Volume Title: The Rate and Direction of Inventive Activity Revisited

Volume Author/Editor: Josh Lerner and Scott Stern, editors

Volume Publisher: University of Chicago Press

Volume ISBN: 0-226-47303-1; 978-0-226-47303-1 (cloth)

Volume URL: http://www.nber.org/books/lern11-1

Conference Date: September 30 - October 2, 2010

Publication Date: March 2012

Chapter Title: The Confederacy of Heterogeneous Software Organizations and Heterogeneous Developers: Field Experimental Evidence on Sorting and Worker Effort

Chapter Authors: Kevin J. Boudreau, Karim R. Lakhani

Chapter URL: http://www.nber.org/chapters/c12366

Chapter pages in book: (p. 483 - 502)

10

The Confederacy of Heterogeneous Software Organizations and Heterogeneous Developers
Field Experimental Evidence on Sorting and Worker Effort

Kevin J. Boudreau and Karim R. Lakhani

10.1 Introduction

Ubiquitous yet invisible, software plays an integral role in the global economy. It is essential for the effective functioning of most modern organizations, critical to the advancement of knowledge in many fields, and often indispensible to many individuals' daily activities. The economic footprint of software is quite large. In 2007, in the United States, more than 110,000 firms engaged in the production and sale of packaged and custom-developed software and related information technology (IT) services. These firms generated in excess of $300 billion dollars in direct revenue (National Science Foundation 2010), making this one of the largest US industries. Purchased software is complemented by programs created within organizations that use it as an input for conducting business activities (Mowery 1996). The

Kevin J. Boudreau is assistant professor in the Strategy and Entrepreneurship Department at the London Business School. Karim R. Lakhani is assistant professor of business administration at Harvard Business School. Boudreau and Lakhani colead the NASA Tournament Lab at the Institute of Quantitative Social Science at Harvard University.

We are grateful to members of NASA's Space Life Sciences Directorate, including Jeff Davis, Elizabeth Richard, Jennifer Fogarty, and Bara Reyna, for assistance in identifying an appropriate computational engineering challenge. The TopCoder team, including Jack Hughes, Rob Hughes, Mike Lydon, Ira Heffan, Jessie Ford, and Lars Backstrom, provided invaluable assistance in carrying out all aspects of the experiment. This research particularly benefitted from thoughtful comments by Kenneth Arrow, Pierre Azoulay, Iain Cockburn, Peter Coles, Daniel Elfenbein, Silke Januszewski Forbes, Shane Greenstein, Nicola Lacetera, Mara Lederman, Joshua Lerner, Muriel Niederle, Gary Pisano, Al Roth, Sandra Slaughter, Scott Stern, Catherine Tucker, Eric von Hippel, Heidi Williams, and D. J. Wu. Eric Lonstein provided outstanding project management and research assistance. All errors are our own. Kevin Boudreau would like to acknowledge the financial support of the LBS M-Lab. Karim Lakhani would like to acknowledge the generous support of the HBS Division of Research and Faculty Development. A Google Faculty Research Grant supported both authors.

extent of internal software production and investment is considerable, with most firms typically spending 50 percent more for new, internally developed software than for software obtained through external vendors (Steinmueller 1996). More recently, open source software communities have emerged as viable creators of large-scale "free" software (Lerner and Tirole 2002). In the United States alone, more than three million individuals work as software developers (King et al. 2010), the majority employed by establishments that sell neither software nor software-related services (Steinmueller 1996).[1]

A striking feature of this industry (although perhaps not limited to software) is the wide variety of types of organizations in which software is produced, a veritable patchwork or confederacy of heterogeneous organizations. Software is developed by entities as diverse as small entrepreneurial firms, departments in large, multinational organizations, universities, outsourcing consultancies, collaborative endeavors like open source software communities, and the proverbial "garage" firms. Dissimilarities in these settings can extend beyond simple work rules, and relate to profound differences in institutional character. Compare, for example, the "software factories" in Japan, the "scientific" approach utilized by European electronics and technology champions, the ordered, engineering method pioneered by the US military and Software Engineering Institute, and the "slightly out of control" bootstrapped development practiced by Silicon Valley firms (Cusumano 2004). Within these different kinds of organizations, the work itself might be organized according to wildly divergent procedures (Cusumano et al. 2003). A given project might follow a "waterfall" development process that utilizes military-like hierarchical command and control structures in one department. It might, alternatively, employ small feature-teams working on delineated functions. Or it might utilize paired "agile" programming arrangements, or involve internal developers working closely with an external open source community. More recently organizations are using external innovation contests to develop the software (Boudreau and Lakhani 2009). Software-developing organizations have historically continually changed and tinkered with their development practices in search of the "silver bullet" without ever arriving at a clear resolution as to the single best approach (Brooks 1975).[2]

At least as striking as the organizational heterogeneity is the heterogeneity of workers, particularly their motivations and behavioral orientations. These issues have attracted considerable research attention on account of

1. DataMonitor, a professional market research firm, estimates global 2009 revenues of software and related services firms to be $2.3 trillion (DataMonitor Report 0199-2139), and International Data Corporation (IDC) projects the direct global software developer population to exceed 17 million individuals by 2011 (IDC Report 1517514).

