HARVARD | BUSINESS | SCHOOL

The Contingent Effect of Absorptive Capacity: An Open Innovation Analysis

Andrew A. King
Karim R. Lakhani

Working Paper

11-102

The Contingent Effect of Absorptive Capacity: An Open Innovation Analysis

Andrew A. King
Tuck School of Business
Dartmouth College
114 Buchanan Hall
Hanover, NH 03755, USA
Phone: (603) 646-8985
Email: andrew.a.king@tuck.dartmouth.edu

Karim R. Lakhani
Harvard Business School
Harvard University
433 Morgan Hall
Boston, MA
Phone: 617.401.7332
Email: k@hbs.edu

Abstract: Technological advancement and innovation requires the integration of both external knowledge and internal inventiveness. In this paper, we unpack the concept of absorptive capacity and separately explore the effect of different types of prior experience on the capacity to adopt external knowledge and make internal inventions. We also measure how absorptive capacity is influenced by changes in design "paths". We investigate nine open source programming contests in which 875 software programmers submit over 4.7 million lines of code. We conduct our analysis at the individual level and identify how programmers gain the ability to adopt and invent valuable code. Our evidence both confirms the theory of absorptive capacity and suggests refinements to it. We find that prior experience with both adoption and invention can indeed improve the capacity to adopt and invent valuable code, but we find that experience with adoption has the largest effect on invention capacity. We also find that major changes in the design "path" both advance and impede absorptive capacity. Changes in path allow rapid experience with alternative ideas, and this eventually aids adoption and invention capacity. However, these changes temporarily harm the ability of programmers to create valuable inventions. We discuss the implications of our findings for the literature on absorptive capacity and open and distributed innovation.

Version Date: April 3, 2011

1 Introduction

It is now considered axiomatic that success at technological innovation requires the integration of both external knowledge and internal inventiveness (Cohen and Levinthal 1990; Arora and Gambardella 1994; Cassiman and Veugelers 2006; Volberda, Foss, and Lyles 2010). At all levels of analysis, innovators have been shown to rely on external knowledge for technical problem solving (Cohen and Levinthal 1989, 1994; Rothaermel and Alexandre 2009; Hoang and Rothaermel 2010; Csaszar and Siggelkow 2010). Yet the connection between adoption of external ideas, invention, and performance has been incompletely specified. For example, do experiences with adoption improve the capacity to invent? Does invention experience increase the capacity to adopt? And how are either influenced by changes in the overall design approach being used to solve a technical problem? Separating out these effects has been prevented in part by the difficulty of distinguishing adoption, invention, and changes in overall design approach. In this article, we use an unusually high fidelity dataset to begin to unpack these issues.

Cohen and Levinthal (1989,1990) assigned the term “absorptive capacity” to the general capability of individuals, groups, and firms to recognize the value of new information, choose what to adopt, and apply it to innovation. Essential to the concept of absorptive capacity is the idea that accumulated experience with adoption and invention improves the capacity to recognize and absorb high quality external ideas and create valuable inventions. Over the past two decades the absorptive capacity concept has gained currency amongst scholars, resulting in it being cited in over one thousand publications and operationalized in hundreds of research articles (Volberda et al, 2010).

As described by Cohen and Levinthal (1990: pg 133), absorptive capacity has two constituent components: 1) the capacity to adopt ideas from the outside world, what we call “adoption capacity”; and 2) the capacity to create new inventions, what we refer to as “invention capacity”. Cohen and Levinthal (1990) also suggest that “prior related experience” influences absorptive capacity, but they do not distinguish experience gained from prior adoption of external ideas from experience gained through internal invention.

Although the concept of absorptive capacity has been immensely influential, recent systematic reviews of the extant literature have identified areas lacking empirical support (Volberda et al 2010; Zahra & George 2002; Lane, Koka and Pathak 2006). One surprising realization has raised particular concern: although the foundation of the concept is based on theories of individual learning (Cohen & Levinthal 1990), the theory has seldom been investigated and tested at the individual level (Volberda et al 2010; Zahra & George 2002; Lane et al 2006). Tests at higher levels of aggregation (e.g. organization, unit, and firm) also tend to prevent the separation of different types of prior experience and different types of capacity. As a result, previous research has not separately identified the role of adoption and invention experience or measured their differing effect on adoption and invention capacity.

In this paper, we decompose the impact of prior experience with a) adopting external knowledge and b) internal invention on both adoption and inventing capacity. Specifically we measure capacity in terms of the technical quality of adoption and invention. Furthermore, we explore how absorptive capacity is affected by the decision of individuals to adopt entirely new design "paths" in their search for a better solution to innovation problems. Finally, we shift attention away from firms and focus on individuals (i.e. those that are actually engaged in absorbing and using knowledge) as the focal actors engaged in innovation related problem solving. Figure 1 shows the overall plan of our study design.

Include Figure 1 about here

Recent changes in the context of innovation have made the study of absorptive capacity even more important. Open and distributed innovation, such as collaborative open-source code development (Lerner and Tirole 2002; von Hippel and von Krogh 2003) and innovation contests (Terwiesch and Xu 2008; Jeppesen and Lakhani 2010; Boudreau, Lacetera, and Lakhani 2011), now allow innovators to absorb ideas from many disparate sources. In some cases, they can even copy the entire design approach of another actor (Haefliger, Von Krogh, and Spaeth 2008). This mode of organizing innovation has had a major impact on the software industry via the open source software movement (von Hippel and von Krogh 2003), and it is becoming increasingly prevalent in industries as diverse as fashion design, drug discovery, and content production (von Hippel 2005; Benkler 2006; Murray and O'Mahony 2007; Murray et al. 2008). This setting provides an ideal empirical environment to study absorptive capacity because the intellectual property rules bestow upon all actors the ability to view, access, and reuse the ideas and inventions of others (Murray and O'Mahony 2007). As a result, choices of what external ideas to adopt represent an accurate measure of the ability of the actor to assess the quality of those ideas.

Our research brings the theory of absorptive capacity to the study of open and distributed innovation. Although distributed innovation has garnered substantial attention in the academic literature, we are unaware of any scholarly work that has explored the effect of absorptive capacity in open innovation. Yet it is precisely in this setting that absorptive capacity may be most important. Much of the promise of open-innovation arises from the potential for actors to borrow the best ideas from each other and then use them to inspire new inventions (von Hippel & von Krogh 2003; Murray and O'Mahony 2007). Our research explores how actors learn the skills needed to achieve this promise.

Our specific context consists of nine programming competitions sponsored by The MathWorks Corporation. Each competition ran for one week, and challenged software developers to use MATLAB code to create solutions to difficult mathematical problems (such as the "traveling salesman" puzzle). Code submitted throughout a seven-day period was immediately evaluated against a test-suite that objectively scored performance. Contest winners are provided with a nominal prize (t-shirt). A unique feature of this setting is that, after the first two days of the contest, all code submissions were made open for examination and reuse by anyone participating in the contest. Each game began with the submission of one example codebase from MathWorks itself. During the first 48 hours of each game (or the "dark period" as the contest organizers referred to it), programmers could see this submission, their own code, but the code of no others. After this, the authors could see every submission (including those made during the dark period). Thus a standard individual design competition was transformed into an open-innovation contest.

MathWorks tracked every author, entry, and line of code in each competition – allowing us to gather high fidelity data on all submissions made during the contests. Our data set includes nine competitions over seven years in which 875 individuals submitted more than 4.7 million lines of code. The number of code submissions in a contest ranged from 1631 to 4420.

Our analysis reveals that cumulative experience in inventing and using external knowledge increases an author's capacity to adopt better quality code and invent better performing code. It shows, however, that the effects are multi-faceted. First, it shows the importance of outward-oriented experience (i.e. adoption experience) in creating inward invention capacity. Indeed, out of all of the modes of development, invention capacity is most effectively built by experience with adoption. Second, it shows how changes in design paths influence absorptive capacity. Changes in path, we show, harm invention capacity and actually reverse the effect of invention experience on invention capacity. That is, during a switch to a new design path, previous experience inventing actually harms a programmer's ability to make useful inventions.

Our paper also demonstrates the power of open-innovation in supporting absorptive capacity. An open contest allows authors to see the numerous ideas created by the community, sift through them to find the better ones, and develop both adoption and innovation capacity from doing so. Adoption experience improves an author's ability to select good ideas. It also dramatically improves his/her ability to create better inventions. This in turn improves the quality of the ideas available to the community – encouraging further adoption and improving adoption and invention capacity.

Our paper is organized as follows. In section 2, we review the literature and develop hypotheses regarding the role of cumulative experience and path switching for innovation performance. In Section 3, we discuss our estimation strategy, and we provide results in Section 4. We discuss our results and offer concluding remarks in Sections 5.

2 Literature Review and Hypotheses Development

2.1 Absorptive Capacity

Scholars studying the economics of technical change have departed from the view that knowledge spillovers in the public domain are always easily accessible to any interested party. Instead they have shown that an actor's ability to extract valuable knowledge occurs mostly through their own internal investments in R&D (e.g. Cohen & Levinthal 1989; Arora & Gamberdella 1994; Cassiman and Veugelers 2006). Cohen and Levinthal (1989, 1990, 1994) have argued that innovators develop "absorptive capacity" through their investment in internal R&D which serves to both generate new inventions and also enhances the ability to more effectively exploit external knowledge.

The absorptive capacity literature has primarily operationalized innovation outcomes as either a stock of patents (e.g. Cockburn and Henderson 1998; Katila and Ahuja 2002; Rothaermel and Alexandre 2009) or new products produced (e.g. Cassiman and Veugelers 2006; Escribano et al 2006; Hoang & Rothaermel 2010) and the main measurement of absorptive capacity itself has been R&D intensity or investment (Cohen and Levinthal 1989,1990; Lane et al 2006, Volberda et al 2010; Escribano et al 2006). The literature has emphasized that absorptive capacity positively improves innovation outcomes (speed, quality, and frequency) and that the subsequent organizational learning from internal innovation efforts also recursively improves absorptive capacity itself (Lane et al 2006, Volberda et al 2010).

In the past decade, several overarching review papers have critically examined the use of the absorptive capacity construct (See for example: Lane et al 2006: Volberda et al 2010; Zahra & George 2002). These reviewers unanimously note that the construct is one of the most important to emerge in the fields of management, organizations and strategy (Lane et al 2006: Volberda et al 2010; Zahra & George 2002). However, the reviewers also express concern that absorptive capacity as a construct has become reified in the literature and weakened by measurement that is "diverse", "indirect," and "inaccurate" (Lane et al 2006; Volberda et al 2010).

