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

(for NBER 50th Anniversary Volume on the Rate and Direction of Inventive Activity)

Kevin J. Boudreau (London Business School) &
Karim R. Lakhani (Harvard Business School)

This version: January 4, 2011

Abstract

Software development occurs in a patchwork or "confederacy" of different types of institutions (universities, small start-ups, multinational enterprises, government agencies, etc.) utilizing varied work approaches. Here we speculate on one possible explanation for this organizational heterogeneity: it may reflect inherent heterogeneity of the software workforce, in terms of which kinds of organizations individual workers prefer to work within ("institutional preference"). We take very preliminary steps towards investigating this possibility by devising a novel 10-day field experiment to estimate the differences in behavior that are created by sorting workers into their preferred institutional regimes versus having them unsorted. The experiment involved assigning 1040 elite software developers to either a competitive or a cooperative work regime to create software for NASA's Space Life Sciences Directorate. Half of the subjects—the "sorted" group--were assigned according to their institutional preferences; the other half—the "unsorted" group--were assigned without regard to their preferences. Assignment was done in a manner that sorted and unsorted groups had identical distributions of raw problem-solving ability. We find a remarkably large effect of institutional preference-based sorting on the effort exerted. Sorting on institutional preferences roughly doubled effort within the competitive regime and increased effort by roughly half in the cooperative regime, while accounting for incentives. Our experimental approach and results indicate the importance of accounting for worker preferences in creative activities that drive the rate and direction of inventive activity in the economy.

Acknowledgements: 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, Daniel Elfenbein, Joshua Lerner, Gary Pisano, Al Roth, Scott Stern, Heidi Williams, D.J. Wu, and Catherine Tucker. Shane Greenstein deserves special recognition, as this project and line of research was triggered by comments he made at the 2009 NBER Summer Institute concerning the allocation of different types of workers across different types of organizations in the economy. Eric Lonstein provided outstanding project management and research assistance. All errors remain our own.

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 US, more than 110,000 firms engaged in the production and sale of packaged and custom-developed software, and related 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 software created within organizations who simply use software as an input to carry out non-software business (Mowery 1996). The extent of internal software production and investment is considerable, most firms typically spending 50% more for new, internally developed software than for software obtained through external vendors (Steinmueller 1996). More recently, open source software communities have also emerged as viable creators of large-scale "free" software (Lerner & 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 gets produced–representing a patchwork or confederacy of heterogeneous organizations. Software is developed in such diverse settings as small entrepreneurial firms, departments in large multi-national organizations, universities, outsourcing consultancies, collaborative endeavors like open source software communities, and the proverbial "garage." These differences can go beyond simple work rules, and relate to profound differences in institutional character as those among the "software factories" in Japan, the scientific method reflected in the practices of European electronics and technology champions, the ordered, engineering

---

[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 IDC projects the direct global software developer population to exceed 17 million individuals by 2011 (IDC Report 1517514).

orientation 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 organizations of different kinds, the work itself might also be organized in wildly diverging procedures (Cusumano et al. 2003). A given project might be organized around 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; it might utilize paired "agile" programming arrangements; it might involve internal developers working closely with an external open source community or, it may even involve internal and external 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 a clear resolution of one 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 the importance and difficulty of motivating developers (Sarah Beecham et al. 2008; Sharp et al. 2009), resulting in a stream of work including over 500 papers (Sharp et al. 2009). This large body of work from the 1950s to today point to a range of motivators, including the attraction to software variously lies in 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; opportunities for growth, achievement, and career recognition (e.g., Brooks 1975, Bartol and Martin 1982).[3] Consistent across this work is the notion that the work is, itself, a reward—creating an overlap between the costs and benefits of software development (Weinberg 1971; Schniderman 1980; Lakhani and Wolf 2003). As a group, software developers have tended to identify more with the profession and occupational

[2] For example, Microsoft's various changes in development process are well chronicled by Cusumano and colleagues (Cusumano 1991; Cusumano & Selby 1995; Cusumano & Yoffie 1998) and Sinofsky & Iansiti (2010).
[3] Beecham et al.'s (2008) review of the post-1980 literature on the motivations of software developers identified 21 sources of motivation.

community than with the organizations in which they toil (Couger and Zwacki 1980) and their behaviors are also swayed by norms in the profession. Crucially, Beecham et al. (2008) note that this long list of motivations should be understood as describing population averages, where individual software developers in fact possess complex and distinct, *heterogeneous* sources of motivations. The literature also documents considerable heterogeneity in preferred social interactions during the course of software production. On the one hand, in relation to other professions software developers have been found to require the least need for social interaction both on and off the job (Couger and Zawacki 1980). While other studies have reported the interdependent team structures improved productivity at the individual level and were better suited to tackling more complex tasks (Schniderman 1980, Couger and Zawacki 1980).