2. For example, Microsoft's various changes in development process are well-chronicled by Cusumano and colleagues (Cusumano 1991; Cusumano and Selby 1995; Cusumano and Yoffie 1998) and Sinofsky and Iansiti (2010).

the importance and difficulty of motivating developers (Beecham et al. 2008; Sharp et al. 2009), resulting in a stream of work that includes more than 500 papers (Sharp et al. 2009). This large body of work from the 1950s to today identifies a range of motivators including the sheer joy of building and inventing and "solving puzzles," contributing to society through useful outputs, the continuous challenge of learning new techniques and approaches, and opportunities for growth, achievement, and career recognition (e.g., Brooks 1975; Bartol and Martin 1982).[3] Consistent across this line of research is the notion that the work is, itself, a reward, creating an overlap between the costs and benefits of software development (Weinberg 1971; Schneiderman 1980; Lakhani and Wolf 2005). As a group, software developers have tended to identify more with the profession and occupational community than with the organizations in which they toil (Couger and Zawacki 1980), and their behaviors are also swayed by norms in the profession. Crucially, Beecham et al. (2008) note that this long list of motivators should be understood as describing population averages, with individual software developers in fact influenced by complex and distinct *heterogeneous* sources of motivation. The literature also documents considerable heterogeneity in preferred social interactions during the course of software production. Although, in relation to other professions, software developers have been found to have the least need for social interaction both on and off the job (Couger and Zawacki 1980), other studies have reported that interdependent team structures improved productivity at the individual level and were better suited to tackling more complex tasks (Schneiderman 1980; Couger and Zawacki 1980).

Any number of reasons might explain the confederacy of different institutions devoted to software development. Here we speculate that one possible reason is that the heterogeneity of organizations may be closely tied to the heterogeneity of workers. We conjecture that the wide range of motivations (and concomitant social, psychological, and behavioral orientations) of workers is likely to translate to varying preferences for working in different types of organizations; that is, to an "institutional preference." In very preliminary steps toward investigating a link between organizational heterogeneity and worker heterogeneity, we report here results of a field experiment in which we test whether there might be an efficiency effect of sorting workers into institutional regimes of their preference, and particularly whether sorted workers experience higher motivation, as evidenced by their choice of exerted effort.

In our experiment, more than 1,000 workers were assigned, in groups of twenty, to virtual online "rooms" to solve the same problem. Inside the rooms, participants were organized either in team-"cooperative" or autonomous-

3. Beecham et al.'s (2008) review of the post-1980 literature on the motivations of software developers identified twenty-one sources of motivation.

"competitive" regimes. In the competitive regime, individuals competed against all others in the room; in the cooperative regime, individuals were assigned to one of four five-person teams that competed against each other. These two regimes hardly replicate the full variety of regimes we observe in the confederacy of software organizations. But they do exhibit a range of starkly opposing features that accord with different work approaches in software development; that is, software developers either work on their own or in teams. We divided participants into "sorted" and "unsorted" groups with identical skills distributions. For the sorted group, we elicited their preferences and assigned them to the regime they preferred. The unsorted (control) group was assigned without regard to their preferences, indeed they were not even asked about their institutional preferences. This group therefore constituted the population average distribution of preferences (including both those who liked and disliked the regime to which they were assigned). We were also able to compare the effects of sorting on the basis of institutional preference to the effect of formal incentives, as some groups of twenty competed for $1,000 in prizes, other groups for no prize.

We found that allocating individuals to their preferred regimes had a significant impact on choice of effort level, particularly in the autonomous competitive regime, in which sorted participants worked, on average, 14.92 hours compared to 6.60 hours, on average, for the unsorted participants. The effect was also positive and significant in the team regime, in which the sorted group worked, on average, 11.57 hours compared to 8.97 hours, on average, for the unsorted participants. We devote the bulk of the analysis to confirming the robustness of the result and investigating the nature of this sorting effect.

The rest of the chapter is organized as follows. Section 10.2 outlines the basic approach to running the sorting experiment in a way that enabled us to compare the sorted and unsorted groups on the basis of institutional preferences, with the important feature that they possess identical skills distributions. In section 10.3, we present the sample and variables. Section 10.4 reports our results, comparing mean outcomes across the sorted and unsorted groups. Concluding remarks are presented in section 10.5.

10.2 Experimental Design

In our experiment, we consider the possibility that the extraordinary heterogeneity in organizations and workers in the software industry are somehow linked. Our central goal here is to estimate the extent to which assigning individuals to work within the regime they prefer influences how hard they work. The essence of our approach is quite simple. We define two work regimes: a "cooperative" and a "competitive" regime. We assign half the participants to work within the regime they prefer and the other half without regard to their preferences. Thus, we effectively compare the effort (and underlying motivations) of a "sorted" group, in which 100 percent of participants prefer the regime to which they are assigned, to that of an "unsorted" group that exhibits the population average distribution of preferences.

10.2.1 Field Experiment Context

Given our emphasis on measuring the size of the effect in relation to how different *types* of workers behave under different circumstances, a field setting has the clear advantage of providing more meaningful estimates than a lab setting. Nonetheless, to estimate sorting effects requires an especially controlled environment. We conducted the experiment on the TopCoder open software innovation platform.[4] TopCoder is an online, two-sided platform that produces software for clients via online contests among members of its base of more than 300,000 individuals. This provided a field context with real, elite software developers that afforded an unusual ability to perform manipulations and observe relevant microeconomic variables. Over the ten-day period of the experiment, participants developed computational algorithms to optimize the Space Flight Medical Kit for NASA's Integrated Medical Model (IMM) team in the Space Life Sciences Directorate at Johnson Space Center. TopCoder provided substantial assistance in altering the platform to enable us to run a multitude of treatments concurrently and in isolation, with setting up the NASA problem on the platform, and with running the experiment.

The solution to the real, highly challenging computational-engineering problem of developing a robust software algorithm to recommend the ideal components of the space medical kit included in each space mission was to be used by NASA. The solution had to take into account that mass and volume are restricted in space flight, and that the resources in the kit needed to be sufficient to accommodate both expected and unexpected medical contingencies encountered while in space, lest the mission have to be aborted. The content of the kit also had to be attuned to the characteristics of the space flight and crew. The challenge was thus to develop an algorithm that addressed mission characteristics that traded off mass and volume against sufficient resources to minimize the likelihood of medical evacuation. The problem, being relatively focused, was expected to be solved as an integral project capable of being divided into a set of subroutines and call programs. These sorts of projects might be solved by open source or corporate development teams composed of as many as five people (Carmel 1999) and are also routinely tackled by participants in TopCoder tournaments (Boudreau, Lacetera, and Lakhani 2011).