Common among all the reviewers is the call for future scholars to link the construct back to its roots by initiating analysis that focuses on individuals as the main drivers of knowledge creation, acquisition, and use (Lane et al 2006, Volberda et al 2010) and to concentrate on the underlying micro-mechanisms of absorptive capacity creation and application (Volberda et al 2010). While the absorptive capacity construct has primarily been applied at the level of the firm, its theoretical foundations reside with individual cognitive structures (Cohen & Levinthal, 1990). Absorptive capacity is based on individual actors engaging in problem solving and learning activities that are then aggregated to the levels of groups and organizations. In reviewing the extensive literature on memory, problem solving, and learning, Cohen & Levinthal (1990) posit that absorptive capacity, as instantiated in individuals, is driven by two core interrelated ideas: "learning is cumulative, and learning performance is greatest when the object of learning is related to what is already known" (pg 131).

The cumulative nature of learning is based on findings that demonstrate that intensity of effort in the learning or problem-solving task is a major determinant of performance. The main driver of intensity is individual effort and the cumulative number of practice trials over similar problems (Harlow 1949, Chase & Simon 1973). Learning performance also increases with the stock of prior related knowledge held by the individual. The basic principle is that the prior stock of knowledge held by an individual enhances the acquisition of related new knowledge and the use of this knowledge in new settings (Hilgard and Bower 1975). The driver for this mechanism is associative learning where prior stored knowledge provides the scaffold by which new knowledge is assimilated and linked to existing concepts, categories, objects, and patterns in memory.

2.2 Untangling Invention and Adoption Experience in Absorptive Capacity

Core to the concept of absorptive capacity is the complementary effect of "outward-looking" accessing and adopting of external knowledge and the "inward-looking" internal invention process (Cohen & Levinthal 1990: pg 133). Any attempt to study the micro-mechanisms underlying absorptive capacity need to account for the fact that the effects of adoption and inventing need to be considered separately (Rothaermel and Alexandre 2009; Hoang and Rothaermel 2010). Yet, little research has attempted to separate experience in adoption and invention, or distinguished the effect of this experience on the quality of the ideas selected for adoption or invented through internal effort.

Both experience with invention and adoption contribute to the capacity of actors to invent and adopt ideas. Both provide related experience in analyzing the performance of a design and in identifying where to invest time in improvement. Invention experience builds skills in structuring complex problems and encourages deep thinking about alternatives. Invention, like any process, involves routinized steps, which can ease future invention (Simon 1973, Klahr and Simon 2001). It is not enough for individuals simply to "know something" via formal training; invention only improves if individuals repeatedly engage in the tasks that require creative use of knowledge (Ericsson 2006). Chase & Simon (1973) build on the work of de Groot (1965) to come up with the rule of thumb that world-class expertise in a task (in their case chess), required ten years of sustained effort or 10,000 hours. More generally, accumulation of experience in a given task results in performance improvements across a variety of settings and is the basis for the presence of the learning curve effect amongst individuals (Mazur and Hastie 1978) and organizations (Argote 1999, Haleblian and Finkelstein 1999).

Experience with invention also aids in the adoption of ideas from others. The act of inventing solutions forces the individual to gain a deeper, first-hand knowledge about the structure of the problem and understand the variety of potential solution approaches (Jonassen 2003, Baron 1988). It clarifies the objective function against which ideas are compared and enables the development of general heuristics that can be used to solve a problem (Hong and Page 2003). Armed with this knowledge about the problem and potential solution approaches, an individual can more fruitfully assess the ideas of others and adopt those that are most viable and productive based on their own invention experience. Given a wide choice of alternative ideas, individuals with increasing invention experience are more likely to select the ideas of others from a comparison group that is at the frontier of knowledge and solution development (Lewin and Massini 2003).

> ***Hypothesis 1–*** Experience with internal invention will lead to greater **a)** invention capacity and **b)** adoption capacity.

In contrast to experience with invention, experience with adoption provides authors with familiarity with alternative perspectives and approaches to problems. It stimulates creative thinking by breaking authors free from preconceptions. In studies of the role of experience on problem solving, Luchins (1942) found that individuals exposed to a solution to a complex problem overwhelmingly used the same (complex) methodology to solve simpler problems. Subjects were observed saying "how stupid I am" or "how blind I am" when they were later confronted with more effective solutions. Related work by Gordon (1961) also pointed to "blindness to solutions" as a main hindrance for effective problem solving. Experience with alternative ideas obtained through adoption can help overcome this "blindness" and encourage better invention. In particular, Csaszar and Siggelkow (2010) have argued that adoption and imitation enables innovators to "dislodge" themselves from their existing low performing solutions and escape to better outcomes. Lee, Posen and Yi (2010), use Alchian's Conjecture (1950) to propose that adoption, even if imperfect can provide the necessary serendipitous inputs to innovators to internal develop solutions that in some cases can be superior to existing approaches. Hence external learning can pay handsome dividends for internal innovation activity.

Central to the absorptive capacity concept is the notion that experience with adoption allows better and easier absorption of new external ideas. Experience with adoption, Cohen and Levinthal (1990, pg 130) note is akin to "learning to learn," which is a key factor in individual-level absorptive capacity. They cite two examples to illustrate the importance of prior related knowledge and experience for absorptive capacity: first citing Ellis (1965) they note that students with prior exposure and mastery of algebra do subsequently much better in mastering calculus, and second citing Anderson et al (1984) they note that learning a new computer programming language is much more difficult the first time as opposed to having some prior background in programming. Generally speaking, vicarious learning, i.e. learning from observing others, improves as individuals gain more experience with the actions directly (Gioia and Manz 1985).

These two arguments provide the underpinning for the second precept of absorptive capacity: that prior experience absorbing knowledge enhances an individual's ability to absorb new external knowledge and apply it productively:

> ***Hypothesis 2*–** Experience with adoption externally generated inventions will lead to greater **a)** invention capacity and **b)** adoption capacity.

2.3 Design Paths and Absorptive Capacity

Scholars now generally accept that innovation-related problem solving usually occurs on specific technological trajectories (Dosi 1982; Nelson and Winter 1982; Vincenti 1994b, 1993). A design trajectory (or "path" in our terminology) is the paradigm or approach used to organize the relevant knowledge to solve a particular problem (Dosi 1982, Nelson & Winter 1982). Innovation scholars have convincingly shown that multiple technical approaches may be available to solve the same problem (Nelson and Winter 1982; Dosi 1982; Clark 1985). For example, Vincenti (1994a, 1994b) has exhaustively documented the simultaneous emergence of multiple designs for aircraft landing gear – each one representing a novel take on the problem of safe airplane landing (Hong and Page 2003). Indeed, a robust competition may exist among proponents of competing approaches to demonstrate the performance characteristics of their particular approach (Vincenti 1994a; Suarez and Utterback 1995; Dosi 1982). As the design contest evolves, poor performing technical approaches are discarded in favor of the better performing ones, and innovators switch completely to new approaches or mix elements of approaches to form new directions (Vincenti 1993; Murmann and Tushman 1998; Utterback & Suarez 1993).

Changing the technological approach of a design, however, is not entirely effortless or free. Changes in technological paradigms also entail "that one has got to start (almost) from the beginning in the problem-solving activity" (Dosi 1982: 154). Research within cognitive and social psychology adds to the above view by emphasizing the effects of past experience with a problem as a barrier to novel innovation (Lovett & Anderson, 1996). Experience with one particular design approach assists in problem resolution by allowing the solver to see its applicability to the problem at hand (Saugstad, 1955; Staats, 1957). However, solvers pay a price for experience in problem-solving when the base solution paradigm is different in nature from the solution a solver has worked with in the past. The reason for this is that experience in problem solving has a tendency to produce attitudes and biases that favor the choice of problem solving strategies found successful in one instance to subsequent problems irrespective of the similarities or differences in approach to the one experienced earlier (Lovett and Anderson, 1996).

A number of researchers studying problem solving at the organizational level have argued that prior experience leads to a number of biases that block the organization from seeing more effective alternative problem solving approaches (March and Simon 1958; Nelson and Winter 1982). As demonstrated in the literature on evolutionary economics (Dosi, 1982; Nelson and Winter 1982), organization learning (Levitt & March, 1988), and technology management (Anderson & Tushman, 1990), the search for solutions to novel technological problems often involves a "local search" process. Thus, prior experience can cause innovators to localize their problem-solving search to familiar domains and thus cause difficulty when switching to a new technological approach or design path (Helfat 1982, Stuart and Podolny 1996, Sørensen and Stuart 2000). This effect will be negative for both external adoption experience and internal inventing experience:

> ***Hypothesis 3*** – A change in design path reduces the effect of invention experience on **a)** invention capacity and **b)** adoption capacity.
>
> ***Hypothesis 4*** – A change in design path reduces the effect of adoption experience on **a)** invention capacity and **b)** adoption capacity.

3 Methods

3.1 The MathWorks MATLAB Programming Contests

The last decade has seen significant interest in open and distributed innovation as an alternative institution governing innovation creation and disclosure. While there are many definitions of the phenomenon, we focus our attention on settings where individuals exert private effort in creating innovations and yet publicly disclose their creations for others to use. This institutional model of organizing the innovation process has historical precedents reaching as far back as the as the industrial revolution (Allen 1983; Nuvolari 2004; Osterloh and Rota 2007).

While scholarly interest in open-innovation, as exemplified by open source, has soared (von Hippel and von Krogh 2003), the notion of absorptive capacity has not been extensively investigated or integrated by this literature. This is surprising because the logic of open source depends on actors having the ability to access the knowledge of others and apply it to their own internal innovation needs. In a setting unconstrained by proprietary intellectual property concerns, absorptive capacity of individuals participating in the open-innovation context would appear to be a critical construct explaining performance. Thus adoption and inventing experience in open-innovation settings could help to explain performance differences among participants.

Programming contests sponsored by The MathWorks Corporation represent interesting examples of open source competitions (Gulley 2004). Held approximately every six months, these week-long, web-based contests challenge participants to develop software code in the MATLAB language to solve a complicated mathematical problem (e.g., the "traveling salesman" problem) in any of a range of domains including biology, supply chain management, mathematics, and physics. Participants are provided a detailed problem statement, a limited test-suite that enables them to privately evaluate their code writing efforts, and access to a Web interface by which to submit code to the contest scoring engine. The automated scoring engine, different from and more expansive than the limited test-suite available to the participants, evaluates each submission in terms of algorithmic accuracy (in the case of the traveling salesman problem, for example, minimizing travel distance) and computational efficiency (e.g., minimizing CPU execution time) and then generates an objective score for each entry.