There are any number of reasons to 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—an "institutional preference". In very preliminary steps towards investigating a link between organizational heterogeneity and worker heterogeneity, here we report field experimental results where we test whether there might be an efficiency effect of workers sorting into institutional regimes of their preference, and particularly whether sorted workers experience higher motivation, as evidenced by their choice of effort.

In our experiment, over 1000 workers were assigned to groups of twenty virtual online "rooms" that were organized either in team-"cooperative" or autonomous-"competitive" regimes and worked to solve the same problem. In the competitive regime, individuals competed against all others in the room; in the cooperative regime, individuals were assigned to one of four teams of five workers, with the teams competing against each other. These two regimes hardly replicate the full variety of regimes we observe in the confederacy of software organizations. However, they do exhibit a range of starkly opposing features that accord with different work approaches in software development, i.e. software developers either work on their own or in teams. We divided participants in "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. This group therefore comprised 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 20 competed for $1000 in prizes; for others there was 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 average hours 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 2 outlines the basic approach to running the sorting experiment in such a way as to compare groups that are sorted and unsorted on the basis of institutional preferences, with the important feature that they possess identical skills distributions. In Section 3 we present the sample and variables. Section 4 presents results, comparing mean outcomes across sorted and unsorted groups. Section 5 provides concluding remarks.

2 Experimental Design

In our experiment, we consider the possibility that the extraordinary heterogeneity in organizations and in workers in the 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 we assign without regard to their preferences. Thus, we effectively compare the effort (and underlying motivations) of a "sorted" group, in which 100% of participants prefer their regime to which they are assigned, to an "unsorted" group, which has the population average distribution of preferences.

2.1 Field Experiment Context

Given the emphasis on measure the size of effect in relation how different *types* of workers behave under different circumstances, a field setting has a clear advantage of providing more meaningful estimates than a lab setting. Nonetheless, estimation of sorting effects requires an especially controlled environment. We conducted the experiment on the TopCoder open software innovation platform.[6] TopCoder is an online two-sided platform that produces software for clients via online contests amongst its member base of over 200,000 individuals. This provided a field context with real, elite software developers; but, at the same time, the context provided an unusual ability to perform manipulations and to observe relevant microeconomic variables. Over the 10-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

[6] Boudreau, Lacetera and Lakhani (2011) use the TopCoder context to analyze the impact of increasing competition on performance in software contests. They provide considerable detail on the TopCoder setting.

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 a 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 5 people (Carmel 1999). They are also regularly solved by participants in TopCoder tournaments.

2.2 An Assignment Procedure to Divide Participants into Sorted and Unsorted Groups with Identical Skills Distributions

A central challenge to our experiment is presented if institutional preferences are correlated with skill. In such a case, differences in behavior would reflect these skills differences in addition to any differences between sorted and unsorted groups *per se*. So as to assure 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 1. The goal of our approach is to create groups or "virtual rooms" of 20 participants drawn from the same skills distribution (and equivalent unobserved characteristics), but who had 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, essentially we created a rank order of all participants. We then divide 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 the 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.[7] The resulting subgroups were assigned to the cooperative and competitive regimes.

Important to note, the groups who 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 pair, however, 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 towards 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 20 individuals, in virtual web-based "rooms". Cooperative rooms were formed of four teams composed of five individuals (also randomly formed). We "mirror" this random assignment in the unsorted group, assigning ordered pairs to comparable groups.

[7] 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 towards 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 $1000 cash prize and VIP access to one of the few remaining NASA Space Shuttle launches. In addition, we also randomized the presence of room-level incentives in our experiment by offering $1000/room cash prizes to 24 rooms (12 competition regime rooms and 12 cooperation regime rooms – equally split between sorted and their skills matched unsorted groups). Thus, if a sorted participant was assigned to a room with a $1000 cash prize, so was his ordered pair in the unsorted group. Note that the participants did not ex-ante know if they would be competing for room level prizes or not.