4. Boudreau, Lacetera, and Lakhani (2011), in using the TopCoder context to analyze the impact of increasing competition on performance in software contests, provide considerable detail on the TopCoder setting.

10.2.2 An Assignment Procedure for Dividing Participants into Sorted and Unsorted Groups with Identical Skills Distributions

The potential correlation of institutional preferences with skill poses a special challenge to our experiment. In such a case, differences in behavior would reflect skills differences as well as any differences between the sorted and unsorted groups per se. So as to assure that we do not conflate skills differences with the effect of preferences per se, we devise an assignment procedure that exploits both matching and randomization, as summarized in figure 10.1. The goal of our approach is to create groups, or "virtual rooms," of twenty participants drawn from the same skills distribution (and

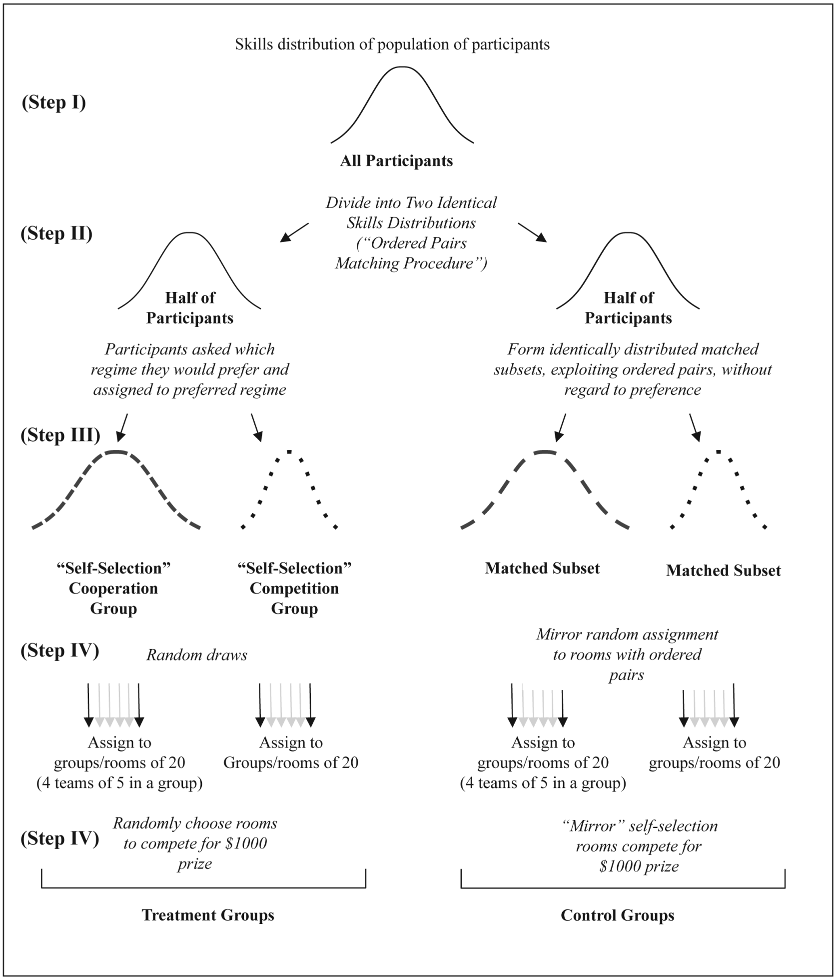

Fig. 10.1 Overview of experimental assignment

equivalent unobserved characteristics), but with different tastes for the two regimes. The construction of the sorted and unsorted groups begins by dividing the participants into two groups with identical skills distributions. This is accomplished by ordering all participants in the population from top to bottom according to their TopCoder skills rating.[5] Essentially, we created a rank order of all participants. We then divided this rank order into ordered pairs (top two highest skills, third and fourth highest skills, etc.) and randomly allocate one member of each to the sorted and the other to the unsorted group.

We then asked just the participants in the sorted group which regime they preferred. This was done in private bilateral communications between the TopCoder platform and individual participants, each of whom was asked: "Might you be interested in joining a team to compete against other teams?" Relative preference for the competitive or cooperative regime was to be indicated on a 5-point Likert scale.[6] The resulting subgroups were assigned to the cooperative and competitive regimes.

It is important to note that the groups that prefer the competitive and cooperative regimes will not have the same skills distributions if there is any correlation between skill and preference. By assigning ordered pairs of the unsorted group to the same regime as their sorted pairs, we assure that sorted and unsorted groups in both cooperative and competitive regimes have identical skills distributions. We thus constructed groups identical in skills distributions that differed systematically in terms of their preferences for regimes. The sorted group was uniformly orientated toward the regime to which it was assigned; the random-assignment group had population average preferences, with some individuals preferring, and others not, the regime in question.

The sorted groups of cooperative and competitive participants were then divided into groups of twenty individuals who competed in virtual, web-based "rooms." Cooperative rooms were formed of four teams composed (also randomly) of five individuals. We "mirror" this random assignment in the unsorted group, assigning ordered pairs to comparable groups.

5. The TopCoder skill rating is based on historical performance of the coders on the platform. It is derived from the chess grandmaster evaluation system "Elo." Boudreau, Lacetera, and Lakhani (2011) provide further detail on how it is derived.

6. Participants were first asked their preference between the regimes, then given the following options: (1) I DEFINITELY would prefer to join a team; (2) I think I MIGHT prefer to join a team; (3) I am indifferent or I am not sure; (4) I think I MIGHT prefer to compete on my own; and (5) I DEFINITELY would prefer to compete on my own. They were then provided with additional descriptive details about each of the regimes and asked the same question. We then asked them to consider the possibility that both cooperative and competitive regimes were always available on the TopCoder platform, and to indicate on a provided list of options what fraction of their time they would imagine spending in either regime. The order of responses, whether oriented toward the competitive or cooperative regime, was randomized. The second question (the one asked after clarifying the precise rules of each regime) was used as the basis for making allocation decisions.

