has been very pragmatic in its choice of organizational structure. Somewhat foreshadowing the changes to come, the then-CEO Tom Preston-Werner said in October 2013: "A year from now we might be using something that's not open allocation, but is the evolution of open allocation, as we've determined that something else works better. And that's really fine."[10]

What can we learn from organizations like GitHub?

Until 2014, GitHub seemed a prominent representative of new trends in organization design towards non-hierarchy. Successful examples from a variety of different industries would suggest that this type of organizational form can be applied to a variety of environments. This leads to the question: what does hierarchy do? And when (if ever) do we need it?

Yet, GitHub today is not the GitHub it used to be. It has the same product, and serves the same customer base, but it now has a formal hierarchy. What can we make of this shift? Are there limits to the effectiveness of non-hierarchical forms? What is their selection environment? Do they only apply for a particular type of employees? Or work culture?

Finally, what might the future bring for GitHub? Was hierarchy the right response to their new demands? Will their new structure pertain? Or will their simultaneous demands for control and autonomy lead to ongoing design changes in the future?

Is hierarchy necessary?

Richard M. Burton

GitHub began without hierarchy; now, it has a traditional hierarchy. Why? Can an organization thrive or even survive without hierarchy? If so, under what conditions or contingencies can it survive? Is hierarchy necessary?

Hierarchical organizations are not new: Moses took his thousands to hundreds on down to tens; the Roman Army had its centurions which were aggregated up to legions; the Roman Catholic Church is hierarchical and is long lived. In contrast, we do not have many examples of longed lived non-hierarchical organizations. In this short essay, I proceed from hierarchy to no hierarchy—the reverse of GitHub. The GitHub zoo story is told the other way around—no hierarchy to hierarchy. Here, we will begin with hierarchy and see what might happen if we remove the hierarchy from the organization. That is, can we take hierarchy away? Can an organization exist without hierarchy—even for a short time?

Hierarchy and rules

Weber's (1948) essay on the elements of Bureaucracy is a fundamental statement of organization. Weber, p. 197, established the basis for hierarchy:

> The principles of office hierarchy and of levels of graded authority mean a firmly ordered system of supervision of lower offices by the higher ones. Such a system offers the governed the possibility of appealing the decision of a lower office to its higher authority, in a definitely regulated manner.

The higher office supervises the lower office, and a lower office can appeal a decision to a higher office, but not the reverse. Decision-making and appeal are involved. Weber goes on to say that hierarchy is prevalent in private and public organizations. Even before hierarchy, Weber (1948) introduced rules, knowledge, and task specialization of work as fundamental: "The management of the office follows general rules, which are more or less stable, more or less exhaustive, and which can be learned. Knowledge of these rules represents a special technical training which the officials possess." p. 198. That is, there are rules for most everything that can occur; further, individuals in the organization can learn these rules with training. These rules and their implementation are in principle sufficient for all organizational issues. Weber (1948) elaborates: "There is the principle of fixed and official jurisdictional areas, which are generally ordered by rules that is, by laws or administrative regulation." p. 196. "…The reduction of the modern office management to rules is deeply embedded in its very nature". p.198 "…only persons who have the generally regulated qualifications to serve are employed".

Hierarchy, rules, specialization, and qualified individuals are elements of the bureaucracy. Is the organization over specified? If the rules are complete as Weber indicates, can an organization exist without hierarchy?

In GitHub, the rule of two is a very powerful rule. The rule of two directs effort by all of the employees; equally important, it throws out what will not be done if only one person thinks it is a good idea; it helps develop a climate of cooperation among the employees as they must interact if their projects are to be undertaken; and it encourages coordination among the projects which are undertaken. There are also other rules or heuristics which are very important: the hiring practice of employees finding individuals who are technically competent and fit the GitHub climate or culture, the rule that conflict would be resolved through discussion and consensus. Now, GitHub's motto "In Collaboration we Trust" is a statement of value, who we are, and how we will proceed. Put together, there are many rules in the Weberian sense which define GitHub.

Why hierarchy: the contingencies

Do we need hierarchy? There are good reasons for hierarchy which suggests that contingencies are important. Weber (1948): "The development of the money economy, in so far as a pecuriary compensation of the officials is concerned, is a presupposition of bureaucracy", p. 204 "…precision, unity, strict subordination, reduction of friction and of material and personal costs – these are raised to the optimum point in the strictly bureaucratic administration, and especially in its monocratic form. As compared with all collegiate, honorific, and avocational forms of administration, trained bureaucracy is superior on all these points". p. 214. In short, with trained employees, the bureaucracy is efficient with precision, unity, strict subordination, and the reduction of friction as required in a money economy with competition. In terms of contingency, the economic environment which requires efficiency creates a need for bureaucracy. However, Weber (1948) did not argue that bureaucracy is adaptable to the new environment or technology. It is a machine organization; it is stable, but not necessarily adaptable to change. Going back to GitHub, it worked well with rules and without hierarchy when there was no need for an organizational change and no conflict or friction and resources were plentiful. Efficiency was not the main issue, but new products were, which does not necessarily imply a change in organization.