2.3 The Cooperative and Competitive Regimes

Our primary unit of analysis of competition was the 20-person groups of direct competitors we created. The $1000 cash prize, if present, was divided among the top five competitors: $500 for the first place, and $200 for the second place, $125 for third place, $100 for fourth place, and $75 for fifth place. Individuals could see the list of the other 19 competitors on their "head-up" display, with their "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. Scores of existing submissions by all competitors in a room also appeared alongside competitor names.

The cooperative regime also involved 20 individuals in a virtual room, with 5 prizes. However, in this case, the 20 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 determined by the team with the highest submission (any team member could make submission). The winning team in the cash prize treatment divided the $1000 by an anonymous poll of team members (after the competition but before announcing winners) as to how each team member 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 to that point in time by other teams.

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 non-random 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) to the experiment. Of the half of participants who were asked their preference (the sorted group), 34.9% expressed a clear preference for the cooperative, 50.5% a clear preference for the competitive, regime.[10] The remaining 15.6% 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 negligible effect on the results.)

Of the rooms formed, only twelve rooms (44% of the sample), six sorted and six unsorted, competed for cash prizes amounting to $1,000 per room.[12] Prizes were first

[10] We originally targeted half the entire group of 1,098, but did not receive responses from a small fraction of individuals.
[12] We chose 12 simply because participation in the experiment exceeded expectations, and we had not budgeted for more than 12 prizes for the competitive regime.

assigned randomly across the sorted rooms. The "mirror" rooms of ordered pairs with corresponding assigned competitors were then also allocated $1,000 prizes.

3.1 Variables

This section discusses the meaning and construction of variables used in the analysis.

3.1.1 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 10-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 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.[13] (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

---

[13] Nearly all participants who submitted solutions responded. A research assistant who contacted 100 of the non-submitters 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 non-respondents by filling in zero hours as a relatively precise approximation. It became clear through interviews with non-submitters 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 non-submitters. Worthy of note, however, is that a number of non-submitters whom we discovered had worked a non-trivial number of hours before choosing not to submit did respond to the survey.

of value) and is easily interpreted. The results do not depend on which of the two measures of effort we use in the analysis.

<TABLE 1>

<TABLE 2>

3.1.2 Explanatory Variables

The key explanatory variable, *SortedonPreferences*, indicates whether a competitor was in a sorted or random assignment group. A second explanatory variable, *CashPrize*, indicates observations/individuals 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, 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.[14]

3.1.3 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.

[14] 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 et al (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.

4 Results

The average number of hours worked by participants during the 10-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 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.[16] 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% difference, compared to 11.57 and 8.97 hours, respectively, in the cooperative regime, a still large but considerably smaller 29% difference).

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.[17] 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.) The particularities of team dynamics go beyond the scope of our analysis here.

<TABLE 3>

---

[16] We found the differences in magnitudes to be surprisingly small and statistically insignificant, given the usual predictions of moral hazard in teams (Holmstrom 1982). Section 1.1.1 presents evidence consistent with the possibility of complementary effort choices that might plausibly be associated with any number of mechanisms such as socialization, and mutual monitoring in the cooperative regime that go beyond the usual notions of moral hazard in teams.

[17] Consistent with this interpretation, we find that sorted teams posted greater numbers of intra-team communications on the private team, online bulletin board.

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 to better interpret results. Ordinary least squares regression results with robust standard errors are reported in

	Dependent Variable = *HoursWorked*							
	Competitive Regime					Cooperative Regime		
Model:	1	2	3	4	5	6	7	8
Explanatory Variables	Two-Way Correlation	Linear Sklls Control	Skills-Level Dummies	Ordered Pair Diffs	Prize Control	Two-Way Correlation	Ordered Pair Differences	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-Squared	.04	.04	.05	.55	.09	.04	.55	.09

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; heteroskedasticity robust standard errors reported.