The only way that participants can know the performance of their designs is to submit them and participants can submit code to be scored as often as they would like. After the 48th hour of the competition, the contest Web site maintains a dynamically updated leader board with scores and rankings of each code submission and associated author. The submission that earns the lowest score (in the case of the traveling salesman problem, minimum travel distance and execution time) at the end of the seven-day competition wins a nominal prize (typically a t-shirt).

A unique feature of these contests is a rule that dictates automatic information disclosure. A typical contest has three phases: dark, twilight, and light. Participants can join (or leave) the contest at any time, in any phase. During the first two days, authors cannot view the code of other contestants[4]. During the five-day "light" phase all participants are afforded access to all submitted code (the scores and ranks of which are at this time known). The code in another's submission can be tapped for insights or adopted in part or in entirety. Our study focuses in on the five-day light period where all code is freely available.

The MathWorks contest data provides a unique, laboratory-like setting to use objective measures to study invention, adoption and codebase switching. That contest participants are focused on solving the same innovation problem enables us to compare the performance of many individuals without worrying about task comparability. In addition, the presence of a common test suite creates an objective criteria for technical performance against which all code submissions can be compared. Focusing on the technical performance of many code submissions for the same problem enables us to overcome concerns about how to measure innovation performance in field settings. The setting is also relatively controlled in that we can observe the entry and exit of each participant.

We also are afforded fine-grained tracking of individual lines of code that enables us to precisely observe both invention and code adoption. For each submission by a contest participant/author, we have information on the origins of every single line of code and thus we can attribute it as a new invention (in the game) by the author, a newly copied line from another author, a line previously invented by the author and a line previously copied by the author in previous

[4] During the second day (hour 24 to 48), or the "twilight" period, participants can see the scores of others but not their code.

submissions. We can also deduce the technical parentage of each submission and allocate it to various alternative design paths in the contest.

Table 1 shows descriptive statistics for the nine games included in our sample. In total, our sample includes 875 unique gameXauthor combinations. In total, these authors submitted 23,532 entries, comprising over 4.7 million lines of code.

Include Table 1 about here

3.2 Empirical Strategy

To understand the development of invention and adoption capacity, we measure how experience is related to the “quality” of the lines of computer code invented or copied in each submission. In our setting, the “quality” of each line of code allows an accurate way to compare the ideas invented, copied, or reused by each author. Because we are interested in lines newly adopted or invented, we pay particular attention to the quality of these lines. We use a combination of dummy variables and interaction terms to separately identify how experience influences the quality of lines newly adopted or invented by the author. We use fixed effects for author and game to help remove the effect of differing author capabilities.

To estimate the quality of each line, we use a technique closely related to the Shapley value (Roth 1988, Shapley 1934). For each line, we compute the average marginal contribution of that line to all submissions in which it was included. This marginal contribution (the degree the line tends to improve or reduce the score of submission in which it is included) is our measure of quality.

In the next section, we provide more detailed information on how we compute our dependent and independent variables, and on how we conducted our statistical analysis.

3.3 Variables

3.3.1 Dependent Variable

Our dependent variable, *line quality*, is created by estimating the line’s average contribution to the each submission in which it is included.   Thus, our measure of *line quality* estimates the extent to which the inclusion of that line raised or lowered submissions in which it was included.

To calculate *line quality*, we regress a set of line dummies on the log score of each submission in that game. We chose to use log score instead of linear score because participants reported that improvement becomes more difficult as the scores approach the performance frontier.   For each of the nine games, we computed:

$$Y_s = \ln\left\{score_s - \min_{s=1,S}(score) + 1\right\} \qquad \text{Eq. 1}$$

where submission s is one of S submissions during the game. We then performed a simple OLS analysis of these scores:

$$Y_s = \mathbf{A}\mathrm{D}_\mathrm{t} + \mathbf{B}\mathrm{X}_\mathrm{L} + \mathrm{e}_{\mathrm{sL}} \qquad \mathrm{Eq.\,2}$$

where D_t is a set of hourly time dummies marking the time of submission s. X_L is a set of line dummies marking the use of line L in that submission s. The error term for line L in submission s is marked as e_{sL}. Each observation of a line is included in the estimate, so that the coefficient vector B contains the average contribution of that line to the score of the submissions in which it was used.

Our measure has the advantage of simplicity, but it assumes that the lines are independent and additive to the overall submission score. We chose to use this simple approach for several reasons. First, its interpretation and calculation is understandable and tractable. Second, this approach to measuring composite quality of individual lines is quite similar to calculating the well accepted “Shapley value”, or the fair allocation of credit or costs in situations involving multiple agents working cooperatively (Roth 1988; Shapley 1934).

3.3.2 Independent Variables

Invented is a dummy variable indicating if the author of the entry invented that line for the current submission. It takes a value of one is that is a unique line that is first appearing in the game. Otherwise it takes a value of zero. In other words, each line is only invented once, and only marked once as invented in our data.

Adopted is a dummy variable indicating that the line has been used by the author for the first time in the game and the author did not invent the line (i.e. he/she adopted it from someone else). It takes a value of one when the line is first adopted by the author. It takes a value of 0 otherwise. Each line that an author adopts is marked as *Adopted* only once per author (the first time it is used).

Invention Experience measures the extent to which the author has invented lines previously in a game. For each author, it aggregates *Invented* up to that moment in the game. In other words, if the author has invented 5 lines of code that were included in previous submission, *Invention Experience* will be the log function of five plus one.

Adoption Experience measures the extent to which the author has adopted lines from other players previously in this game. For each author, it aggregates *Adopted* up to that moment in the game.

Switch path indicates if the author has moved to a new software “path” with the current submission. A path is defined as a stream of entries that all trace their parentage to a previous original submission. The submissions in a common path all represent modifications of this original submission. When an author adopted lines a code-base from a stream of submissions that trace from a different original submission, then we say he/she switches paths. Figure 2 shows the scores from the top most used paths in one of the games we analyze. Note that paths do not last throughout the entire game, but paths do overlap in time. Note also that some of the paths initially begin far off the performance frontier and improve over time.

Parentage of submissions is determined in two ways. The large majority of the time, when an author chooses to switch to another code-base, he/she clicks a button to download the code-base. When an author does this prior to submitting, we assume that the next submission is based on this code-base. This method is used to determine code-base transfers for 81% of the transfers used in the analysis.

Based on interviews of codes and game organizers, we learned that in a few cases authors adopt entire code-bases by browsing the alternative code-base and then selecting and copying all of the code (e.g. by using the keystroke short cuts ctrl-A/ctrl-C/ctrl-V). These transfers do not leave the same historical trace as those above. To determine these transfers, we created an automatic discriminator. This discriminator first uses a probit model to predict actual reported code-base adoption based on the similarity of the overlap of code between the new submission and a previous submission. We then use the coefficients from this estimate and the probability scores it creates to estimate unmarked cases where an entire code-base was adopted. When an author reports no parent for their current code, we use our model to estimate if they actually have adopted another code-base. If our model says there is a greater than 90% chance that they have adopted a particular previous submission, then we mark this submission as the parent. The 90% cut-off is arbitrary and we conducted robustness tests using 1) differing thresholds and 2) only the actual reported parents (i.e. those cases where the authors followed the official downloading procedure).

Include Figure 2 about here

3.3.3 Control Variables

We include several control variables to better account for differences in submissions over time and to help to distinguish the effect of our results from other explanations.

Author's First Submission is a dummy variable marking the very first submission by the author in that game. It is coded as one (1) for the first submission and zero otherwise.

Time in Game is the log time in hours that the author has been playing in a game. The author's time begins upon his/her first submission. For the rest of the game, *Time in Game* is measured relative to this first submission.

Submission Experience is the log number of previous submissions (+1 to allow calculation for the first submission) by an author in the game up to the current time.

Novelcombo is a pair-wise count of lines of borrowed code that have previously not been together in any other prior entry by any other author. It is a measures of the extent to which the pattern of code borrowing by an author is novel.

Complexity utilizes McCabe's graph theoretic complexity measure of flow control based on the number of linearly independent execution paths in a program (McCabe 1976). This variable is generated automatically by MATLAB for every function in a software submission. We use the maximum function value reported as our measure of the complexity of a submission.

Lint captures the irregularity of the code. Johnson (1978) developed a program to alert programmers to potential errors in the construction of textual artifacts. A program may compile and run and still be poorly constructed, hampering performance and adoption. Our measure of *Lint* is a count of the number of Lint messages generated during the analysis of an entry by MATLAB. The lower the value of *Lint*, the less likely that there are errors in the submission.

3.4 Analytical Method

To test our hypotheses, we need to predict the quality of lines adopted or invented by the author under all contest conditions. We do this by estimating a regression model where the dependent variable is line quality and the independent variables are measures of the author actions and experience at that point in the game. We also include controls for the attributes of the submission. Because each game is different both in its characteristics and the rate at which solutions are uncovered, we include gameXtime fixed effects (i.e. time fixed effects for each game) to allow each game to have a unique improvement progression. We also include fixed effects for each gameXauthor. These dummy variables control for fixed differences in the author or in the match between their abilities and the game.

As discussed earlier, we perform our estimation at the line level (L). In other words we predict the quality of the line (Q_{gsl}) given the conditions (X) in the game g, submission s, time t, author a. As noted above, we include fixed effects for each combination of game and author $\mathrm{u_{ga}}$ and time effects for each game ∂_{gt}.

$$Q_{gsl} = \mathbf{B}\,\mathrm{X_{gas}} + \mathrm{u_{ga}} + \partial_{\mathrm{gt}} + \mathrm{e_L} \qquad \text{Eq. 5}$$

Because the use of lines are not fully independent observations, we cluster the standard errors for each line.

3.5 Descriptive Statistics

Table 2 shows the pooled descriptive statistics for all nine games. For the average submission, about 1% of the lines were newly invented for that submission. About 6% of the lines were newly adopted from another author for that submission. However, the number of adopted lines and inventions depend dramatically on whether or not the author is switching code-bases. Although they only invent about 2 lines per normal submission, and adopt another 3 new ones, authors tend to adopt 80 lines when switching paths and invent another 9. Each author's average adoption experience is 533 lines (max: 2100 lines). The average invention experience is 138 lines (max: 3684).