Is hierarchy necessary in Weber's (1948) bureaucracy? To be sure, Weber (1948) did not consider this question: what would happen if the hierarchy were to be extracted from the bureaucracy? What would be left and would it work—or how well would it work? Without the hierarchy, there would be rules of what to do which would be documented and widely known among the employees; there would be employees who know their jobs and can do them well; there would be precision of action; there would be little friction as individuals follow the well-known rules; there would be no reason to appeal a decision. In principle, hierarchy would not be needed; it is not necessary. We might say that Weber's bureaucracy is over specified; the rules and culture are sufficient.

Rules: sufficient or not

Baligh's (2006, Chapter 3) theory of organization structures includes the elements: people, decision rules for operating variables, information structures, and reward structures. The organization structure is then the relations among these elements to obtain desired outcomes, e.g., efficiency. The decision rules should have properties of consistency and rule addition. For people, enfranchisement, independence, or freedom to make rules is needed. The reward system is tied to people by those who make the rewards and those who receive them. Ownership, involvement, consistency, and goal based rewards are important. Baligh's (2006) model is a complete statement of the organization without hierarchy in the traditional authority notion. Here, hierarchy is how the rules are put together and their relationship among the rules. Baligh (2006) would not preclude the traditional hierarchy, but it is not needed. The relationships among people, operating rules, information, and rewards are sufficient. Unlike Weber (1948) who does not deal with change, Baligh (2006) incorporates mechanisms to change the rules as new technologies and environment demand. That is, there are rules to change the operating rules. At GitHub, it does not appear that there were rules to change rules, e.g., the rule of two needed a hierarchy to change.

Then what happens when operating rules are less than perfect and things go wrong? This is exactly what the hierarchy is to do: interfere when things go wrong, i.e., the appeals of decisions (Weber, p. 197). That is, the rules are not complete in the Baligh (2006) sense: not everyone knows the rules or follows them; the rules do not fit the situation; the rules may be incomplete; the rules may be inconsistent, no one is assigned to change the rules. The rules are imperfect, and the hierarchy is a mechanism to deal with this situation. This organization might work in a very stable situation where there are no exceptions or variations, no change, and no new strategy which requires new rules or specializations. It is an idealized world which is rare—temporary at best.

Information processing and coordination

In 2014, GitHub also introduced hierarchy, new rules, and processes to coordinate projects as the rule of two was not sufficient for the allocation of effort and coordination of the employees' activities. Weber did not explicitly consider the need for coordination and did not invoke information processing, but it seems implicit that his officials need to be coordinated. That is, Weber's (1948) specializations must work together. We might infer that Weber took coordination for granted, although he left it implicit. Clearly, his bureaucracy was not composed of independent units; both the hierarchy and the rules brought the specializations or work at the desks together for overall coordination. Many modern writers (e.g., Puranam et al. 2012) are quite explicit that coordination of activities is fundamental for organizing. Using information processing frameworks, the hierarchy is needed in organizations to obtain coordination among subunits.

The information processing view posits that an organization uses information to coordinate and control its work under uncertainty. The organizational capacity to process information—oral and written—is limited by individuals and the communications among people. Thus, the organization assigns the information processing activities to individuals and organizational subunits so that the organization can coordinate its work in an efficient manner and obtain desired outcomes.

Galbraith (1974) framed the organizational design challenge as the balance of the organization's information processing capacity with the information processing requirements. That is, in order to make good decisions, coordinate, and control, the information processing capacity must exceed the uncertainty in the work tasks and the environment. With new IT technologies, the organization can coordinate up and down the hierarchy more quickly and with more detail. In 2014, GitHub replaced the rule of two with hierarchical decision-making to allocation and coordination between tasks. GitHub's earlier non-hierarchical organization with the rule of two reduced the need for information as projects could be selected without reference to the overall firm coordination; yet, coordination was voluntary and emergent. This earlier structure may have required excess resources and inefficiency. In 2014, GitHub's scarce resource was talent and it needed to utilize its people's skills well. In short, it had to choose its projects to meet customer demands and coordinate its activities well. At the extreme if there are no scarce resources and projects do not need to be coordinated, then the information requirements are minimal and the rule of two would work well.

Burton et al. (2015) multicontingency theory is an information processing approach for the firm to plan and coordinate its activities to implement its strategies and achieve the firm goals. The organizational design problem is stated in three parts: the overall organizational task, the decomposition of this large task into smaller tasks which individuals and subunits accomplish, and the coordination and control of these smaller tasks so that the overall task is realized. There are two central questions here: how to create the individual subtasks and how to put them together. Without a hierarchy, in GitHub, the projects were determined by the rule of two. The coordination of these subtasks into a coherent whole information system which meets the customers' demands was voluntary and emergent. GitHub's climate which included discussion and consensus was important. But was it enough?