4.1.1 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.[19] We focus first on the results for the competitive regime. For ease of comparison, model (1) simply reiterates the two-way correlation of *HoursWorked* with *SortedonPreferences* from the competitive regime (essentially equivalent to the earlier stratified comparison of means in). Model (2) re-estimates the *SortedonPreferences* 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 sorted, changes slightly more (from 6.60 to 8.07), but the difference is statistically insignificant. To control for possible

[19] This includes differences in skill or unobservables correlated with skill.

subtle non-linearities, Model (3) adds dummies for different bands of skill levels to capture possible non-linear effects, but the estimated coefficient on *SortedonPreferences* is statistically identical and virtually unchanged (8.36 versus 8.33). Model (4) provides a most stringest skill control by simply comparing each sorted individual to its ordered pair, calculating the difference (implemented by simply including ordered pair fixed effects). The estimated effect is again statistically unchanged (although this most stringent control only leads to a slightly large coefficient). Given the random selection of rooms that should receive prizes, the introduction of *CashPrize* to the model should also not have any effect on the estimated coefficient *SortedonPreferences*.[20] Therefore, each of these coefficient estimates is thus statistically identical to the simple comparison of means presented in (14.92 – 6.6 = 8.32 hours).

Importantly, the coefficient on *CashPrize* also provides some indication of the relative impact of sorted versus the formal incentive instrument used in this context, the $1,000 prize. The coefficient on *CashPrize*, 9.12 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 that estimates of the *SortedonPreferences* coefficient are insensitive to the various controls. Model (6) reiterates the two-way correlation of *HoursWorked* with *SortedonPreferences* from the cooperative regime (essentially equivalent to the earlier stratified comparison of means in), 2.6 additional hours for inviduals who sorted in the cooperative regime. Re-estimating the effect on the basis of directly comparing ordered pairs (model 7) or introducing *CashPrize* and controls for different levels of skills (model 8) generate similar estimates. The estimated coefficient on *SortedonPreferences* is 2.60 hours. Model (6) essentially re-estimates model (4) with each of the controls, but for the cooperative regime. Including each of the controls does not significantly change the

[20] We must go back to a model estimated on the basis of ordered pair differences, given there is not variation in CashPrize within ordered pairs, given the assignment procedure assured that if one member of an ordered pair was in a group with a prize, the situation would be mirrored for the other pair.

coefficient on *SortedonPreferences* (2.47 hours). Again, these estimates are statistically the same as those from the simple comparison of means in (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.[21]

<TABLE 4>

4.1.2 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 preferences with the institutional context on choice of effort. But it might still be the case that individuals make 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 where subjects' preference 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 where 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 5.

Therefore, we focus on the 15% sorted participants who chose a neutral response when asked to gauge their relative preferences for regimes (i.e., those who expressed "I am

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

indifferent or I am not sure"[23]). A possible limitation with this approach is that a neutral view of the cooperative regime may have in fact implied some level of openness to an interest in this regime (given the competitive regime is in fact the usual TopCoder regime).[24] To better isolate those whose statement of preferences were more likely to have been shaped by an exogenous factor other than their inherent preferences, we surveyed individuals' preferences with a survey instrument whose order varied, sometimes presenting the competitive regime first and other times presenting the cooperative regime first. As presented in model (1) the order of the question significantly affected the statement of preferences. Re-estimating the model on this 15% of the sample (156 observations) results in a statistically identical estimate of the coefficient on *CashPrize*, but the coefficient on *SortedonPreference* goes to zero, suggesting zero Hawthorne effect.[25]

<TABLE 5>

4.1.3 An Approach to Re-weighting to Directly Compare the Different Sorted Groups

Whereas the skills distributions were, by design, the same across the sorted and random assignment groups, we should expect that that sorting generated differences in skills distributions across competitive and cooperative groups. Figure 2, Panel I presents the distribution of skills of participants that 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 *SortedonPreferences* in the

[23] Recall that indifferent individuals were assigned to the cooperative regime (Section 2.2).

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

[25] 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 is considerably larger than the IV estimate, remains statistically indistinguishable from zero while the coefficient on *CashPrize* strikingly remains virtually unchanged in magnitude or significance in this model.

cooperative and competitive regimes should not be directly comparable if the magnitude of an individual sorted effect is somehow related to skill.