Not surprisingly, a number of variables are correlated. The time the author has been in the game, the number of entries, the number of adopted lines, and the number of inventions are all correlated. Interestingly, experience in invention is correlated with experience in adoption (R=0.55), but not enough to make it difficult to interpret the results of our analysis. *Author's First Submission* and *Adoption Experience* are correlated by construction (0.816). When an author makes his/her first submission, he/she has not yet adopted any lines. Fortunately, given the large size of the database there are sufficient observations to allow accurate coefficient estimations. Where appropriate, we conduct joint significance tests to insure that the 95% confidence interval for two correlated variables does not include zero.

Include Table 2 about here
----

4 Results

Table 3 shows the results of our regression analysis. Since we predict the quality of lines of three types (invented, copied, or reused), one must be careful in interpreting the coefficients. The base case is the lines that the author reuses in this submission (that is they had invented or adopted them in a previous submission). The coefficients for variables that include *invented* or *adopted* (either as main effects on in interactions) measure the quality of invented or adopted lines. Similarly, the inclusion of the variable *switch path* distinguishes lines when they are part of the first submission when an author switches to a new codebase.

Since our models also include gameXauthor fixed effects, we report the <u>within</u> R^2 for each model. This R^2 represents the percent of the author's changing performance (i.e. use of high quality lines) explained by the model variables. As we specify more complete models, the within R^2improves from 13% to 16%. Each model provides a significant improvement ($p < 0.001$) in explanatory power over the previous one. Model E explains the most variance, and we discuss it most completely below.

Our use of author and time fixed effects for each game also means that each author's data is "demeaned" in each game. As a result of these two factors, estimations should be interpreted as relative to the author and to the time in the game. In other words, our model predicts the quality of the author's lines of code relative to the use of code at a point in time and relative to that author's average use of code.

4.1 The Effect of our Control Variables

Considering first the control variables, all of the models reveal that authors tend to pick high quality code-bases to copy when they first join the game or when they switch paths. The coefficient estimation for *Author's First Submission* is positive and significant, and the effect for *Switch Path* is positive but not always statistically significant. This demonstrates that authors tend to choose relatively good code-bases when they first enter a game. Also, when switching paths, authors usually improve the quality of their code-base. This makes sense since authors are likely to pick leading design paths to join. As we will see later, however, this average result hides a more nuanced effect, because the effect of switching paths actually changes as authors gain more experience.

Interestingly, both *Time in Game* and *Cumulative Entry* have small negative coefficients. These must be interpreted carefully, however, because both time and cumulative entry are correlated with more adoption and invention experience. As a result, these coefficients become larger (and more significant) when the effect of other types of experience are included in more complete models. In total, the consistent sign and significant of the coefficients for *Time in Game* and *Cumulative Entry* suggest that authors perform poorly (i.e. their code-bases include inferior lines) when they simply are spending time in the game or submitting entries without many newly adopted lines or inventions.

Include Table 3 about here

4.2 The Effect of Inventions and Adoptions Average Codebase Quality

Turning now to our variables of interest: across all models the coefficient for the dummy variable *Invented* is significant and negative and the coefficient for *Adopted* is significant and positive[5]. This indicates that when an author invents a line, it usually has lower quality than the average line used in the game. In other words, if the average author only added this line to his/her code-base, it would actually hurt his/her relative score or ranking in the game. As odd as this seems on first analysis, it actually matches the most basic facts of invention: most new ideas are bad. The average new idea usually reduces performance (Fleming, 2001).

In contrast, when an author adopts a line, it usually has an above average quality. Again, this makes intuitive sense. Lines that are adopted have been used at least once before and the adopter can observe whether or not the line seems to have improved the score of the code-base of which it is a part. In other words, adopted ideas, because their effect can be estimated empirically before use (by evaluating the score of the submission in which they are a part) are usually of good quality.

The results with respect to the main effect of *Invention Experience* and *Adoption Experience* further illuminate our analysis. The coefficient for *Adoption Experience* is positive and significant. This suggests that the more an author adopted lines, the better become the lines in his/her code-base. There are two possible explanations for this. First, since most copied lines are high quality, there may be a direct additive effect of including more adopted lines. It is also possible that author's that experiment with more adopted lines are better able to select the best lines from among the ones he/she has tried out. If so, the author can gain more than just the mean contribution of the average copied line because after a trial he/she retains only the best ones.

[5] The non-included case is the lines that had previously been adopted or invented and are being reused.

In contrast, the coefficient capturing the main effect for *Invention Experience* is negative and significant. This suggests that, remarkably, the more inventions made by the author the worse the score of the average line in his/her code-base. Why the authors are not again able to select and retain only their useful inventions is not entirely clear. One possibility is that authors hang on to too many of their own inventions. In the average submission, 27% of the lines were invented by the author. Given that authors adopt lines at nearly 6.5 times the rate that they invent, the equilibrium number of invented lines in a submission should be only 13%. It appears then that locally invented lines build up in an author's submission over time. Given that locally invented lines are inferior, we may have found one explanation for the coefficient for *Invention Experience*: authors hold on to too many of their own ideas. In this paper, we can only speculate on why they do so. Perhaps they are emotionally attached to certain ideas. Perhaps they are rationally experimenting with ideas in the hope that they will provide a competitive advantage. In the future, we hope to explore these explanations.

4.3 Absorptive Capacity: Average Effects

We turn now to test our hypotheses of the effect of experience on invention and adoption capacity. Considering first the effect of invention experience, we find two significant coefficients in Model B. The coefficient for *InventionExperienceXInvented* is negative and significant while the coefficient for *InventionExperienceXAdopted* is positive and significant. To fully interpret these coefficients, however, one must consider the main effect of invention experience as well. The full effect of experience on invention quality requires adding this effect. When this is done, we find that the full effect of invention experience on invention is -0.051 - 0.112 = - 0.163. Thus invention experience actually reduces invention capacity! In contrast, invention experience has a mild improving effect on adoption capacity (-0.051 + 0.089 = 0.037).[6]

If we were to stop our analysis here, we would conclude that we have disproved H1a and supported H1b. That is, existing predictions are right about the effect of invention experience on adoption capacity, but wrong about the effect of invention experience on invention capacity! However, this analysis does not consider the effect of path switching and we will revisit this preliminary interpretation later in this section.

Turning now to the effect of adoption experience on absorptive capacity (H2a and 2b), we find in Model B that the coefficients for *AdoptionExperienceXInvented* and *AdoptionExperienceXAdopted* are both negative (though only one is significant). However, once again, care must be taken in interpreting these coefficients. One must also consider the very strong effect of adoption experience on the entire code-base (B for *AdoptionExperience* = 0.551). When this effect is included, adoption experience improves both the quality of the inventions and the adopted lines being made by the author. The net effect of Adoption Experience on invention capacity is 0.544 (i.e. 0.551 - 0.007), and the net effect of *AdoptionExperience* on the quality of adoption is 0.091 (i.e. 0.551 - 0.460). Thus we confirm both Hypothesis 2a and 2b,

[6] Note, as authors gain experience the quality of the reused code in their code base changes. Since our coefficients are measured relative to this base code, we need to account for these quality changes in calculating the quality of adopted or invented lines. If one wanted to measure instead the quality of invented and adopted lines relative to the reused code, one would look instead at the raw coefficients.

adoption experience increases invention and adoption capacity. We show however, that when considering the effect of incremental experience (adoption or invention of a line of code), experience in adoption has the dominant effect on invention capacity.

4.4 The Effect of Path Switching on Absorptive Capacity

In the above discussion (4.3), we ignored the effect of path switching. Because the analysis we considered had not separated out the effect of path switching, our coefficients suggested that innovation experience harmed innovation capacity. This result aggregated the effect of invention experience when a) authors continue on a given design path and b) when they switch paths. We now complete our unpacking of absorptive capacity (as shown in Figure 1) by separating 8 cases: 2 types of experience (adoption and invention) influence two outcomes (adoption and invention quality) under two conditions (path continuance and patch switching). Model E measures all of these cases. Figure 3 graphs the results.

As shown in Figure 3a, invention experience improves both adoption and invention capacity, but the value of invention experience on invention is lost, indeed reversed, during a change in design path. More invention experience improves the quality of inventions made when the author continues on a path. However, this experience actually harms the quality of inventions made when switching paths.

As shown in Figure 3b, adoption experience improves both adoption and invention capacity, whether or not the author switches path (see Figure 3b). However, the effect of adoption experience on invention capacity is reduced during a path switch.

In total, our results suggest that a switch in design path harms invention capacity. Authors are less able to apply experience gained from adopting code to create quality inventions. Experience gained in previous inventions actually harms their ability to invent. This remarkable result suggests that the more authors have engaged in experience on previous paths, the more difficulty they have in inventing on a new path – at least initially.

With respect to our hypotheses, our results confirm Hypothesis 3b and 4B. That is switching harms the effect of experience (be it from adoption or invention) on invention quality. When authors switch paths, experience from adoption or invention is either less useful in supporting invention or actually counterproductive. We are unable to confirm hypotheses 3a and 4a. We find no evidence that experience gained inventing or adoption impedes authors from divining useful elements to adopt into a new design path.

Include Figure 3 about here

What might explain these intriguing results? What might cause invention experience to actually harm author's absorptive capacity during a path switch. One explanation is that authors get accustomed to particular ways of thinking that guide their attempts at improvement. The more they have been practicing this thinking, as evidenced by their innovation activity, the more their thinking has become fixed. Thus, when they switch to a new code-base with a different plan, they have difficulty adapting their innovative direction immediately.

In contrast, adoption experience may help authors scan their surroundings for alternative ideas. In doing so, they stay mentally flexible to alternative technological approaches. Thus, while some of the benefit of adoption experience is lost during a path switch, such experience continues to improve invention capacity – even during a change in design path.

Table 4 shows a summary of our results. We find conditional support for H1a: when paths are continued, we confirm that invention experience increases invention capacity. We support H1b: invention experience increases adoption capacity. We also confirm adoption experience increases both invention (H2a) and adoption capacity (H2b). With respect to the effect of path switching, we find that switching paths damages the effect of prior experience on invention capacity, whether that prior experience comes from previous invention (H3a) or adoption (H4a). We find no evidence that switching paths harms the effect of prior experience on adoption capacity. Thus we fail to confirm H3b and 4b.