In the multicontingency theory, the configuration or organization form and the task design specify how the overall task is decomposed into smaller tasks. The coordination of these pieces into a whole is accomplished with a fit among the people and their skills, the leadership and organizational climate, coordination, control, information and knowledge systems, and incentives. These several elements must fit together to achieve the firm's strategy and overall goals. These elements work within a configuration hierarchy which can be functional, divisional, matrix, or simple. With the rule of two as a way to select projects, GitHub did not need a configuration hierarchy to select projects. But the coordination of these projects may have been problematic. Opportunity losses could have been the following: (a) the selected projects did not fit the overall strategy; (b) the selected projects could have been both redundant and left holes in the strategy, and the allocation of employee effort to the projects could have been inefficient; (c) the timeliness of the projects to meet customer needs could have been an issue. There are many possibilities, but as long as there were paying customers and slack internal personnel resources, it was not a problem; most any organizational design would work. GitHub's hierarchy with VPs and middle management changed the choice of projects to meetings. A HR department was introduced. These changes were consistent with needed elements for fit in the multicontingency model. Within the multicontingency model, what would happen if the hierarchy were to be taken away? Would the remaining elements be sufficient? The remaining organizational elements coordinate the activities by providing the right information to the right individuals at the right time. GitHub may have suffered the same problem with its rule of two. It is conceivable that the firm's activities could be selected well. But it is less certain that the projects would be coordinated. The multicontingency model without hierarchy has severe limitations; it leaves too many important issues unresolved.

Concluding comments

In this exploratory essay, I have used the GitHub firm as inspiration to examine what would happen if we removed the hierarchy from several organizational models: Weber's (1948) bureaucracy, Baligh's (2006) rule-based model, Galbraith's (1974) information processing model, and Burton et al.'s (2015) multicontingency model. With the exception of Baligh (2006), these models have hierarchy. When the hierarchy is taken away, rules are then the central organizational mechanism for organization.

GitHub has moved from no hierarchy to hierarchy. Is hierarchy necessary? For GitHub, it is an intriguing story which is still being played out. More generally, it is an open question. We know that a hierarchy yields stability; is no hierarchy inherently unstable? Can it be agile? Is no hierarchy just a temporary solution, which can be changed and adapted quickly? Or should we conclude that without any hierarchy, change is easy and can be normal; bureaucracy makes change slow, but possible. Are the opportunity losses of a non-hierarchical organization too large? In good times, we can do away with hierarchy and move on; when hierarchy is needed, we use it and then discard it. Can we move back and forth as GitHub case seems to suggest?

GitHub creates more questions than answers. The knowledge canvas is largely blank; there is much to do. We know that no hierarchy is rare with only a few examples; hierarchy is everywhere. Is hierarchy necessary? It seems that no hierarchy can be temporary at best and probably short lived.

Acknowledgements

My thanks to Sean Fath, Dorthe D. Håkonsson, and Borge Obel for their helpful comments on earlier drafts.

GitHub and the "boss-less" organization: adjustment costs, selection, and vacillation

Jackson Nickerson

Haakonsson et al. (2015) introduced us to GitHub in their organizational safari and posited three questions. First, is GitHub 1.0 representative of an innovative and valuable new type of "boss-less" organization? Second, what can we make of the organizational shift associated with GitHub 2.0? And third, what might the future bring for GitHub? In the remainder of my comment, I offer reactions to all three questions.

GitHub 1.0 and the boss-less organization

Is the "boss-less" GitHub 1.0 an innovative and valuable new type of organization? My first reaction is to think of GitHub 1.0 as a trial, a potential organizational innovation. All too often, we think of innovations in terms of products and services, but organizational forms also are potential innovations. As such, relevant questions are the following: what is this new boss-less form of organization good for? Under what boundary conditions or parameters might the form create and capture value? And what implications does the boss-less organization create for its future? Addressing these questions can begin to unpack the conditions under which GitHub's organization might yield positive performance.

Selection environment

Before exploring these specific questions that can help to assess the organization's value, I believe that we can benefit by first understanding the selection environment in which the organization operated. In the absence of selection pressures, practically, any organizational form might perform well! Is GitHub a unique form or is it one of many feasible alternative forms that might be unlikely to survive when subjected to increased competitive pressure? My general sense is that during the initial years of the organization, selection pressures were limited if not nonexistent. This lack of pressure means that the boundary conditions for the boss-less organization may be substantially narrower when exposed to a harsher selection environment and, comparatively, alternative forms might have performed equally well or better if given the chance.

Degree of decomposability

All organizational forms have boundary conditions—the parameter ranges within which an organizational form can create and capture value. The existence of competing organizational forms may yield even narrower parameter ranges, where the organization has a comparative advantage over other organizational forms.

If *all* tasks, adjustments, and adaptations can be specified ex ante and execution verified ex post, then I do not think a boss is needed for an organization (although a negotiator might be needed). A technology, which is simply the method for producing an output, that is highly decentralized and decomposed into independent modules (what Simon (1962) referred to as decomposable problems) with few workers supporting each module, simply does not need a boss to adapt to disturbances (otherwise known as shocks, unanticipated problems, ongoing learning, shifting internal and external environments, etc.). In such fully decomposed problems, coordination occurs autonomously by each individual (or small team) because of the lack of complicating interactions with other activities. But if complex linkages exist between the different modules or reputations and other aspects spillover from one module to another, even if these linkages and spillovers are weak, then unanticipated adaptation and adjustment needs create pressures to coordinate. Some type of coordination mechanism is needed to support mutual adaptation and adjustment.

In my view, "boss-less" organizations are likely to break down (i.e., perform poorly) when disturbances create ongoing and real-time unanticipated adaptation needs that require coordination. These adaptation needs put pressure on people to adjust and adapt in a coordinated way, when the underlying activities, processes, or technology are interconnected. As Simon might have said, such conditions arise when the problem elements are nearly or non-decomposable. Put differently, bosses typically serve the purpose of constraining subordinates' choices to gain some type of coordination across activities. Constraints are needed because we human beings frequently desire some level of autonomy and self-determination. We generally think differently, have different information and knowledge sets, have different motivations, and, as a result, frequently may not make choices in concert.