The submitter (individual or team) of the best performing code across the entire tournament was eligible to receive a \$1,000 cash prize and VIP access to one of the few remaining NASA Space Shuttle launches. We also randomized the presence of room-level incentives in our experiment by offering \$1,000/room cash prizes to twenty-four rooms (twelve competition regime rooms and twelve cooperation regime rooms, equally split between sorted, and skills matched unsorted, groups). Thus, if a sorted participant was assigned to a room with a \$1,000 cash prize, so was this participant's ordered pair in the unsorted group. Note that the participants did not know, ex ante, if they would be competing for room-level prizes.

10.2.3 The Cooperative and Competitive Regimes

Our primary unit of analysis of the competition regime was the twenty-person group of direct competitors. The \$1,000 cash prize, if present, was divided among the top five competitors: \$500 for first place, \$200 for second place, \$125 for third place, \$100 for fourth place, and \$75 for fifth place. Individuals could see the list of the other nineteen competitors on their "head-up" display with "handle" name and color code by skill. (Clicking through on a name provided a complete history of that participant's performance on the TopCoder platform and a precise breakdown of their skill rating. Scores of existing submissions by all competitors in a room appeared alongside competitor names.)

The cooperative regime also involved twenty individuals in a virtual room with five prizes. However, in this case, the twenty participants were divided into four, five-person "teams." These individuals could communicate and share code via a private discussion board. The winning team in a room was the team with the highest scoring submission (any team member could make a submission). In the cash prize treatment, the \$1,000 was divided by an anonymous poll of the members of the winning team (after the competition, but before the winners were announced) regarding how each believed the prize should be shared, with prizes awarded based on average percentages. Each team could only observe other team members and the best submission at any given time by other teams.

10.3 Sample and Variables

It should also be emphasized, with regard to our research objective of measuring the selection effects of a sort, that the TopCoder membership hardly represents a random sample of individuals from the economy, or even from the software developer labor market. At the time of the experiment, some 15,000 TopCoder members regularly participated on the platform. Because the population in the experiment reflects a choice to voluntarily participate, the results should be interpreted as "treating on the treated," or assigning what is a nonrandom population to different treatments. Although there is considerable diversity in this group, which includes individuals from many countries and from industry as well as students and researchers, it remains a subset of the wider population of the global software developer labor market, and estimates of effects of sorted versus random assignment of workers should therefore be smaller than what might be possible were we to construct a more diverse sample from the broader labor market.

Our sample includes 1,040 observations (participants). Of the half of participants who were asked their preference (the sorted group), 34.9 percent expressed a clear preference for the cooperative regime, and 50.5 percent a clear preference for the competitive regime.[7] The remaining 15.6 percent of participants in the sorted group expressed uncertainty or indifference between the regimes. We assigned this latter group to the cooperative regime for two reasons. First, we interpreted this indifference to indicate some openness to the cooperative regime (TopCoder's usual regime is similar to the competitive regime). Second, we preferred to balance the numbers across regimes. (Dropping the indifferent observations from the analysis has a negligible effect on the results.)

Of the rooms formed, only twelve rooms (44 percent of the sample), six sorted and six unsorted, competed for cash prizes amounting to $1,000 per room.[8] Prizes were first assigned randomly across the sorted rooms. The "mirror" rooms of ordered pairs with corresponding assigned competitors were then also allocated $1,000 prizes.

10.3.1 Variables

We now discuss the meaning and construction of variables used in the analysis. Table 10.1 provides variable definitions and table 10.2 presents summary statistics.

Dependent Variables

We exploit both observational and self-reported survey measures of effort. The observational measure is the number of submissions made by each participant over the course of the zero-day experiment (*NumSubmissions*). This is a direct indication of the intensity of development, given that software testing and evaluation required that code be submitted to the platform so that its performance in relation to the test suite could be assessed and it could be assigned a score. (Participants' last submission became their final score.) Submitting code in this fashion was costless and resulted in virtually instantaneous feedback.

Our preferred main dependent variable records the total number of hours participants invested in the preparation of solutions throughout the course

7. We originally targeted half the entire group of 1,098, but did not receive responses from a small fraction of individuals.

8. We chose twelve simply because participation in the experiment exceeded expectations and we had not budgeted for more than twelve prizes for the competitive regime.

Table 10.1 Variable definitions

Variable	Definition
HoursWorked	Number of hours worked by an individual participant during the course of the experiment
NumSubmissions	Number of solutions submitted to be compiled, tested, and scored by an individual participant during the course of the experiment
SortedonPreference	Indicator switched to one for participants who were asked their preferences regarding the regimes and subsequently assigned to their preferred regime
CashPrize	Indicator switched to one for participants within a group of twenty that competed for a $1,000 cash prize
SkillRating	Measure of general problem-solving ability in algorithmic problems based on historical performance on TopCoder platform

Table 10.2 Summary statistics

Variable	Mean	Std. dev.	Min.	Max.
HoursWorked	10.6	18.7	0	190
NumSubmissions	2.56	5.63	0	42
SortedonPreference	.50	.49	0	1
CashPrize	0.44	.50	0	1
SkillRating	1,184	538	0	3,797

of the event. This self-reported estimate of the total number of hours worked (*HoursWorked*) was reported in a survey administered the day after the event closed.[9] (Participants were required to respond to this question electronically, as the experiment closed, in order to receive a NASA-TopCoder commemorative t-shirt imprinted with their name.) *HoursWorked* is our preferred variable, as it directly conveys meaning (and perhaps even some indication of value) and is easily interpreted. The results do not depend on which of the two measures of effort we use in the analysis.