To enable more direct comparisons of the magnitude of effects in the cooperative and competitive regimes, we re-estimate effects while re-weighting the data from the competitive regime to have the same skills distribution as that of the cooperative regime (as in Figure 2, Panel II). As reported in Table 6, the re-estimated the model on competitive data, reweighted to share the skills distribution of the cooperative regime increases the estimated coefficient on *SortedonPreferences* from 8.32 hours to 10.28 hours. The estimated effect on CashPrize, by comparison, drops to 6.74.

<FIGURE 2>

<TABLE 6>

5 Conclusions

Software design and development is done in settings as diverse as small entrepreneurial firms, large multi-national organizations, universities, outsourcing consultancies, collaborative efforts like open source software communities, and the proverbial solo developer in the garage. Seemingly, just as diverse and heterogeneous is the pool of software developers that work in these organizations. This paper took very preliminary steps towards investigating whether there might be a link between heterogeneity of organizations and workers by assessing whether sorting of software workers to the regime of their preference has an effect on the motivations and effort exerted.

We devised a novel sorting experimental method that allowed 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 their preferences—while assuring that both group possessed identical distributions of raw problem-solving ability. Thus, in contrast to more conventional experimental approaches, which 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 roughly doubled effort, on average. In the cooperative regime, estimates were smaller—although still large—with a roughly 30% increase. Estimates were similar across a range of specifications. We also devised a method to explicitly estimate any Hawthorne effects that may have resulted from the approach we used to elicit individuals' preferences (based on an instrumental variables estimate on a subsample of the data) and found these to be statistically indistinguishable from zero.

There are, of course, many limitations and endless remaining work in investigating possible links between worker heterogeneity and organizational heterogeneity in software (and other contexts), in a competitive economy where firms and workers match in equilibrium. In relation to the particular experiment conducted here, the analysis presented here focused on estimating mean differences, rather than more detailed analysis of distributions of outcomes or associated demographic worker attributes. The analysis here focused on 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 focused on effort and did not study effects on overall performance and productivity. There was also an indication in results presented here that sorting may have created more subtle effects in organization of and patterns of collaboration in the cooperative regime that were not investigated 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 micro-data on scientific and technical workers and the links between incentives and creativity (Azoulay et al 2009); preferences for work environments (Stern 2003); and the organization of scientific teams (Jones et al 2008). As individual and team level productivity issues for creative workers becomes increasingly more salient for organizational and national level performance (Radner 1993; Hong and Page 2003), this stream of research (and future related work) has the potential to provide relevant theoretical, empirical and practical insights.

6 References

Azoulay, P., Zivin, J. & Manso, G., 2009. Incentives and Creativity: Evidence from the Academic Life Sciences. *NBER Working Paper #15466*.

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

Beecham, S. et al., 2008. Motivation in Software Engineering: A systematic literature review. *Inf. Softw. Technol.*, 50(9-10), 860-878.

Boudreau, K.J., Lacetera, N. & Lakhani, K.R., 2011. The Effects of Increasing Competition and Uncertainty on Incentives and Extreme Values in Innovation Contests. Management Science forthcoming.

Brooks, F.P., 1975. *The mythical man-month : essays on software engineering*, Reading, Mass.: Addison-Wesley Pub. Co.

Carmel, E., 1999. *Global software teams : collaborating across borders and time zones*, Upper Saddle River, NJ: Prentice Hall.

Couger, J.D. & Zawacki, R.A., 1980. *Motivating and managing computer personnel*, New York: Wiley.

Cusumano, M. et al., 2003. Software Development Worldwide: The State of the Practice. *IEEE Softw.*, 20(6), 28-34.

Cusumano, M. & Selby, R., 1995. *Microsoft secrets: how the world's most powerful software company creates technology, shapes markets and manages people*, New York, NY: Free Press.

Cusumano, M.A., 1991. *Japan's software factories : a challenge to U.S. management*, New York: Oxford University Press.

Cusumano, M.A., 1997. How Microsoft makes large teams work like small teams. *Sloan Management Review*, 39(1), 9-20.

Cusumano, M.A. & Yoffie, D.B., 1998. *Competing on Internet time: lessons from Netscape and its battle with Microsoft*, New York: The Free Press.

Cusumano, M.A., 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.

Holmstrom, B., 1982. Moral Hazard in Teams. *The Bell Journal of Economics*, 13(2), 324-340.