Problem, context, and model of human behavior

I often think about three factors when elaborating boundary conditions of a particular organizational form. Boundary conditions are related to understanding the *type of problem* for which the organization is good at finding a solution. Building on Herbert Simon and others, Nickerson and Zenger (2004) have written about how the attributes of the problem (complexity and decomposability) can be matched with the costs and competencies of alternative organizational forms (what we call governance structures, following Williamson 1996) to facilitate the search for valuable solutions. Simon's (1973) problem ill-structuredness (see also Macher's (2006) more recent treatment for organizations) offers another important problem attribute. The problems' attributes that an organization can successfully tackle provide one aspect of an organization's boundary conditions. But, problem attributes are not the only factors through which to assess the usefulness and boundary condition of an organizational structure.

Another factor is *context*. In particular, I think of how long it takes to build, execute, and interpret an experiment or pilot (organizational or otherwise) in response to an actual or perceived problem as an important contextual factor (see Furr et al. (2016)). For instance, a coder can quickly write software code, compile it, and then quickly evaluate it if the code works or not. If the software does not work, the coder can quickly fix it and try again. If such experiments and pilots are very low cost, then responding to disturbances also is low cost and sophisticated coordination mechanisms and governance may not be needed. But as soon as experiments require investments that take substantial time and resources to assemble, utilize, and interpret the results, then the cost of mistakes and poor coordination grows dramatically. This conjunction of problem attributes and context in which experiments and learning takes place can create substantial adaptation demands. Boss-less organizations are likely to work well only for contexts in which experimental/pilot costs are low.

A third factor is the *nature of the person.* The field of organizations often utilizes a simple model of human behavior, although all too often, the model is not specified. Human behavior and cognition need to be accounted for in the design of the organization. For instance, see Foss and Weber's (2015) recent paper reflecting on the need to better understand and specify behavioral and cognitive assumptions.

Additionally, an organizational design itself can create an environment in which people self-select to join an organization based on an attractive set of values, ideals, missions, and expectations (think about why people join not-for-profits, volunteer organizations like Wikipedia, or work for the federal government). As a result, self-selection may interact with the organizational mode to deliver a strong identity and high performance that is not justified simply by the mechanisms within organizational form alone. For instance, individuals might be attracted to GitHub 1.0 and strongly embrace the corresponding organizational identity. Unfortunately, this strong identity has a downside in that it greatly raises adjustment costs, should a different organizational form be needed in the future. Therefore, strong identity with a boss-less organization is advantageous so long as future changes to the organization's structure and identity are unlikely.

GitHub 2.0 and bosses in organization

With shifting customer demands, the expansion into new and different customer segments and problems, and what I suspect, is a more challenging selection environment, GitHub 1.0 transitioned to GitHub 2.0. The introduction of hierarchical elements into the organizational structure should not have come as a surprise. Serving corporate clients where reputation is important and tackling larger projects involving more programmers are just two of the changes that may have ushered in adaptation needs for which bosses provide an appropriate organizational solution. I am confident that the infusion of management and leadership met with at least some (if not stiff) resistance because of strong identities formed by joining the boss-less version of the organization. Making the decision to switch governance as well as leading organizational change may have taken time, effort, and resources because adjustment costs to resolve coordination problems and change employee expectations and identity.

In business, many vital aspects can change over time. Especially important is the changing nature of demand as well as the selection environment. I remember reading during the dot com boom of the late 1990s a lot of hype about new organizational forms. All too common were statements like the world has changed and all organizations will have to adopt the new form or die. In 2000, the selection environment changed, the nature of demand seemed to change, and the hype gave way to more realistic assessments of organizational forms. Taken together, a working hypothesis is that boss-less organizational forms may be innovations but the boundary conditions may imply a limited range for their usefulness.

My comment suggests that weak selection environments and decomposed and well-structured problems in a context where searching for solutions is low cost may be a starting point for more precisely identifying boundary conditions of GitHub's boss-less organization. If the boss-less organization generates a strong identity among those who are attracted to it then adjustment costs for future organizational shifts may be higher—perhaps much higher—than alternative organizational forms, which may present another important boundary condition. Low growth strategies may be better than high growth ones for the organization if strong identity leads to high adjustment costs.

The future of GitHub

In general, I have to believe that organizational adaptation—although costly and time-consuming—is good for GitHub, because without it, the organization is likely misaligned to the new problems it needs to solve, given its desire to grow. As Nickerson, Zenger, and Boumgarden (2012) have argued, organizational adaptation may be needed even if demand and the selection environment remain constant. Organizations often vacillate between centralized and decentralized modes as leadership struggles with opposing organizational logics like innovation and accountability and the pathologies of one mode accumulate and give rise to the need for an alternative organizational mode.

Whether from external pressure or internal ones, I therefore anticipate future organizational change. Every organizational form has its benefits and pathologies. Benefits often come quickly and pathologies take a while to accrue. When the latter become too great the leader will seek out another organization form to tackle the accrued issues. When? I cannot predict for sure, but I had look for 2 to 4 years between GitHub 2.0 and 3.0.