Explanatory Variables

The key explanatory variable, *SortedonPreference,* indicates whether a competitor was in a sorted or random assignment group. A second explana-

9. Nearly all participants who submitted solutions responded. A research assistant who contacted 100 of the nonsubmitters who did not respond to the first survey found that each had devoted less than one hour to the project and had not made a submission. This enabled us to complete the nonrespondents by filling in zero hours as a relatively precise approximation. It became clear through interviews with nonsubmitters that they generally believed they would not receive a commemorative t-shirt whether they responded to the survey or not, accounting for the sharp difference in response rate between submitters and nonsubmitters. Worthy of note, however, is that a number of nonsubmitters whom we discovered had worked a nontrivial number of hours before choosing not to submit did respond to the survey.

tory variable, *CashPrize,* indicates that observations/individuals were associated with rooms for which there was a $1,000 cash prize. A third explanatory variable, *Competition,* is set to one to indicate the competitive regime, and zero to indicate the cooperative regime.

Our measure of general ability to solve algorithmic problems is TopCoder's own rating system, which essentially calculates a participant's ability to solve problems on the basis of past performance. We refer to this variable as *SkillRating.* We use specifically the rating calculated for what TopCoder terms "Algorithm" matches, software solutions to abstract and challenging problems akin to the problem in the experiment.[10]

Additional Variables

In robustness tests, we use two additional variables collected for those in the sorted group. The variable *LikertScale* captures the Likert scale responses of those asked their preferences. Recall that the numerical responses in this variable correspond to the following scale: (1) I DEFINITELY would prefer to join a team; (2) I think I MIGHT prefer to join a team; (3) I am indifferent or I am not sure; (4) I think I MIGHT prefer to compete on my own; and (5) I DEFINITELY would prefer to compete on my own. The variable *OrderofQuestion* captures whether the survey was designed to present all aspects of introducing regimes with the cooperative or competitive regime first.

10.4 Results

The average number of hours worked by participants during the ten-day experiment was 10.54 (standard deviation = 18.74 hours). Sorted individuals worked, on average, 13.27 hours (maximum 190 hours), unsorted individuals only 7.78 hours (maximum 120 hours). *NumSubmissions* was also higher for sorted participants, at 2.79 versus 2.20 for unsorted participants.

Table 10.3 breaks down the effects for the competitive and cooperative regimes. Average *HoursWorked* was only slightly higher in the competitive (10.82 hours) than in the cooperative (10.27 hours) regime.[11] In both regimes, *HoursWorked* was significantly higher for sorted participants, the starkest differences being in the competitive regime (14.92 hours for sorted participants versus 6.6 hours for unsorted participants, a 126 percent difference, compared to 11.57 and 8.97 hours, respectively, in the cooperative regime, a still large but considerably smaller 29 percent difference).

10. This has been found through the decade of operation of TopCoder to be a robust measure of skills, and is even commonly used in the software developer labor market when hiring (See Boudreau, Lacetera, and Lakhani [2011] on this measure). Nonetheless, to the extent that it might be imperfect, the randomization procedures (in particular, pair ordering and randomization of which party self-selects) should erase any possible systematic biases in estimates.

11. We found the differences in magnitudes to be surprisingly small and statistically insignificant, given the usual predictions of moral hazard in teams (Holmstrom 1982).

Table 10.3 **Simple cross-tabulation comparison of means**

Competitive regime						
Unsorted			Sorted			
Variable	Mean	Standard deviation	Variable	Mean	Relative to unsorted	Standard deviation
HoursWorked	6.60	13.46	*HoursWorked*	14.92	226%	24.99
NumSubmissions	1.98	5.00	*NumSubmissions*	3.77	191%	7.22
Cooperative regime						
Unsorted			Sorted			
Variable	Mean	Population std. dev.	Variable	Mean	Relative to unsorted	Population std. dev.
HoursWorked	8.97	15.70	*HoursWorked*	11.57	129%	17.61
NumSubmissions	2.44	5.53	*NumSubmissions*	1.78	73%	4.07

For *NumSubmissions,* levels were also on the order of twice as high for sorted (3.77 submissions) than for unsorted (1.98 submissions) participants. That average *NumSubmissions* was lower for sorted participants in the cooperative regime we speculate reflects greater coordination of activity across team members.[12] Given this apparent complication in using *NumSubmissions,* we take *HoursWorked* as a more direct reflection of effort exerted. (Indeed, all regression results to follow hold for *NumSubmissions,* but only for the competitive regime.) Particularities of team dynamics are beyond the scope of our analysis here.

10.4.1 Regressions

Although the earlier comparisons of means provide meaningful results, analyzing the data within a regression framework enables us to explicitly assess the experimental assumptions and better interpret results. Ordinary least squares regression results with robust standard errors are reported in table 10.4.

Assessing the Assignment Procedure

If the estimation procedure was effective and left no systematic differences across treatments, the estimates should be unchanged when we include skill controls.[13] We focus first on the results for the competitive regime. For ease of comparison, model (1) simply reiterates the two-way correlation of

12. Consistent with this interpretation, we find that sorted teams posted greater numbers of intrateam communications on the private team online bulletin board.
13. This includes differences in skill and unobservables correlated with skill.