Include Table 4 about here

4.5 The "Average" Development of Adoption and Invention Capacity

The multifaceted nature of our analysis makes it difficult to assess how an author's absorptive capacity develops during an average contest. To get a sense of this, we estimated the performance of an average author over time. Figure 4 provides an estimate for an author that entered the game at hour 48, submitted an average number of entries (13) before the end of the game, and switched design paths approximately the average number of times (3). This author also adopted and invented the average number of new lines. The 0 value on the Y axis represents a line of average quality. We provide data on the quality of the lines invented or adopted at each submission (including the three times the author submitted on a new path).

Include Figure 4 about here

As can be seen clearly, authors adopt lines that are better than the average available line and their skill in doing this improves slightly over time. Lines adopted with paths are on average better than those adopted separately and this trend continues throughout the contest[8]. The average invention, in contrast, is initially very inferior to the average line. However, the authors rapidly learn from experience how to invent, so that inventions (made when not switching paths)

[8] The initially adopted lines are relatively the best – presumably because authors join the context by adopting the current leader.

improve dramatically. This experience has much less of an effect on invention quality when authors switch paths.

4.6 The Net Effect on the Quality of the Code-base

So far, we have considered the effect of experience on adoption and invention capacity. We have not yet evaluated the effect of invention and adoption experience on the performance of the author's submissions. Such performance is a function of the quality of all of the code in a submission (adopted, invented, and retained). To understand how an average author's performance changes during a contest, we estimated the average quality of a line in the code-base for the same average author as that analyzed in Figure 4.

Figure 4 shows that the author's first submission tends to include code with above average quality. This is not surprising, since entering authors can choose among the better pre-existing submissions to get started. However, author's initial attempts to improve these submissions are not positive. Because authors have little or no prior experience, they invent poor quality lines and thus drag down the performance of the code-base. Eventually, after several failed attempts to improve the current path, that author switches to a new path. When he/she does so, their score degrades further because temporarily the author loses the value of the invention experience they have gained, and the invented lines they add to this new path actually damage its performance. However, the switch in paths brings with it a turning point for the author. The path switch brings with it a large influx of new ideas (new lines of code). This experience provides authors with a new basis for learning that increases their capacity to invent and adopt valuable lines and sets them on a path of improvement. Improvement is slow at first, but the next path switch further accelerates this process. The authors again gain a pulse of experience with alternative ideas and this further improves their ability to invent and adopt.

Include Figure 5 about here

Thus, our analysis suggests that authors face a complicated tradeoff decision in managing their absorptive capacity. To gain the necessary experience to be able to invent beneficial lines of code, they must make wholesale changes in their code-base by switching to new design paths. However, doing so temporarily reduces their ability to create quality inventions. They must stick to a path in order to be able to make better inventions, but in doing so, their experience with alternative ideas (lines of code) lags. They must jump to new paths in order to be able to spark their future invention potential, but in the short term this jump actually reduces their ability to invent.

5 Discussion

5.1 Implications for Absorptive Capacity

In this paper we further unpack the theory and phenomenon of Absorptive Capacity by considering the effect of two kinds of experience (adoption and invention) on two kinds of innovative improvement (adoption of external ideas and internal invention). We also consider how switches in design paths moderate both effects.

In general, we find evidence supporting the main claims of absorptive capacity, but we also are able to suggest some refinements. We do find that both invention and adoption experience generally both increase invention capacity and adoption capacity. But we find that the size of the effects varies. Adoption experience dramatically improves invention capacity. In our context, to be a good inventor, a designer simply must copy. We infer that this is because adoption provides stimulating alternative ideas that spark new invention.

We also find that switching has a dramatic effect on invention capacity. It reduces the quality of inventions made on a new path and it reduces the effect of prior invention experience. In fact, it reverses it – creating not invention capacity but invention **INcapacity**. We infer that this is because authors have difficulty switching their creative direction and mental models to fit the new design path. They are stuck in old ways of thinking and need some time with the new technological approach before they can again invent effectively. This is consistent with Allen's research showing that engineers designing NASA technologies were often unable to switch to new approaches – even once they were thought to be better (Allen, 1984).

Our research shows that absorptive capacity can be acquired in bundles. When an author switches paths, they gain a large amount of experience with new ideas. These new ideas then help them to invent and adopt in the future. Thus, path switching includes both costs and benefits. It provides long-term improvement by increasing invention and adoption capacity. But it harms invention capacity in the short term.

Our two results thus contain a difficult management tension. To learn to invent, a designer must adopt and the best way to do so is to adopt an entire approach (or design path in our terminology). However, such adoption temporarily damages invention capacity. Thus, learning to invent requires a difficult leap into the unknown.

We hope that future research will extend our analysis in a number of ways. Most importantly, our research reveals the importance of something we have begun to call "selection capacity". One of the critical findings of our study is that invented lines are usually inferior and adopted lines are only slightly better than average. For an author near the performance frontier, adding in these ideas will only harm their performance. We show, however, that for adopted lines the opposite is true: more adoption leads to better overall quality. We infer that this is because authors quickly remove the worst performing lines from their code-base. In future research, we hope to investigate how authors gain the capacity to select and retain only the best ideas.

5.2 Limitations of the Study

Our research reveals the value of further unpacking the nature of absorptive capacity, but it contains many limitations which should suggest caution in making too broad extrapolations. First, we consider each line as if it is independent from others. That is it contributes to the overall score of the entry as if it simply added to a pooled score. It may be that some lines will contribute more in combination with others. Lines may come as part of functioning modules that must be used together. And over time modules may be getting adopted instead of lines. However, we don't think that inclusion of a line in a module would bias our quality scores, so we think that our approach to isolate each line's net effect on the score provides a meaningful way to analyze individual contributions. Adoption of modules rather than lines would influence the independence of our observations and might bias our standard errors. We have tried various methods to test the robustness of our results, including bootstrapping techniques, and find that our results are robust.

Second, we do not consider the effect of strategy on the quality of the lines used by an author. It is possible that some authors choose to use parts of their overall knowledge in any given submission so as to deliberately avoid getting the high score and thus draw the attention of others. Based on interviews of participants and managers, we think this behavior was rare. Reportedly, authors desired to be able to claim the leading position at any point in the game.

The actual game also limits the generality of our findings. We do include nine different contests in our analysis, but all of them had a similar rule structure. Thus, authors in contests with different rules might learn differently. Two rules seem particularly important to consider. First, the game provided complete transparency after the 48th hour. Authors could simply download another's code-base and inspect it. In other contexts, they might receive only an imperfect signal about other's ideas. In other settings, one of the features of absorptive capacity might be the ability to break through another's protective veil. This veil was not present in our setting.

A second rule that might limit the generality of our results was the implicit reward system in the competition. Officially, each game had only one winner, so authors may have adjusted their behavior with this in mind. They might, for example, have been more willing to try risky inventions because they wanted to see if they could boost their scores to the front. In future research we hope to change the incentive structure in some of the games to see if it affects behavior.

5.3 Implications for Open-innovation

For the broader scholarly and practical community interested in innovation, our research provides new insight on adoption and invention in open-innovation settings. Most critically, it reveals how important such open-innovation is for successful innovation. Our analysis suggests that the single most important way individuals learn how to invent is by adoption the approach of others. In doing so they gain a jolt of new ideas and experience that sets them on an improvement path. Had they been unable to copy, our model suggests that on average they would have been unable to make only marginal improvements on the best submission created during the “dark period” (before hour 48) in the game. In other words, most or all of the improvement in the programs achieved from hour 48 to 175 can be attributed to the change from a closed to open-innovation regime.

Our research also shows, however, that authors (or their managers) working in an open-innovation environment face a difficult decision when considering whether to switch to another’s technological approach. Switching harms invention capacity, at least temporarily, and for the first path switch the performance of the overall design suffers as a result. Thus, authors and managers must be willing to make a costly leap into the unknown in the hope that it will improve their performance in the long term.

Innovation too requires courage. Inventions are usually harmful, so authors must be tempted to simply adopt the best code-base and make few modifications. Yet, invention provides a benefit to the author’s capacity to adopt and invent in the future. Again, authors and managers must have the courage or foresight to knowingly make changes they expect to be harmful so that improvement will be possible. This also addresses the free-riding concern that is present in open-innovation settings as it shows that self-investment in invention is necessary to gain the benefits of the spillovers of others.

References

Alchain, A. A. 1950. “Uncertainty, evolution and economic theory.” *Journal of Political Economy* 58(6):211–221

Allen, R.C. 1983. Collective Invention. *Journal of Economic Behavior and Organization* **4**(1) 1-24.

Allen, T. J. 1984. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization*. Cambridge, MA: MIT Press.

Anderson, P., M. L. Tushman. 1990. Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change. *Administrative Science Quarterly* **35** (4) 604-633.

Anderson, JR., R. Farrell, R. Sauers. 1984. Learning to program in LISP. *Cognitive Science* **8**:87-129.

Arora, A., A. Gambardella. 1994. Evaluating technological information and utilizing it: Scientific knowledge, technological capability, and external linkages in biotechnology. *Journal of Economic Behavior & Organization* **24** 91-114.

Argote, Linda. 1999. Organizational Learning: Creating, Retaining & Transferring Knowledge. 1st ed. Springer.

Baron, Jonathan. 2007. Thinking and Deciding. 4th ed. Cambridge University Press.

Benkler, Y. 2006. The wealth of networks: how social production transforms markets and freedom. Yale University Press.

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

Cassiman, B., R. Veugelers. 2006. In Search of Complementarity in Innovation Strategy: Internal R&D and External Knowledge Acquisition.. *Management Science* **52** 68-82.

Chase, W.G, H.A. Simon. 1973. Skill in chess. *American Scientist* **61**:394-403.

Clark, K.B. 1985. The interaction of design hierarchies and market concepts in technological evolution. *Research Policy* **14** 235-251.

Cockburn, I, R Henderson. 1998. Absorptive capacity, coauthoring behavior, and the organization of research in drug discovery. *The Journal of Industrial Economics.*

Cohen, W.M., D.A. Levinthal. 1989. Innovation and Learning: The Two Faces of R & D. *The Economic Journal* **99** 569-596.

Cohen, W.M., D.A. Levinthal. 1990. Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly* **35** 128-152.

Cohen, W.M., D.A. Levinthal. 1994. Fortune Favors the Prepared Firm. *Management Science* **40** 227-251.

Csaszar, F.A., N. Siggelkow. 2010. How Much to Copy? Determinants of Effective Imitation Breadth. *Organization Science* **21** 661-676.