Epilogue

I believe that in our attempt to discover enduring principles about how organizations influence and shape behavior, we as academics have much more to learn about organizations, particularly from a governance perspective. In the discussion of GitHub 1.0, we did not ask questions comparatively, which makes it difficult to understand the tradeoffs between benefits, pathologies, and boundary conditions of a single organizational form. Comparative analysis is essential to understanding boundary conditions.

In general, our field has not done a great job of providing a comparative analysis of the benefits and pathologies of alternative organizational forms especially when they appear to be innovations. Instead, the field tends to suffer from boosterism where organizational innovation is concerned. We are probably just a little too quick to jump on the bandwagon when we see a supposedly new and novel form and say "Hey, this is awesome and solves all sorts of problems so it should be used everywhere."

My preference is to start by taking a hard look at a new organizational situation and assessing the extent to which our full range of theories may apply before we call for new theories to explain what is going on. Refinements of existing theory often provide parsimonious explanations, which enhances the accumulation of knowledge. Blanket statements like "We have got to get rid of all the old theories and start anew" are fundamentally problematic. I believe that opportunities do exist to develop new theories that more parsimoniously generate existing predictions with less apparatus as well as identify new phenomena and new theories to explain those phenomena. Nonetheless, we should hark back to Oliver Williamson's advice to engage in a slow, molecular, and definitive comparative analysis before we leap to the recommendation that "we need a whole new theory."

The GitHub story: three explanations

Phanish Puranam

In a Rashomon-like fashion, I want to offer three different accounts of what might have happened at GitHub. Of course, this plurality of accounts partly reflects our lack of access to an insider's narrative, but since the objective of the Zoo is to stimulate discussion rather than anoint a particular answer as the "truth," I see this as a fruitful exercise.

The first version is what we may call the story of "the Inevitability of Authority." Any organization to exist must have solved four fundamental problems: task division, task allocation, coordination, and motivation (Puranam et al. 2014). Authority plays a very important role in solving each of these four problems in organizations of even modest size. A founder/boss decides what tasks need to be done, who does them, how people are compensated for doing them, and how different people performing different tasks coordinate their actions (and steps to resolve issues around these when necessary).

Why? Because the alternative, self-organization based on peer to peer consensus, is not usually scalable. Human agents have opinions (and interests). As n, the number of agents in a system, increases, the possibility of disagreements increases quadratically as a function of $(n)(n - 1)/2$. Except in some atypical situations, such as when task structures are highly decomposable, or norms for consensus highly evolved, this poses a constraint on scale. In contrast, authority allows us to scale by making it possible for large numbers of people to act as an organization with agreed solutions to the fundamental problems of organizing and without succumbing to the quadratic explosion.

How? Not necessarily because authority is correlated with wisdom, but because it acts as a stopping rule that arrives at a decision faster than consensus and as a source of stability that guards against excessive local adaptation. It need not always lead to better decisions, but the advantages of speed and coordination are only magnified as *n* increases. This account suggests that while all startups begin by looking like GitHub1.0 during the "search" phase of their lives, eventually, the scale and the need to implement a chosen solution in a stable manner dictates that they end up looking like GitHub 2.0—with a well-established hierarchy of authority.

The second version is a story about "Structure as Symbol." Early on, when GitHub was still growing and developing its product, being able to attract employees who understood and were immersed in the ethos of the customers mattered. GitHub the organization took on and signaled many of the same attributes as GitHub the product. As GitHub grew to the point of needing significant external investment, and gaining large enterprise customers, it now had to appear like a legitimate member of the category—"business." That is what these stakeholders expected to see, and that is what GitHub became for them. It therefore manifested the rational ritual of a conventional hierarchy (Meyer and Rowan 1977). This helped obtain legitimacy in the eyes of key stakeholders and perhaps also functioned as a mechanism to exit those members who no longer fit the requirements of the organization. In this account, it is possible that GitHub was never quite as boss-free as it first appeared nor has it become as conventionally hierarchic as it now appears.

The third version is (obviously) a hybrid of the first two. Perhaps, the underlying organizational problem has changed through scale and technology maturity, as have the pressures to conform to expectations of external stakeholders. There are also interesting possible interactions between the two; the ethos of GitHub the product may well have been a useful framing device to coordinate the expectations and efforts of employees working in GitHub the organization when the product was first being developed; now, the ethos of GitHub the corporation provides the framing to help employees deliver reliability and robustness of solution. Goal frames, as Lindenberg (2013) has taught us, can complement and compensate for the limitations of structure.

Whichever of these accounts is true, it is clear that the story of GitHub 1.0 vs GitHub 2.0 offers a very useful window for us to think about the triplet of roles that organization design can play: *structuring*, the interactions of participants towards the attainment of an organizational goal; *sorting*, in (and out) the appropriate participants; and *framing*, the collective expectations of the participants within the system.

Dynamic design at GitHub

Todd Zenger

GitHub is a fascinating case study in organizational design and its dynamics, revealing the virtues of a boss-less enterprise, its inherent limitations, and the potential benefits of dynamic change. While GitHub composed a highly successful $2 billion enterprise featuring self-selected teams, remotely employed workers, and no formal hierarchy or titles, it more recently imposed middle and senior-level managers, reigned in many of those working remotely, and introduced more formal policies and procedures. The case illuminates several important principles of organization design—principles that reflect both the allure and implications of decentralization or autonomy and the limitations that prompt the frequent need to abandon or retrench from them. These principles include the following.