Table 10.4 **Ordinary least squares (OLS) estimates of sorting effect**

	Competitive regime					Cooperative regime		
Explanatory variable	Model 1 Two-way correlation	Model 2 Linear skills control	Model 3 Skills-level dummies	Model 4 Ordered pair differences	Model 5 Prize control	Model 6 Two-way correlation	Model 7 Ordered pair differences	Model 8 Prize control
SortedonPreference	8.33***	8.33***	8.36***	8.71***	8.32****	2.60*	2.50*	2.48*
	(1.75)	(1.75)	(1.76)	(1.79)	(1.71)	(1.47)	(1.43)	(1.40)
CashPrize					9.14***			9.88***
					(1.85)			(1.48)
SkillRating		–1.09	–4.87		–3.60			2.01
		(1.59)	(4.30)		(4.19)			(4.22)
Skills Dummies			Yes		Yes			Yes
Constant	6.60***	8.07***				8.97***		
	(.84)	(2.28)				(.98)		
R^2	.04	.04	.05	.55	.09	.04	.55	.09

Notes: Dependent variable = *HoursWorked.* Heteroskedasticity robust standard errors reported.
***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

HoursWorked with *SortedonPreference* from the competitive regime (essentially equivalent to the earlier stratified comparison of means in table 10.3). Model (2) reestimates the *SortedonPreference* coefficient with *SkillRating* included as a control. The estimated coefficient is virtually unchanged, and the coefficient on the constant, which effectively captures mean effort without sorting, changes slightly more (from 6.60 to 8.07), but the difference is statistically insignificant. To control for possible subtle nonlinearities, model (3) adds dummies for different bands of skill level to capture possible nonlinear effects, but the estimated coefficient on *SortedonPreference* is statistically identical and virtually unchanged (8.36 versus 8.33). Model (4) provides the strongest skill control by simply comparing and calculating the difference between sorted individuals and their ordered pairs (by simply including ordered pair fixed effects). The estimated effect is again statistically unchanged (although this most stringent control only yields a slightly larger coefficient). Given the random selection of rooms to receive prizes, the introduction of *CashPrize* to the model should also not have any effect on the estimated coefficient *SortedonPreference.*[14] Each of these coefficient estimates is thus statistically identical to the simple comparison of means presented in table 10.3 (14.92 – 6.6 = 8.32 hours).

Importantly, the coefficient on *CashPrize* also provides some indication of the impact of sorting relative to that of the formal incentive instrument used in this context, the $1,000 prize. The coefficient on *CashPrize,* 9.14 hours with a standard error of 1.85 hours, is statistically indistinguishable from the effect of allowing individuals to self-select to competition for cases in which competition is their preferred regime.

An analogous set of regressions performed on the cooperative regime similarly confirms estimates of the *SortedonPreference* coefficient to be insensitive to the various controls. Model (6) reiterates the two-way correlation of *HoursWorked* with *SortedonPreference* from the cooperative regime (essentially equivalent to the earlier stratified comparison of means in table 10.3), 2.6 additional hours for individuals who sorted into the cooperative regime. Reestimating the effect on the basis of directly comparing ordered pairs (model 7) or introducing *CashPrize* and controls for different levels of skills (model 8) generates similar estimates. The estimated coefficient on *SortedonPreference* is 2.60 hours. Model (6) essentially reestimates model (4) with each of the controls, but for the cooperative regime. Including each of the controls does not significantly change the coefficient on *SortedonPreference* (2.47 hours). Again, these estimates are statistically the same as those obtained from the simple comparison of means in table 10.3

14. We must go back to a model estimated on the basis of ordered pair differences given that there is no variation in CashPrize within ordered pairs because the assignment procedure assures that if one member of an ordered pair is in a group with a prize the situation will be mirrored in the other pair.

(11.57 – 8.97 = 2.60 hours). The effect of the formal cash incentive in the cooperative regime, as estimated by the coefficient on *CashPrize* (9.88 hours), is essentially the same as in the competitive regime (and the sorted effect in the competitive regime), and considerably larger than the sorted effects in the cooperative regime.[15]

An Approach to Estimating the Magnitude of Any Hawthorne Effects

Our goal was to use revealed preference as a means of allocating individuals to the regimes for which they have an inherent preference or taste. Therefore, the earlier regressions are intended to estimate the impact of this "alignment" of an individual's preference for institutional context on choice of effort. But it might still be the case that individuals made different choices simply because they were asked their preferences at all. This is a possible Hawthorne effect of sorts that should be a concern in any sorting experiment in which subjects' preferences have been directly elicited or a direct choice has been presented.

To estimate the magnitude of any such effect of eliciting preferences (as opposed to what those preferences happen to be) is challenging in an experiment in which assignments followed revealed preferences without any variation. Our approach is essentially one of detecting Hawthorne effects by comparing the subset of sorted and unsorted participants with similar preferences. If there is a Hawthorne effect, then individuals with similar institutional preferences should behave differently in sorted and unsorted groups. Results are presented in table 10.5.

Therefore, we focus on the 15 percent of sorted participants who chose a neutral response when asked to gauge their relative preferences for regimes (i.e., "I am indifferent or I am not sure"[16]). A possible limitation to this approach is that a neutral view of the cooperative regime may, in fact, imply some level of openness to an interest in this regime (given that the competitive regime is, in fact, the usual TopCoder regime).[17] To better isolate participants whose stated preferences were more likely to have been shaped by an exogenous factor than to reflect their inherent preferences, we surveyed individuals' preferences using an instrument that varied the order, sometimes presenting the competitive regime, other times the cooperative regime, first. As presented in model (1), the ordering of the question significantly affected the statement of preferences. Reestimating the model on this 15 percent of the sample (156 observations) results in a statistically identical estimate

15. As earlier noted, this result is perhaps surprising in light of the theory of moral hazard in teams (Holmstrom 1982).

16. Recall that indifferent individuals were assigned to the cooperative regime (section 10.2.2).

17. A second possible limitation is we rely on the (unobserved) preferences of ordered pairs being effectively neutral, on average.