Hong, L. & Page, S.E., 2001. Problem Solving by Heterogeneous Agents. *Journal of Economic Theory*, 97(1), 123-163.

Jones, B.F., Wuchty, S. & Uzzi, B., 2008. Multi-University Research Teams: Shifting Impact, Geography, and Stratification in Science. Science 322, 1259.

Lakhani, K.R. & Wolf, R., 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*. Cambridge, MA: MIT Press.

Lakhani, K.R. & von Hippel, E., 2003. How Open Source Software Works: Free User to

Lerner, J. & Tirole, J., 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, VA: National Science Foundation. Available at: 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), 1109-1146.

Salop, J. & Salop, S., 1976. Self-selection and turnover in the labor market. *The Quarterly Journal of Economics*.

Schneiderman, B., 1980. *Software Psychology: Human Factors in Computer and Information Systems*, Scott Foresman & Co.

Sharp, H. et al., 2009. Models of motivation in software engineering. *Information and Software.*

Sinofsky, S. & Iansiti, M., 2010. *One strategy! : organization, planning, and decision making*, Hoboken, N.J.: Wiley.

Steinmueller, W.E., 1996. The U.S. Software Industry: An Analysis and Interpretive History. In David C. Mowery, ed. *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure*. New York: Oxford University Press.

Stern, S., 2004. Do scientists pay to be scientists? *Management Science*, 50(6), 835-854.

Weinberg, G.M., 1971. *The psychology of computer programming*, New York,: Van Nostrand Reinhold.

7 Tables

Table 1 Variable Definitions

	Variable	Definition
(1)	*HoursWorked*	Number of hours worked by an individual participant during the course of the experiment
(2)	*NumSubmissions*	Number of solutions submitted to be compiled, tested and scored by an individual participant during the course of the experiment
(3)	*SortedonPreference*	Indicator switched to one for participants who were asked their preferences regarding the regimes and subsequently assigned to their preferred regime
(5)	*CashPrize*	Indicator switched to one for participants within a group of 20 that competed for a $1000 cash prize
(6)	*SkillRating*	Measure of general problem solving ability in Algorithmic problems based on historical performance on TopCoder platform
(7)	*LikertScale*	1-4 integer response indicating relative preference for competitive regime, where higher scores indicate greater relative preference for competitive regime over cooperative regime
(8)	*OrderofQuestion*	Indicators switched to one when questionnaire to elicit preferences was designed to present competitive regime first and cooperative regime second

Table 2 Means, Standard Deviations and Correlations

	Variable	Std. Dev.	Min	Mean	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	*HoursWorked*	18.75	0	190							
(2)	*NumSubmissions*	5.63	0	42	.57						
(3)	*SortedonPreference*	.50	0	1	.15	.05					
(4)	*Competition*	.50	0	1	.01	.07	.01				
(5)	*Prize*	.50	0	1	.25	.20	.01	-.03			
(6)	*SkillRating*	538.39	188	3797	.03	.20	.01	.18	.07		
(7)	*LikertScale*	1.38	1	5	.07	.17	n/a	.88	-.03	.18	
(8)	*OrderofQuestion*	.50	0	1	.03	.01	n/a	.07	.00	-.01	.10

Table 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

Table 4 OLS Estimates of Sorting Effect

	Dependent Variable = *HoursWorked*							
	Competitive Regime					Cooperative Regime		
Model:	1	2	3	4	5	6	7	8
Explanatory Variables	Two-Way Correlation	Linear Sklls Control	Skills-Level Dummies	Ordered Pair Diffs	Prize Control	Two-Way Correlation	Ordered Pair Differences	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-Squared	.04	.04	.05	.55	.09	.04	.55	.09

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; heteroskedasticity robust standard errors reported.

Table 5 IV Estimate of Hawthorne Effect

Explanatory Variables	Dependent Variable = *LikertScale*	Dependent Variable = *HoursWorked*
	1	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-Squared	.05	

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; heteroskedasticity robust standard errors reported.

Table 6 Re-estimated Results from Cooperative Regime to Match Skills Distribution of Cooperative Regime

Explanatory Variables	Depenedent Variable = *HoursWorked*
	Competitive Regime
SortedonPreference	10.2814***
	(2.08)
CashPrize	6.7416***
	(2.20)
Skills Dummies	Yes
R-Squared	.12

Notes. *, **, and *** indicate statistical significance at the 10%, 5% and 1% levels, respectively; heteroskedasticity robust standard errors reported.