Dosi, G. 1982. Technological paradigms and technological trajectories. *Research Policy* **11** 147-162.

Ellis, H.C. 1967. The Transfer of Learning. *The Macmillan Company.*

Ericsson, K.A. 2006. The influence of experience and deliberate practice on the development of superior expert performance. *The Cambridge Handbook of Expertise and Expert Performance* 683-704, edited by K. Anders Ericsson, N Charness, Paul J. Feltovich, R. R. Hoffman. Cambridge University Press, Cambridge, UK.

Escribano, A., A. Fosfuri, J.A. Tribó. 2009. Managing external knowledge flows: The moderating role of absorptive capacity. *Research Policy* **38** 96-105.

Fleming, L. 2001. Recombinant uncertainty in technological search. *Management Science* **47** 117-132.

Fernandes, R, and H Simon. 1999. “A study of how individuals solve complex and ill-structured problems.” *Policy Sciences* 32 225-245.

Gioia, D., C.C. Manz. 1985. Linking Cognition and Behavior: A Script Processing Interpretation of Vicarious Learning. *The Academy of Management Review* **10** 527-539.

Gordon, W. J. 1961. Synectics: The development of creative capacity. Harper and Row, New York.

Gulley, N. 2004. In praise of tweaking: a wiki-like programming contest. *interactions* **11** 18–23.

de Groot, A.D. 1965. Thought and Choice in Chess. *Amsterdam University Press.*

Haefliger, S, G Von Krogh, S Spaeth. 2008. Code Reuse in Open Source Software. *Management Science* **54** 180-193.
Haleblian, J., S. Finkelstein. 1999. The Influence of Organizational Acquisition Experience on Acquisition Performance: A Behavioral Learning Perspective. *Administrative Science Quarterly* **44** (1):29–31.
Harlow, H.F. 1949. The formation of learning sets. *Psychological Review* **56** 51-65.
Hilgard, E.R., G.H. Bower. 1975. Theories of learning. Prentice-Hall.
Hong, Lu, Scott E. Page. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America* **101** (46):16385 -16389.
Hoang, H., F.T. Rothaermel. 2010. Leveraging internal and external experience: exploration, exploitation, and R&D project performance. *Strategic Management Journal* **31** 734-758.
Jeppesen, L.B., K.R.Lakhani. 2010 "Marginality and Problem-Solving Effectiveness in Broadcast Search." *Organization Science* **21** 1016-1033.
Johnson, S. 1978. Lint, a C program checker. *Computer Science Technical Report*.
Jonassen, David H. 2003. Learning to Solve Problems: An Instructional Design Guide. 1st ed. Pfeiffer.
Katila, R, G Ahuja. 2002. Something old, something new: A longitudinal study of search behavior and new product introduction. *Academy of Management Journal*.
Klahr, D., H.A. Simon. 2001. What Have Psychologists (And Others) Discovered About the Process of Scientific Discovery? *Current Directions in Psychological Science* **10** 75-79.
Lakhani, K.R., E. von Hippel. 2003. How Open Source Software Works: Free User to User Assistance. *Research Policy* **32** 923-943.
Lane, P.J., R.K. Balaji, S. Pathak. 2006. The Reification of Absorptive Capacity: A Critical Review and Rejuvenation of the Construct. *Academy of Management Review* **31** 883-863.
Lee, J., H.E. Posen, S. Yi. 2010. The Power of Imperfect Imitation. Atlanta Competitive Advantage Conference 2010 Paper. Available at SSRN: http://ssrn.com/abstract=1553350
Lerner, J., J. Tirole. 2002. Some Simple Economics of Open Source. *Journal of Industrial Economics* **50** 197-234.
Lewin, A. Y, and S. Massini. 2003. Knowledge creation and organizational capabilities of innovating and imitating firms. *Organizations as knowledge systems: Knowledge, learning and dynamic capabilities* 209–237.
Levitt, B., J. G. March. 1988. Organizational learning. *Annual Review of Sociology* **14** 390-340.
Lovett, M. C., J. R. Anderson. 1996. History of Success and Current Context in Problem Solving: Combined Influences on Operator Selection. *Cognitive Psychology* **31** (2) 168-217.
Luchins, A. S. 1942. Mechanization in problem solving: the effect of Einstellung. *Psychology Monographs* **54** (248).
March, J. G., H.A. Simon. 1958. *Organizations*. Wiley.
Martin, X., W. Mitchell. 1998. The influence of local search and performance heuristics on new design introduction in a new product market. *Research Policy* **26** 753-771.
Mazur, J. E, R. Hastie. 1978. Learning as accumulation: A reexamination of the learning curve. *Psychological Bulletin* **85** (6):1256–1274.
McCabe, T. 1976. A complexity measure. *IEEE Transactions on Software Engineering* **SE-2**(4) 308-320.
Murray, F., P. Aghion, M. Dewatripont, J. Kolev, S. Stern. 2010 Of Mice and Academics: Examining the Effect of Openness on Innovation. NBER Working Paper #14819.
Murray, F., S. O'Mahony. 2007. Exploring the Foundations of Cumulative Innovation: Implications for Organization Science. *Organization Science* **18** (2) 1006-1021.
Murmann, J.P., M.L. Tushman. 1998. Dominant Designs, Technology Cycles and Organizational Outcomes. *Research in Organizational Behavior* **20**.
Nelson, R.R., S.G. Winter. 1982. An evolutionary theory of economic change. Belknap Harvard, Cambridge, MA.

Nuvolari, A. 2004. Collective invention during the British Industrial Revolution: the case of the Cornish pumping engine. *Cambridge Journal of Economics* **28** (3) 347-363.

Osterloh, M., S. Rota. 2007. Open source software development—Just another case of collective invention? *Research Policy*.

Rothaermel, F.T., M.T. Alexandre. 2009. Ambidexterity in Technology Sourcing: The Moderating Role of Absorptive Capacity. *Organization Science* **20** 759-780.

Roth, Alvin E. 2005. The Shapley Value: Essays in Honor of Lloyd S. Shapley. *Cambridge University Press.*

Saugstad, P. 1955. Problem-solving as dependent upon availability of functions. *British Journal of Psychology* **46** 191-198.

Shapley L.S. 1953. A Value for *n*-person Games. *Contributions to the Theory of Games*, volume II, by H.W. Kuhn and A.W. Tucker, editors. *Annals of Mathematical Studies* **28** 307–317. Princeton University Press.

Simon, H.A. 1973. Does Scientific Discovery Have a Logic? *Philosophy of Science* **40**:471-480.

Sorensen, J. B., T. E. Stuart. 2000. Aging, Obsolescence, and Organizational Innovation. *Administrative Science Quarterly* **45** (1) 81-112.

Staats, A. W. 1957. Verbal and instrumental repose-hierarchies and their relationship to problem-solving. *American Journal of Psychology* **70** 442-446.

Stuart, T. E., J. M. Podolny. 1996. Local search and the evolution of technological capabilities. *Strategic Management Journal* **17** 21-38.

Suarez, F, J Utterback. 1995. Dominant designs and the survival of firms. *Strategic Management Journal* **16** 415-430.

Terwiesch, C, Y Xu. 2008. Innovation Contests, Open-innovation, and Multiagent Problem Solving. *Management Science* **59** 1529-1543

Utterback, J. M., F. F. Suarez. 1993. Patterns of industrial evolution, dominant designs and firms's survival. *Research on Technological Innovation, Management and Policy* **5** 47-87.

Vincenti, W.G. 1993. What Engineers Know and How They Know It: Analytical Studies from Aeronautical History. *The Johns Hopkins University Press.*

Vincenti, W.G. 1994a. The Retractable Airplane Landing Gear and the Northrop "Anomaly": Variation-Selection and the Shaping of Technology. *Technology and Culture* **35** 1-33.

Vincenti, W.G. 1994b. Variation--selection in the innovation of the retractable airplane landing gear: the Northrop ‘anomaly'. *Research Policy* **23** 575-582.

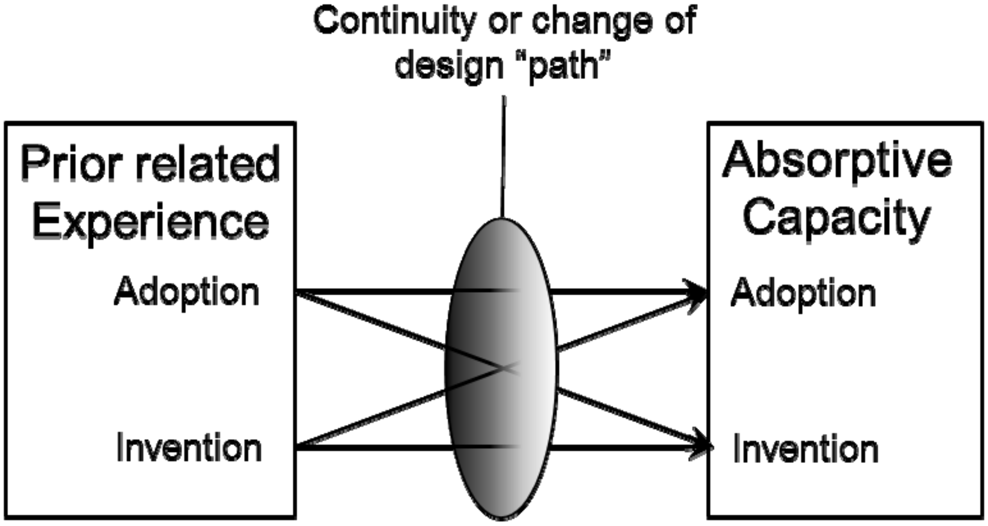
Volberda, H.W., N. J. Foss, M.A. Lyles. 2010. Absorbing the Concept of Absorptive Capacity: How to Realize Its Potential in the Organization Field. *Organization Science* **21** 931-951.

von Hippel, E. 2005. Democratizing Innovation. MIT Press, Cambridge, MA:.

von Hippel, E., G. von Krogh. 2003. Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science. *Organization Science* **14** (12) 209-223.