Table 10.5 Instrumental variable (IV) estimate of Hawthorne effect

Explanatory variable	Dependent variable = *LikertScale* Model 1	Dependent variable = *HoursWorked* Model 2
SortedonPreference		–.05
		(4.40)
CashPrize	–.10	9.43***
	(.12)	(2.79)
SkillRating	.37	3.79
	(.37)	(8.20)
Skills dummies	Yes	Yes
QuestionOrder	0.28**	
	(.12)	
R^2	.05	

Note: Heteroskedasticity robust standard errors reported.
***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

of the coefficient on *CashPrize,* but the coefficient on *SortedonPreference* goes to zero, suggesting zero Hawthorne effect.[18]

An Approach to Reweighting to Directly Compare the Different Sorted Groups

The skills distributions being, by design, the same across the sorted and random assignment groups, we should expect sorting to have generated differences in skills distributions across the competitive and cooperative groups. Figure 10.2, panel I presents the distribution of skills of participants who sorted themselves into the competitive and cooperative regimes (equivalently, their ordered pairs in the unsorted group). This was unavoidable in this sorting experiment, in which preferences were correlated with skill. Consequently, earlier estimates of the coefficients on *SortedonPreference* in the cooperative and competitive regimes should not be directly comparable if the magnitude of an individual sorted effect is somehow related to skill.

To more directly compare the magnitude of effects in the cooperative and competitive regimes, we reestimate effects, reweighting the data from the competitive regime to have the same skills distribution as that of the cooperative regime (as in figure 10.2, panel II). As reported in table 10.6, when the model is reestimated on competitive data, reweighted to share the

18. The estimated Hawthorne is also statistically insignificant without the use of the instrumental variable, with an estimated coefficient on *SortedonPreference* of 3.72 (s.e. = 2.61). This estimate, which is considerably larger than the instrumental variable (IV) estimate, remains statistically indistinguishable from zero, whereas the coefficient on *CashPrize,* strikingly, remains virtually unchanged in magnitude or significance in this model.

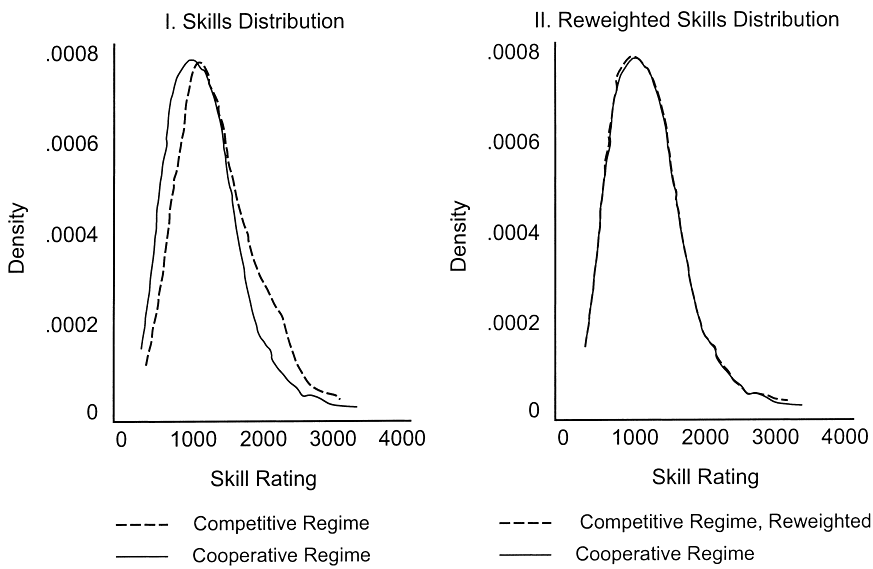

Fig. 10.2 Skills distribution in competitive and cooperative regimes

Table 10.6 Reestimated results from cooperative regime to match skills distribution of cooperative regime

Explanatory variable	Competitive regime
SortedonPreference	10.2814***
	(2.08)
CashPrize	6.7416***
	(2.20)
Skills dummies	Yes
R^2	.12

Notes: Dependent variable = *HoursWorked*. Heteroskedasticity robust standard errors reported.

***Significant at the 1 percent level.

**Significant at the 5 percent level.

*Significant at the 10 percent level.

skills distribution of the cooperative regime, the estimated coefficient on *SortedonPreference* increases from 8.32 hours to 10.28 hours. The estimated effect on *CashPrize,* by comparison, drops to 6.74.

10.5 Conclusions

Software design and development is done in very diverse organizational settings. Seemingly just as diverse and heterogeneous are the software developers who work in these organizations. This chapter takes very preliminary steps toward investigating whether there might be a link between heterogeneity of organizations and workers by assessing whether sorting software workers into their preferred regimes affected their motivations and the effort they exerted.

We devised a novel sorting experimental method that enabled us to compare a group of software developers who were sorted into a (competitive or cooperative) regime of their preference with a group of individuals who were assigned without regard to preference, assuring that both groups possessed identical distributions of raw problem-solving ability. Thus, in contrast to more conventional experimental approaches that attempt to hold the composition of groups constant while exposing them to alternative treatments, the thrust here was to hold treatments constant while allowing the composition of groups to differ in a rather precise way.

We found the effect of sorting of software developers on the basis of their preference to join the cooperative and competitive regimes in this context to be rather large. In the competitive regime, effort roughly doubled, on average. In the cooperative regime, estimates, albeit smaller, were, at a roughly 30 percent increase, still rather large. Estimates were similar across a range of specifications. We also devised a method for explicitly estimating any Hawthorne effects that may have resulted from the approach we used to elicit individuals' preferences (based on an instrumental variables estimate of a subsample of the data) and found these to be statistically indistinguishable from zero.

The present work, of course, has many limitations, and endless work remains to be done in investigating possible links between worker and organizational heterogeneity in software (and other) contexts in a competitive economy in which firms and workers match in equilibrium. With respect to the experiment conducted here, the analysis is focused on estimating mean differences rather than distributions of outcomes or associated demographic attributes of workers. Specifically, the analysis presented here emphasizes comparisons with just one type of (unsorted) control group; in considering the effect of different "types" of workers, any number of alternative and synthetic control groups might be contrived. The analysis presented here, being focused on effort, did not study effects on overall performance and productivity. There is also an indication in the results presented here that sorting may have generated subtle effects in the organization of, and patterns of collaboration in, the cooperative regime that were not further investigated here.