Individuals sort into organization designs they prefer

Many individuals find autonomy both attractive and motivating (Hamilton 2000; Benz and Frey 2008). Autonomy encourages initiative that inspires creative, value composing actions. For GitHub, the outcome of granting extensive autonomy was superb—a design that attracted high quality, highly motivated employees who in turn generated market value of $6 million per employee. The real sorting power of an organization design, however, becomes most evident when the design is dramatically changed. With the imposition of managers including the transformation of former peers into bosses, and new constraints on remote work, GitHub risks experiencing increased turnover at all levels, including the loss of many of those they wished most to keep. Individuals who sorted in because of the unique design may sort themselves out with the change.

Information technology investment facilitates decentralized designs by providing levels of coordination that would otherwise require greater centralization

Much has been written about the role of information technology in shaping the design of organizations (George and King 1991; Malone et al. 1987). A common argument and finding is that investments in information technology enable decentralization and are more effective in its presence (Brynjolfsson and Hitt 1998; Zenger and Hesterly 1997), including high-performance work practices that feature the use of highly autonomous teams (Hitt and Brynjolfsson 1997). A common mechanism is that information technology substitutes for the type of coordination and control that would otherwise demand centralization. Not surprisingly, the enablers of GitHub's boss-less design appear rooted in information technology. At GitHub, among other IT investments, "a sophisticated set of online chat rooms and chat bots … help coordinate activity" (Lynley 2015), enabling a level of coordination that might otherwise necessitate more centralized control.

Many organizational settings demand both the organizational outcomes common to centralization and the outcomes common to decentralization. Yet, there are tradeoffs in attempting to design for both

In many settings, high performance demands both the initiative and effort that accompany decentralization and the extensive coordination and consistency that accompany centralization. There is, of course, a long history articulating this complementarity and the tension in attempts to simultaneously generate both (Lawrence and Lorsch 1967). Unfortunately, there are inherent design tradeoffs to confront. Robert Gibbons captures well their essence in the simple phrase, "the cost of control is the loss of initiative" (Gibbons 2005: 206). Design efforts to impose control necessarily impede initiative. At the same time, design efforts to elevate initiative impede control. GitHub's experience provides a classic illustration of this tradeoff. Their boss-less design generated tremendous employee initiative but limited coordination and control. In response to a need for greater control, GitHub imposed a measure of centralized control, but experienced a predictable decline in initiative, as well as a decline in attractiveness for employees. This tension between control and initiative underlies the exploration-exploitation tradeoff (March 1991) and the essence of the ambidexterity literature (Raisch et al. 2009; Boumgarden et al. 2012). It was thus an inability to enjoy sufficient control with the boss-less design that prompted GitHub's need to rearchitect the organization.

In organizational settings where both the control that accompanies centralization and the initiative that accompanies decentralization are desired, dynamic design change is often both efficient and likely

In the late 2015 to early 2016, GitHub appears in the middle of a shift from an organization design featuring very limited control with high decentralization to a design featuring greater control and centralization. Of course, in its redesign efforts, GitHub is doing its very best to limit any loss of initiative that accompanies this shift towards greater control. GitHub recognizes the delicate or fragile nature of efforts to maintain both. Their belief is that they can keep the bureaucracy that accompanies greater centralization in check if they "stay vigilant, and let it grow organically and not make it too deliberate" (Miller 2015). As one executive commented, "It's about diligence in making sure things that do change are for good reasons and not erosion or laziness around the culture" (Miller 2015). Unfortunately, all too often, organizations discover that while autonomy and control are complements in performance, they are substitutes in production (Boumgarden et al. 2012). Achieving an abundance of both in equilibrium is desirable, but difficult. Organizations frequently discover that elevated performance is achieved through the dynamics inherent in a sequence of design choices rather than constant refinement of an otherwise static design. In other words, organization designers often discover their inability to engineer their way to a stable design and instead discover that enhanced performance is achieved through ongoing design dynamics.

This pattern is commonplace. For instance, pressures for greater exploitation of their technology prompt Google's recent push to reign in the autonomy inherent in the "20% time" rule—the policy that granted employees considerable autonomy 1 day a week to pursue projects they deemed of value. Facing similar pressure, 3 M reigned in their 15% free time rule in 2002. Nothing prevents either firm from pushing autonomy once again, and some evidence suggests that 3 M has done exactly this, but dynamic shifts are likely efficient in firms that demand both. Moreover, it is arguably the success of the push for autonomy in generating exploration, not its failure that prompts the desire to reduce it. After all, if exploration and exploitation are complements, then success in elevating exploration increases the returns to elevating exploitation and vice versa.

Looking forward

In the wake of ongoing advances in the electronic means available for control, coordination, and measurement, organizational innovations like GitHub that explore extreme decentralization are likely to continue to emerge. However, at the same time, organizational designers continue to confront tradeoffs between designs that afford control and those that prompt initiative. Unless performance environments demand only initiative or control, enhanced performance will likely continue to emerge not merely through efficient design but rather, as evident at GitHub, through their dynamics in designing.