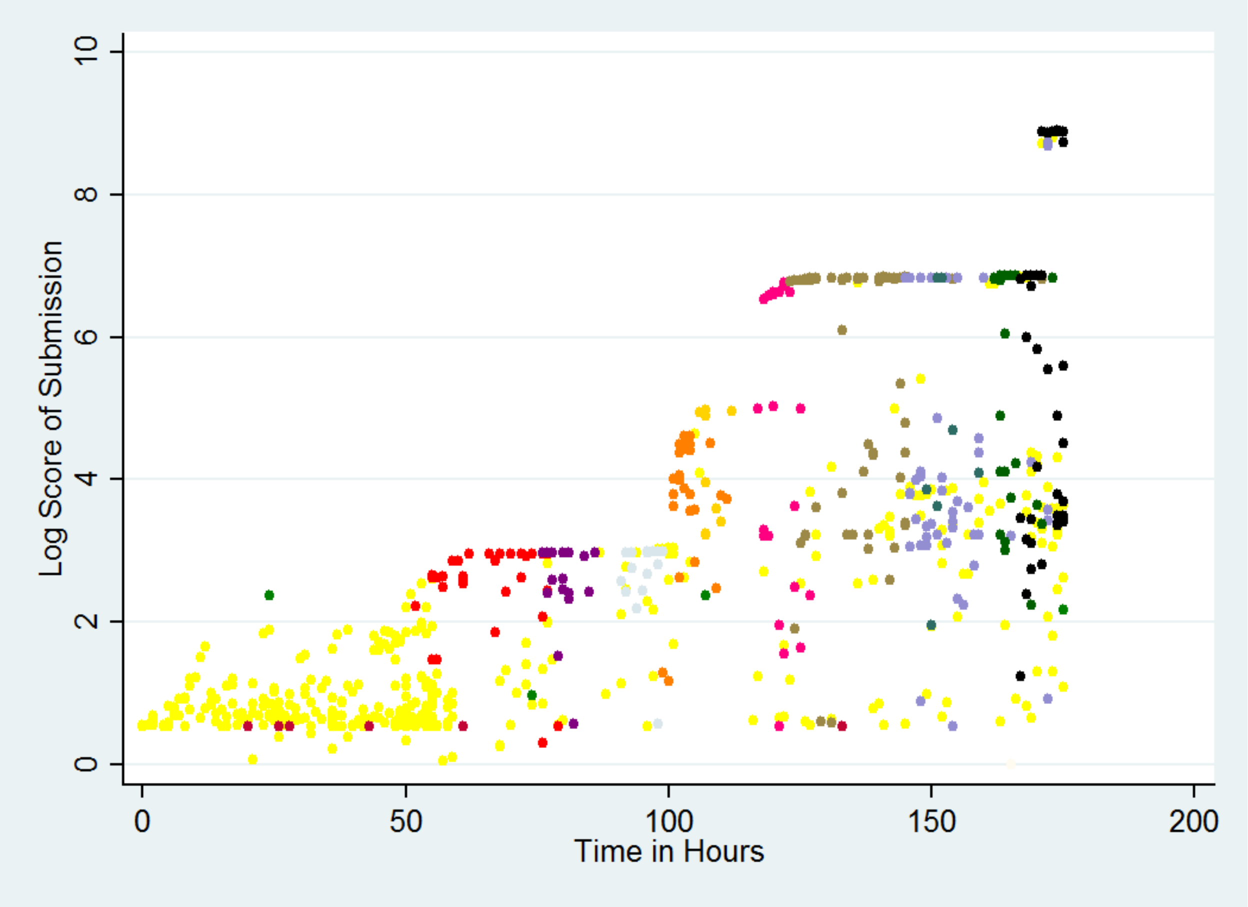
Zahra, S.A, G. George. 2002. Absorptive capacity: A review, reconceptualization, and extension. *Academy of Management Review* **27** 185-204.

Figure 1: Unpacking Absorptive Capacity



We hypothesize that experience with adoption and invention influences both invention and adoption, and we propose that the effect of experience is reduced when solvers switch design paths (i.e. adopt a new codebase as the basis for development).

Figure 2: Design paths in an example Game



Note: Major paths shown in different colors. Minor paths shown in yellow. Before hour 48 (the "dark" period) code is not visible to other authors, so only the patch created by the original (and visible) seeding submission are observed.. Two major paths (green and red) that began during the dark period and then gained prominance later are evident.

Figure 3a and b: Effect of Experience on Absorptive Capacity

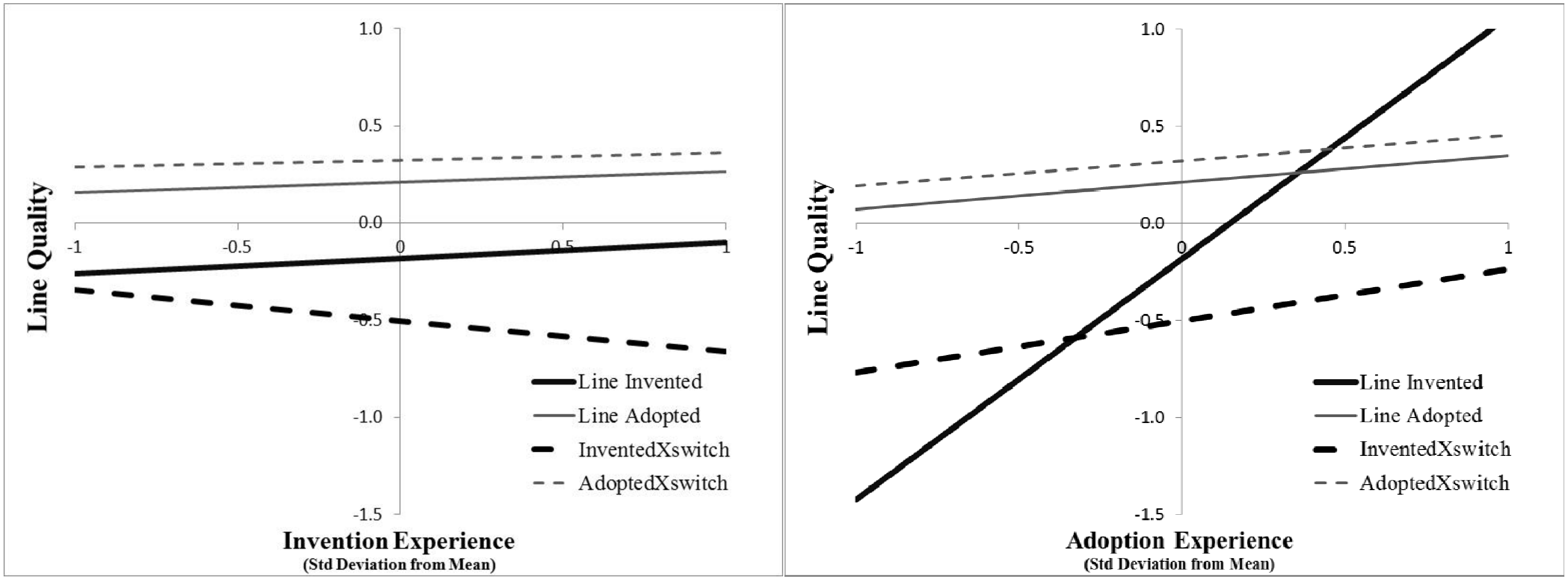
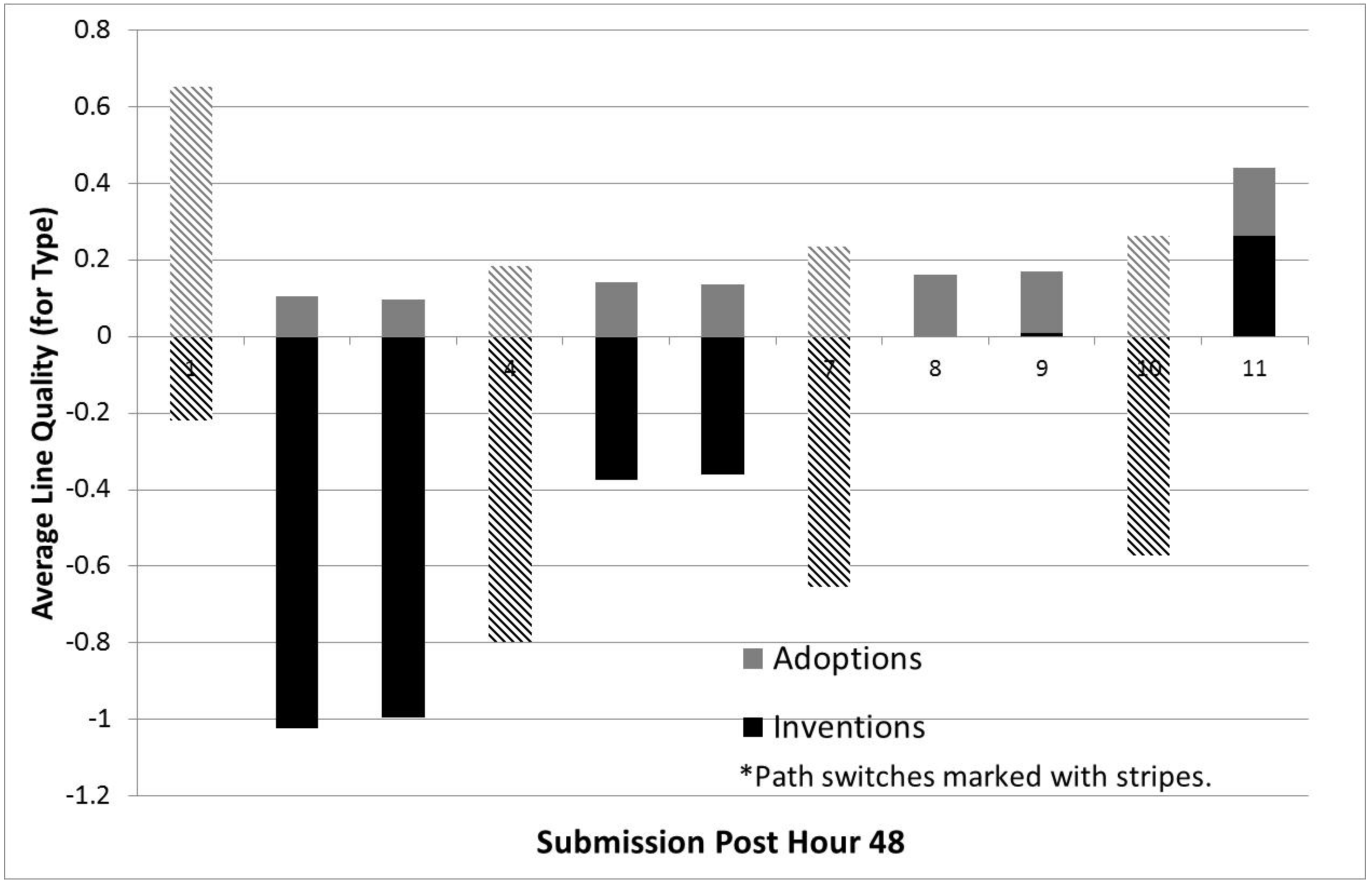


Figure 3a: Invention experience improves the ability of authors to adopt and invent. However, when authors switch "paths", previous invention experience actually reduces invention quality on the new path.

