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

Chapter Authors: Iain M. Cockburn

Chapter URL: http://www.nber.org/chapters/c12367

Chapter pages in book: (p. 502 - 505)

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.

pleased to have the opportunity to discuss a creative and intriguing chapter on this topic. Particularly one with such a startling result: when I first read this chapter my immediate reaction was "holy cow!" Could simply giving people the opportunity to self-sort into their preferred regime of work structure and incentives really result in a doubling of effort? I suspect that most of us have probably introspected at some length on the central question raised by this chapter (how sensitive is effort by knowledge workers to their organizational context) in the context of our own work environment, and will find the magnitude of the effect intriguing. If only the Dean would just move me out of this department into that department . . . if only I were working on the same research questions, but at Google and with stock options . . . Would I really work twice as hard? Would I generate twice as much output?

Beyond idle speculation, addressing these questions empirically means confronting some quite serious problems with treatment and selection. These are, of course, difficult to deal with by looking at observational retrospective data, and I am pleased that the authors have given us a piece of experimental evidence to help us think about the problem. It is also noteworthy that this experiment is being run in the field using real people working on a real task rather than in a lab, although I am somewhat skeptical about the economic significance of the rewards and opportunity costs of participation to the programmers, as well as the significance of the output of these problem-solving teams to the "customer" (NASA). Knapsack problems are an old topic in mathematics, and NASA's engineers seem likely to have developed, refined, and implemented their own solutions to this specific problem many times over the history of the agency.

There are many things to like about this project. But I do have a few comments. The first is that the chapter focuses on measuring supply of effort, rather than on the nature of output. At least in my Dean's office, they don't appear to care much about effort. What they care about is outcomes and output. I think there is an unexploited opportunity here to look more closely at the output of the participants in the experiment. I assume that the Top Coder platform allows some quite nuanced observation of output: presumably the same mechanism that generates the quality rating for the programmer could generate a quality scoring for their solution to this knapsack problem, and there is an objective measure of the performance of each team's algorithm—fraction of wasted space. It would be very interesting to look at whether or not organizational context affects the quality of output, in the sense of better solutions, rather than just the ability to arrive at some solution. If people are allowed to self-sort into one group or another, do they produce more effective, more elegant, or more robust solutions to the problem? I'm not sure I have a prior on this, but would be very interesting to see the data.

Second, the authors focus on effort measured as self-reported hours. My guess is that self-reporting of hours could easily be biased. I am not sure which way it might be biased, but I have a feeling that this is going to be correlated, potentially in some important way, with worker type and their work context. For example, I suspect that the individuals at the top end of the skills distribution would both be more likely to always prefer to compete on an individual basis, and less likely to report truthfully that they had spent 200 hours over ten days working on the problem as opposed to claiming that they had solved it in 90 minutes. A potentially more reliable measure of effort may be the number of submissions, and, interestingly, when this measure is used; while the flavor of the results is generally the same, the magnitudes are lower.

Third, I am concerned about endogenous selection that has not been controlled for by the experimental design. One puzzling feature of these data is the substantial number of people who effectively selected themselves out of this experiment by turning in less than one hour of effort. I am not sure how we should think about this—are these the people who got randomized into a work context they did not like, or are they the ones who look at the problem and realize they have better things to do? Clearly if the selecting out is nonrandom there are substantial problems for understanding and interpreting the results from this experiment. So it would be very helpful to show us, for example, any differences in observable characteristics of those individuals who selected themselves out ex post versus inserting themselves into a work regime ex ante.

Turning from the specifics of the analysis, this chapter raises some more general questions for me. Team production may be the rule rather than the exception in knowledge work: in many activities, the scale and complexity of projects and the need to repeatedly combine highly specialized skills and knowledge may make it impossible or at least economically unattractive for individuals to work in isolation. In which case, organization, incentives, and governance of team production may be an important driver of productivity, and taken at face value the results of this chapter suggest that these may in fact be critical.

Like many other business schools, my employer emphasizes teamwork and team projects in our MBA teaching. Students are more or less randomly assigned to teams, and rewarded for the quality of their joint output. (A particularly good first assignment is to ask them to summarize Holmstrom's "Moral Hazard in Teams" article in a single Powerpoint slide.) After watching this process for a few years I am struck by several aspects of what happens over the course of the semester. Left to themselves, most such randomly assembled teams appear to quickly self-organize into an effective production unit, with clear allocation of tasks and general consensus on goals and priorities. But relatively few teams seem to be able to realize big gains from combining complementary attributes of team members without considerable effort and practice, and a small minority become dysfunctional and fall apart. By the end of the semester, however, most students have mastered the art of teamwork, and typically they report that this is one of the most useful things they learned in business school. What this suggests to me is important roles in knowledge work for both the structure and incentives of team organizations and for heterogeneity among team members in their innate or learned ability to work collaboratively.

Finally, let me focus briefly on the economic setting of this chapter, the software industry. The software sector has produced one of the most interesting new organizational forms of the contemporary economy, the open source software movement. Software is also a technology that seems to disproportionately attract distinct types of people—the Hollywood stereotype of "pale-skinned disheveled young men, slumped over a keyboard in a darkened room" may not be wholly accurate, but surely reflects some important aspects of the labor force—and software firms appear to rely disproportionately on the output of a very small minority of workers. But so do many other knowledge-intensive or creative industries, and I am less certain than the authors that the industry structure of software is uniquely different from other sectors. Nor is it obvious that software workers necessarily respond differently to opportunities to self-select into their preferred organizational structure than those in other occupations. I am provoked therefore to speculate about the results of repeating this experiment in different contexts. This would be very helpful for establishing the broader implications of the results for thinking about the organization of knowledge work, and might provide opportunities to test the robustness of the methodology. A twofold increase in output attributable to the option to self-select into one's preferred organization of work seems very large, but it is not clear what the relevant benchmark might be, and whether this effect should necessarily be larger for knowledge workers versus other workers. I would not be surprised if similar results were obtained for work tasks involving manual labor or mechanical rather than mental dexterity.