Conclusions

Dorthe Døjbak Håkonsson–Maciej Workiewicz

The guiding principle behind the organizational zoo series has been the belief that we can learn something about the general by studying the particular (Puranam and Håkonsson 2015). In that spirit, we took a deep dive approach to the case of GitHub and its transition from a boss-less into a more traditional hierarchy, hoping to illustrate the forces operating in all organizations that compel them to adopt certain organizational design solutions. We are very thankful to our panel of commentators for joining us in this exercise and helping us to distil the lessons from this intriguing case.

GitHub was non-hierarchical but then transitioned into a hierarchical organization. This led many of our commentators into fruitful discussions of the conditions that favor boss-less organizations and the situations where hierarchy is called for.

Nickerson and Burton both mention the role of the selection environment in permitting the boss-less form at GitHub in the early years. As Nickerson puts it: “In the absence of selection pressures, practically any organizational form might perform well.” In other words, in the early years, GitHub could afford to have a less efficient form: be it hierarchy or boss-less. The growth of the firm, increase in the product complexity, and customer demands changed that.

Both Nickerson and Burton also bring back the idea of nearly decomposable systems (Simon 1962) and low task interdependency (Thompson 1967) to discuss GitHub’s earlier limited need for active coordination among several parts of the organization. Burton writes that when projects are small and independent, a boss-less form with its simple rules can take care of the negligent amount of information processing that is required.

With these conditions changed, GitHub resolved to hierarchy, leading our commentators to discuss the role of hierarchy, and the situations that call for it.

Burton and Puranam both discuss how hierarchy plays an important role in dealing with exceptions as well as in solving conflicts among organizational members. Without hierarchy, members either use simple rules or bargain to resolve conflicts, tasks that, like in the GitHub case, become increasingly cumbersome and time consuming as the number of employees increases. As Burton argues: “When operating rules are less than perfect and things go wrong, this is exactly what the hierarchy is to do; interfere when things go wrong”. Managers thus perform a useful function to break impasses and resolve conflicts. With increasing interdependencies and organization size, this benefit of hierarchy became more important and was most likely one of the key drivers of GitHub's transformation.

On the other hand, Zenger also cautions that the same forces that kept the boss-less form in the early years may act against the new hierarchy. Todd Zenger and Jackson Nickerson both point to one of these factors being the GitHub employees themselves, who were attracted to the company and its promise of autonomy and equality. Employees self-selected into and out of GitHub, thus reinforcing the ethos of bosslessness. Here, Puranam's comment on "Structures as Symbols" poses a whole new set of questions about the role of structures in organizing vs symbolic signaling of organizational values.

Overall, our commentators largely seem to agree that GitHub will need to continue experimenting. Zenger discusses how enhanced performance can be achieved through design dynamics: many settings require both control and autonomy, wherefore ongoing design change leads to enhanced performance more than efficient designs. This echoes Nickerson's remarks about the importance of prototyping and experimentation with the form, rather than static designs. In a sense, GitHub is exploring the space of suitable organization designs.

More generally, our commentators seem to support a long-standing premise that when evaluating the efficacy of a particular organizational design, the designer must first understand the primary goals and demands of the organization, i.e., its operations, innovation processes, selection environment, legitimation, etc. Only then is it possible to understand whether the design is appropriate. For example, in the early stages of industry evolution, when GitHub was just beginning to attract its first customers, its boss-less form allowed it to create more value. However, in the later stages, hierarchy helped the company to capture a greater share of the generated value.

The history of GitHub is still being written, and we cannot definitely say what will happen. As the common saying goes, making predictions is hard, especially about the future. Thus, we picked an easier task for our commentators and asked them to predict the past instead. We hope that the resulting discussion about the virtues and perils of replacing the boss-less form with a hierarchy will stimulate new questions and studies for students of organization design.

Endnotes

[1] The company raised 100 million dollars in July 2012 from Andreessen Horowitz.

[2] The article is based on secondary and publically available sources. We did approach the company for commentary, but have not received it at the time of going to press.

[3] https://github.com/holman/ama/issues/135

[4] https://zachholman.com/posts/how-github-works/

[5] https://www.fastcompany.com/3020181/inside-githubs-super-lean-management-strategy-and-how-it-drives-innovation

[6] https://www.fastcompany.com/3020181/inside-githubs-super-lean-management-strategy-and-how-it-drives-innovation

[7] https://github.com/blog/1269

[8] https://www.infoworld.com/article/2608907/application-development/github-s-new-ceo–we-re-serious-about-the-enterprise.html

[9] https://techcrunch.com/2015/06/04/github-expands-to-japan-its-first-office-outside-the-u-s/

[10] https://www.fastcompany.com/3020181/inside-githubs-super-lean-management-strategy-and-how-it-drives-innovation

Acknowledgements
Many thanks to Teppo Felin, Karim Lakhani, and the participants of the "Organizational Safari" session at the Academy of Management conference in Vancouver.

Funding
No funding was used during the development of this manuscript.

Availability of data and materials
This is a conceptual article based on publicly available information and articles.