Figure 1 – Overview of Experimental Assignment

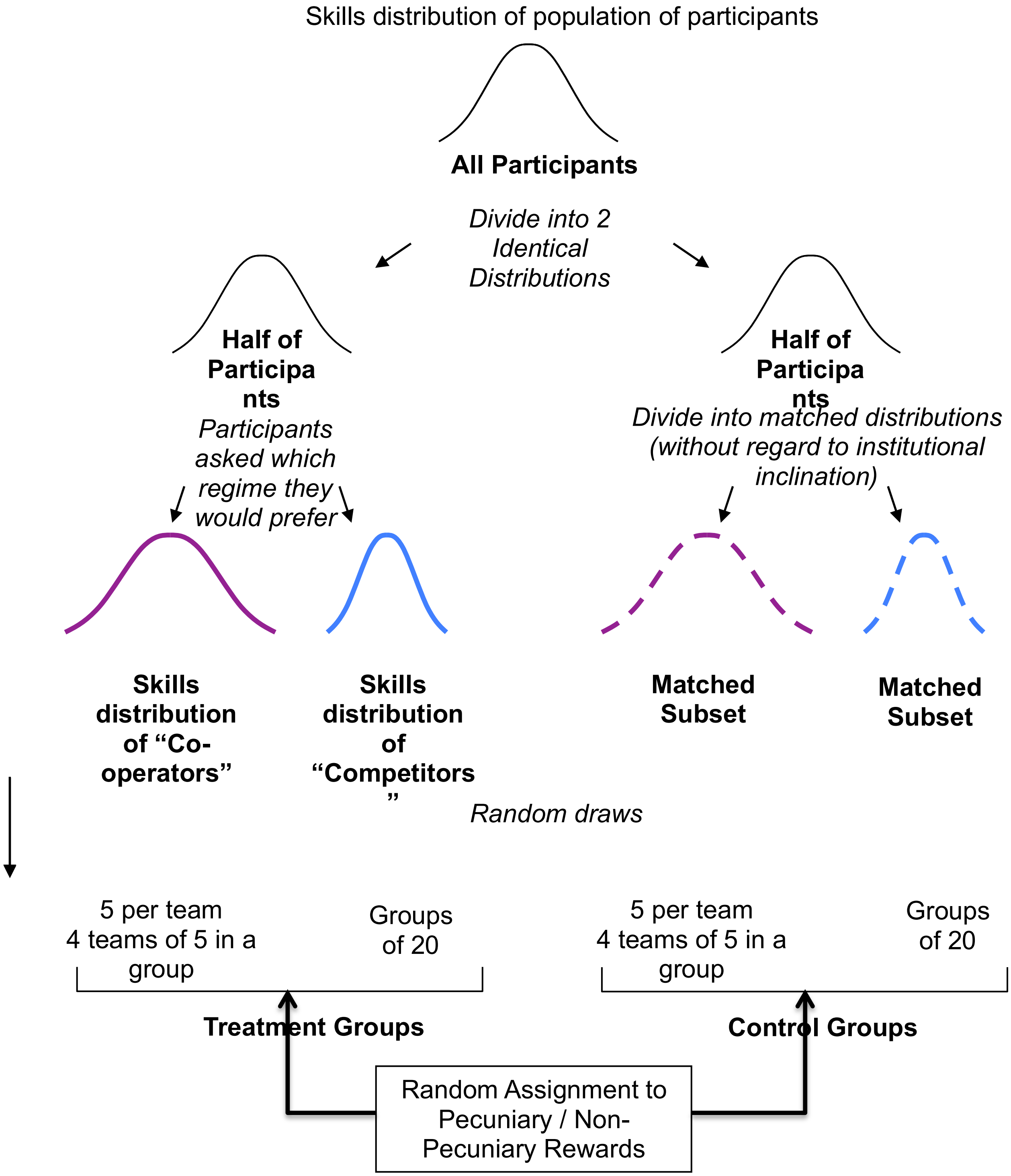

Figure 2 Skills Distribution in Competitive and Cooperative Regimes