Our experimental results provide an opening for further investigation of how workers engaged in inventive activity might be most effectively and efficiently organized. Our work contributes to a nascent field in the economics of innovation that is utilizing microdata on scientific and technical workers and the links between incentives and creativity (Azoulay, Graff Zivin, and Manso 2011), preferences for work environments (Stern 2004), and the organization of scientific teams (Jones, Wuchty, and Uzzi 2008). As individual and team level productivity issues for creative workers become increasingly salient for organizational and national level performance (Radner 1993; Hong and Page 2001), this stream of research (and future related work) has the potential to provide relevant theoretical, empirical, and practical insights.

References

Azoulay, P., J. Graff Zivin, and G. Manso. 2011. "Incentives and Creativity: Evidence from the Academic Life Sciences." *RAND Journal of Economics* 42 (3): 527–54.

Bartol, K., and D. Martin. 1982. "Managing Information Systems Personnel: A Review of the Literature and Managerial Implications." *MIS Quarterly* 6:49–70.

Beecham, S., N. Badoo, T. Hall, and H. Robinson. 2008. "Motivation in Software Engineering: A Systematic Literature Review." *Inf. Softw. Technol.* 50 (9–10): 860–78.

Boudreau, K. J., N. Lacetera, and K. R. Lakhani. 2011. "Incentives and Problem Uncertainty in Innovation Contests: An Empirical Analysis." *Management Science* 57 (5): 843–63.

Boudreau, K. J., and K. R. Lakhani. 2009. "How to Manage Outside Innovation: Competitive Markets or Collaborative Communities?" *Sloan Management Review* 50 (4): 69–76.

Brooks, F. P. 1975. *The Mythical Man-Month: Essays on Software Engineering.* Reading: Addison-Wesley Pub. Co.

Carmel, E. 1999. *Global Software Teams: Collaborating across Borders and Time Zones.* Upper Saddle River: Prentice Hall.

Couger, J. D., and R. A. Zawacki. 1980. *Motivating and Managing Computer Personnel.* New York: Wiley.

Cusumano, M., A. MacCormack, C. F. Kemerer, and B. Crandall. 2003." Software Development Worldwide: The State of the Practice." *IEEE Softw.* 20 (6): 28–34.

Cusumano, M. A. 1991. *Japan's Software Factories: A Challenge to U.S. Management.* New York: Oxford University Press.

———. 1997. "How Microsoft Makes Large Teams Work Like Small Teams." *Sloan Management Review* 39 (1): 9–20.

———. 2004. *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad.* New York: Free Press.

Cusumano, M., and R. Selby. 1995. *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets and Manages People.* New York: Free Press.

Cusumano, M. A., and D. B. Yoffie. 1998. *Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft.* New York: Free Press.

Holmstrom, B. 1982. "Moral Hazard in Teams." *The Bell Journal of Economics* 13 (2): 324–40.

Hong, L., and S. E. Page. 2001. "Problem Solving by Heterogeneous Agents." *Journal of Economic Theory* 97 (1): 123–63.
Jones, B. F., S. Wuchty, and B. Uzzi. 2008. "Multi-University Research Teams: Shifting Impact, Geography, and Stratification in Science." *Science* 322 (5905): 1259–62.
King, M., Steven Ruggles, J. Trent Alexander, Sarah Flood, Katie Genadek, Matthew B. Schroeder, Brandon Trampe, and Rebecca Vick. 2010. *Integrated Public Use Microdata Series, Current Population Survey: Version 3.0.* University of Minnesota. http://cps.ipums.org/cps.
Lakhani, K. R., and E. von Hippel. 2003. "How Open Source Software Works: Free User to User Assistance." *Research Policy* 32 (6): 923–43.
Lakhani, K. R., and R. Wolf. 2005. "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects." In *Perspectives on Free and Open Source Software,* edited by Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani, 3–21. Cambridge, MA: MIT Press.
Lerner, J., and J. Tirole. 2002. "Some Simple Economics of Open Source." *The Journal of Industrial Economics* 50 (2): 197–234.
Mowery, D.C. 1996. *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure.* New York: Oxford University Press.
National Science Foundation. 2010. *National Patterns of R&D Resources: 2008 Data Update.* Arlington: National Science Foundation. http://www.nsf.gov/statistics/nsf10314/pdf/nsf10314.pdf.
Radner, R. 1993. "The Organization of Decentralized Information Processing." *Econometrica: Journal of the Econometric Society* 61 (5): 1109–46.
Salop, J., and S. Salop. 1976. "Self-Selection and Turnover in the Labor Market." *The Quarterly Journal of Economics* 90 (4): 619–27.
Schneiderman, B. 1980. *Software Psychology: Human Factors in Computer and Information Systems.* Boston: Little, Brown and Co.
Sharp, H., N. Badoo, S. Beecham, and T. Hall. 2009. "Models of Motivation in Software Engineering." *Information and Software* 51 (1): 219–33.
Sinofsky, S., and M. Iansiti. 2010. *One Strategy!: Organization, Planning, and Decision Making*. Hoboken, NJ: Wiley.
Steinmueller, W. E. 1996. "The U.S. Software Industry: An Analysis and Interpretive History." In *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure,* edited by David C. Mowery, 15–52. New York: Oxford University Press.
Stern, S. 2004. "Do Scientists Pay to Be Scientists?" *Management Science* 50 (6): 835–54.
Weinberg, G. M. 1971. *The Psychology of Computer Programming.* New York: Van Nostrand Reinhold.

Comment Iain M. Cockburn

The productivity of knowledge workers, particularly "high level" knowledge workers, is a first-order issue for understanding technical change, and I am

Iain M. Cockburn is professor of strategy and innovation at Boston University and a research associate of the National Bureau of Economic Research.