Authors' contributions
Contributions of each author are stated for each subsection of the article and are as follows: DDH and MW carried out the "Introduction" section. RMB carried out the "Is hierarchy necessary?" section. JN carried out the "GitHub and the "boss-less" organization: adjustment costs, selection, and vacillation" section. PP carried out the "The GitHub story: three explanations" section. TZ carried out the "Dynamic design at GitHub" section. DDH and MW carried out the "Closing thoughts and open questions" section. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Fuqua School of Business, Duke University, 100 Fuqua Drive, Box 90120, Durham, NC 27708-0120, USA. [2]Interdisciplinary Center for Organizational Architecture, Aarhus University, Fuglesangs Allé 20, Aarhus V DK-8210, Denmark. [3]Olin Business School, Washington University in St. Louis, Snow Way, 1 Brookings Drive, St. Louis, MO 63130, USA. [4]INSEAD, 1 Ayer Rajah Avenue, Singapore 138676, Singapore. [5]ESSEC Business School, 3 Ave Bernard Hirsch, 95000 Cergy, France. [6]University of Utah, 1655 Campus Center Dr, Salt Lake City, UT 84112, USA.

Received: 15 July 2017 Accepted: 31 August 2017
Published online: 07 November 2017

References

Baligh HH (2006) Organization structures: theory and design, analysis and prescription. Springer, New York
Benz M, Frey BS (2008) Being independent is a great thing: subjective evaluations of self-employment and hierarchy. Economica 75(298):362–383
Boumgarden P, Nickerson J, Zenger TR (2012) Sailing into the wind: exploring relationships among ambidexterity, vacillation, and organizational performance. Strateg Manag J 33(6):587–610
Brynjolfsson, E, Hitt, LM (1998) Information technology and organizational design: evidence from micro data. MIT Working Paper
Burton RM, Obel B, Håkonsson DD (2015) Organizational design: a step-by-step approach, 3rd edn. Cambridge University Press, Cambridge
Foss NJ, Weber L (2015) Moving opportunism to the back seat: bounded rationality, costly conflict, and hierarchical forms. Acad Manag 41(1):61–79
Furr, NJ, Nickerson, JA, Wuebker, R (2016) A theory of entrepreneuring, working paper Olin Business School
Galbraith JR (1974) Organization design: an information processing view. Interfaces 4:28–36
George JF, King JL (1991) Examining the computing and centralization debate. Commun ACM 34(7):63–72
Gibbons R (2005) Four formal(izable) theories of the firm? J Econ Behav Organ 58:200–245
Haakonsson, DD, Workiewicz, M, Puranam, P, Felin, T, Lakhani, KR, Nickerson, JA, O'Mahony, S, Zenger, T (2015) The organization Safari: what can we learn about the universal problems of organizing from GitHub?, Academy of Management Proceedings
Hamilton BH (2000) Does entrepreneurship pay? An empirical analysis of the returns to self-employment. J Political Econom 108(3):604–631
Hitt L, Brynjolfsson E (1997) Information technology and internal firm organization: an exploratory analysis. J Manag Inf Syst 14(2):81–101
Lawrence PR, Lorsch JW (1967) Differentiation and integration in complex organizations. Adminstrative Science Quarterly 12(1):1–47
Lindenberg S (2013) Cognition and governance: why incentives have to take the back seat. In: Grandori A (ed) Handbook of economic organization: integrating economic and organization theory. Edward Elgar, Cheltenham, UK
Lynley, M (2015) GitHub CFO Vlado Herman is no longer at the company. TechCrunch 24, 2015
Macher JT (2006) Technological development and the boundaries of the firm: a knowledge-based examination in semiconductor manufacturing. Manag Sci 52(6):826–843
Malone TW, Yates J, Benjamin RI (1987) Electronic markets and electronic hierarchies. Commun ACM 30(6):484–497
March J (1991) Exploration and exploitation in organizational learning. Organ Sci 2:71–87
Meyer JW, Rowan B (1977) Institutionalized organizations: formal structure as myth and ceremony. Am J Sociol 83(2):340–363
Miller R (2015) At GitHub you don't need no stinkin' office, but there is a nice one if you do. TechCrunch 14:2015

Nickerson JA, Zenger TR (2004) A knowledge-based theory of the firm—the problem-solving perspective. Organ Sci 15(6):617–632

Puranam P, Alexy O, Reitzig M (2014) What's 'new' about new forms of organizing? Acad Manag Rev 39(2):162–180

Puranam P, Håkonsson DD (2015) Valve's way. J Organ Design 4(2):2–4

Puranam P, Raveendran M, Knudsen T (2012) Organization design: the epistemic interdependence perspective. Acad Manag Rev 37:419–440

Raisch S, Birkinshaw J, Probst G, Tushman ML (2009) Organizational ambidexterity. Organ Sci 20(4):685–695

Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106(6):467–482

Simon HA (1973) The structure of ill structured problems. Artif Intell 4(3–4):181–201

Thompson JD (1967) Organizations in Action. McGraw-Hill, New York.

Weber M (1948) Bureaucracy Gerth HH & Mills CW from Max Weber: essays in sociology. Routledge, New York, pp 196–244

Williamson O (1996) Mechanisms of governance. Oxford University Press, New York

Zenger TR, Hesterly WS (1997) The disaggregation of corporations: selective intervention, high-powered incentives, and molecular units. Organ Sci 8(3):209–222