Figure 3b: Adoption experience improves the ability of authors to adopt and invent. Adoption experience strongly influences invention quality. Switching paths reduces the effect of previous adoption experience, but does not reverse the direction.

Figure 4: Estimated in Quality of Absorbed Lines over Time



Alternative

Figure 4: After authors can view each other’s code (after hour 48), they tend to adopt lines with above average quality. In contrast, invented lines are initially of below average quality (and thus harmful to performance). Eventually, near the end of the contest, authors with average experience tend to invent valuable new lines. However, lines invented when author switches paths remain inferior.

Table 5: Estimated Quality of Code-base Over Time for an Average Author

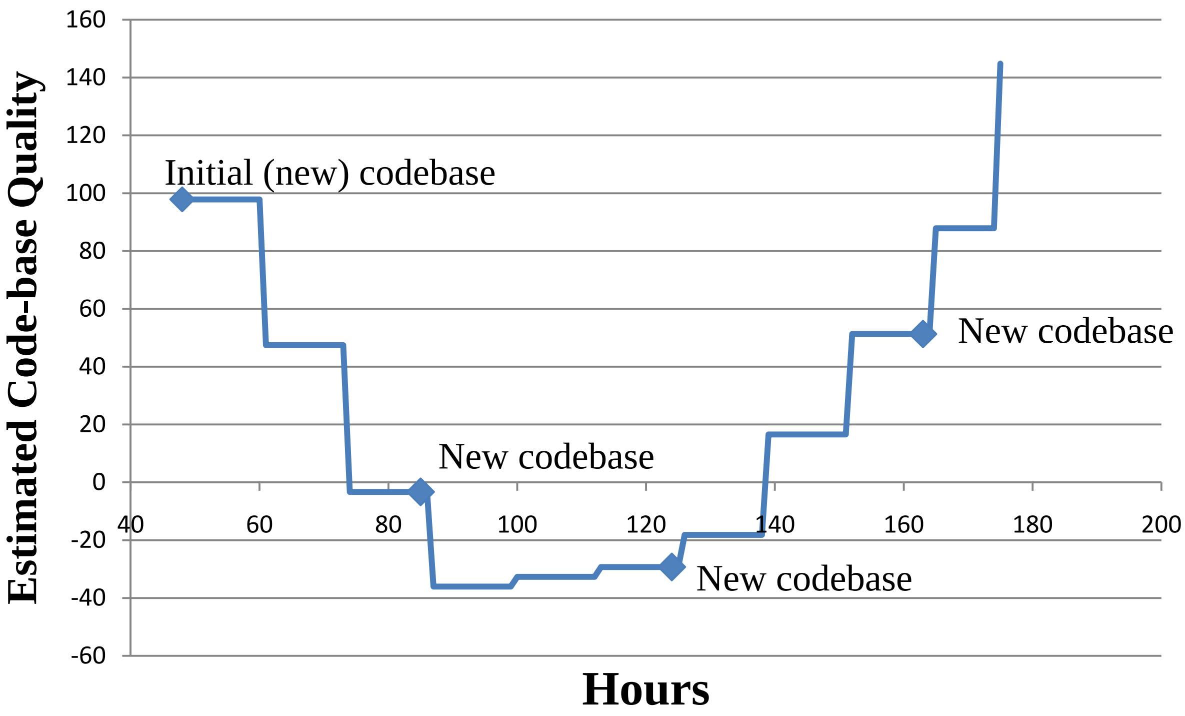


Figure 4: The average author tends to enter the game by adopting the leading code base. As a result, they tend to start out above average and then fall in performance over time. This is because they tend to add inventions which are actually counterproductive. Eventually, the authors jump to a new codebase. They make marginal improvements to this codebase before jumping again to another. Finally, With this third codebase, the authors begin to make meaningful improvements.

Table 1: Descriptive Statistics for Nine MathWorks Games Used in Analysis

			Submissions/Author		Lines /Submission	
Name of Contest	Number of submissions	Number of Authors	Mean	SD	Mean	SD
Molecule	1631	153	10.66	27.03	76.26	49.95
Protein Folding	2437	199	12.24	28.12	381.95	363.69
Trucking Logistics	1659	145	11.45	28.75	264.75	158.65
Election Gerrymander	2392	157	15.23	34.78	407.17	255.29
Furniture Moving	1834	94	19.51	41.75	717.52	442.55
Ants	2206	153	14.41	37.84	106.02	45.15
Sudoku	3061	168	18.22	41.65	197.34	108.65
Blockbuster	4420	168	26.31	111.18	119.85	46.62
Peg Jumping	3912	116	33.74	91.81	221.24	128.31

Table 2: Descriptive Statistics for Analysis of MathWorks Competitions

		Mean	SD	Min	Max	1	2	3	4	5	6	7	8	9	10	11	12
1	Line Quality	0.050	0.924	41.586	13.466	1											
2	Invented	0.009	0.095	0.000	1.000	-0.152	1										
3	Adopted	0.059	0.236	0.000	1.000	0.005	-0.024	1									
5	Invention Experience	0.055	1.632	-5.407	3.259	0.019	-0.032	-0.385	1								
4	Adoption Experience	0.212	1.307	-6.744	1.799	0.221	-0.121	-0.647	0.550	1							
6	Authors First Submission	0.028	0.164	0.000	1.000	-0.022	0.063	0.641	-0.816	-0.497	1						
7	Switch Path	0.158	0.364	0.000	1.000	-0.037	0.075	0.481	-0.400	-0.246	0.390	1					
8	Time In Game	89.057	52.152	0.000	175.484	0.096	-0.041	-0.208	0.473	0.659	-0.288	-0.100	1				
9	Cumulative entry	3.875	1.332	0.693	7.222	0.111	-0.066	-0.372	0.598	0.780	-0.403	-0.261	0.584	1			
10	Complexity	60.821	55.703	1.000	669.000	0.046	-0.033	0.019	0.037	0.020	0.032	0.064	0.130	-0.113	1		
11	Lint	3.249	1.284	0.000	6.358	0.000	-0.027	0.035	-0.061	-0.004	0.040	0.036	-0.070	-0.170	0.618	1	
12	Novel combination	1.433	2.371	0.000	12.171	-0.031	0.056	0.136	-0.093	-0.005	0.046	0.198	0.014	-0.116	0.090	0.111	1

Table 3: Analysis of Program Line Quality in Mathworks Competitions

	A		B		C		D		E	
	coef	se	coef	se	coef	se	coef	se	coef	se
Invented	-0.608***	0.015	-0.380***	0.013	-0.479***	0.017	-0.451***	0.017	-0.371***	0.017
Adopted	0.337***	0.010	0.265***	0.008	0.199***	0.010	0.228***	0.011	0.197***	0.011
Invention Experience	-0.035***	0.002	-0.051***	0.003	-0.049***	0.003	-0.045***	0.003	-0.036***	0.003
Adoption Experience	0.293***	0.010	0.551***	0.019	0.553***	0.019	0.592***	0.021	0.593***	0.021
Invention ExperienceXInvented			-0.112***	0.010	-0.098***	0.010	-0.083***	0.010	0.085***	0.016
Invention ExperienceXAdopted			0.089***	0.003	0.090***	0.003	0.094***	0.003	0.068***	0.005
Adoption ExperienceXInvented			-0.007	0.015	0.007	0.015	0.031*	0.015	0.297***	0.020
Adoption ExperienceXAdopted			-0.460***	0.016	-0.458***	0.016	-0.408***	0.014	-0.528***	0.016
Switch PathXInvention					0.293***	0.030	0.180***	0.029	-0.232***	0.028
Switch PathXAdopted					0.106***	0.009	0.029**	0.010	0.073***	0.010
Invention ExperienceXSwitch							-0.013***	0.001	-0.012***	0.001
Adoption ExperienceXSwitch							-0.109***	0.007	-0.074***	0.008
Invention ExperienceXSwitchXInvented									-0.134***	0.018
Invention ExperienceXSwitchXAdopted									0.003	0.005
Adoption ExperienceXSwitchXInvented									-0.492***	0.018
Adoption ExperienceXSwitchXAdopted									0.080***	0.009
Authors First Submission	1.412***	0.054	0.654***	0.027	0.654***	0.027	0.496***	0.021	0.357***	0.021
Switch Path	0.004	0.002	0.016***	0.002	0.004	0.002	0.051***	0.005	0.038***	0.005
Time In Game	-0.003***	0.001	-0.005***	0.001	-0.005***	0.001	-0.006***	0.001	-0.006***	0.001
Cumulative entry	-0.004	0.003	-0.028***	0.004	-0.029***	0.004	-0.033***	0.004	-0.041***	0.004
Complexity	0.002***	0.000	0.002***	0.000	0.002***	0.000	0.002***	0.000	0.002***	0.000
Lint	0.079***	0.011	0.069***	0.011	0.068***	0.011	0.070***	0.011	0.066***	0.011
Novel combination	0.001**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Number of Observations	4735107		4735107		4735107		4735107		4735107	
Number of Groups (AuthorsXGames)	875		875		875		875		875	
F	2521.56***		2976.40***		2965.95***		2986.99***		3057.06***	
R^2 (within)	0.132		0.154		0.155		0.156		0.161	
Chi Square (rel previous model)			121674.11***		2454.41***		10235***		27619***	

note: *** p<0.001, ** p<0.01. * p< 0.05 . All models include fixed effects for gameXtime and gameXauthor.

Table 4: Summary of Our Findings

Hypothesis	Result
H1– Experience with internal invention will lead to greater a) invention capacity, and	Confirmed conditional on path continuance.
b) adoption capacity.	Confirmed
H2– Experience with adoption will lead to greater a) invention capacity, and	Confirmed*
b) adoption capacity.	Confirmed
H3– A change in design path reduces the effect of invention experience on a) invention capacity, and	Confirmed
b) adoption capacity.	Unsupported
H4– A change in design path reduces the effect of adoption experience on a) invention capacity, and	Confirmed
b) adoption capacity.	Unsupported

* Strongest economic effect.