Iterative Coordination in Organizational Search

Sourobh Ghosh, Andy Wu
Harvard Business School

Abstract

Firms use iterative coordination, or periodic coordination meetings, in their technology development on a presumed link to both exploratory innovation and exploitative performance. We critically evaluate this practice and identify boundary conditions to its effectiveness. With a leading technology firm, we embed a field experiment within a software development competition to measure iterative coordination's effect on firm outcomes and search process. We find that iteratively coordinating firms conduct more productive overall search, but face trade-offs between exploitation and exploration: while sampling more distant neighborhoods on their landscapes, these firms ultimately exploit at the expense of exploring. Our findings contribute to literatures on organizational search and strategy formation in entrepreneurial settings. Methodologically, we introduce a novel experimental data collection methodology enabling granular minute-level search measures.

Introduction/Motivation

Iterative coordination (e.g., Agile stand-up meetings) used by over 70% of organizations to manage software and non-software projects

Practitioner expectations of simultaneous exploration (e.g. creativity) and exploitation (e.g. product quality) are in conflict with conventional wisdom in academic literature

Non-random selection into Agile makes studying this phenomenon within a company difficult

Approach: embed field experiment in one-day app development hackathon tracking early-stage software projects

Search in Firm Software Code

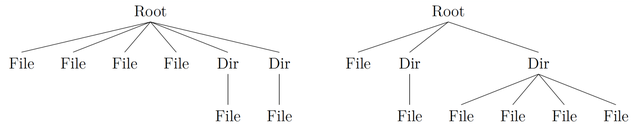

FIGURE 1: File Hierarchy Branching Factors. The two file hierarchies above share the same number of files and directories and a depth of 2 levels but demonstrate different patterns of exploratory and exploitative search.

Variable	Definition	Search Interpretation
Files	Number of files across the file hierarchy	Overall search
Directories	Number of directories across the file hierarchy	Overall search
Lower Descendants	Total nodes (files + directories) below root/first level of file hierarhcy	
Overall Branching	Average branching factor for the entire file hierarchy. Equal to $\frac{files+directories}{directories}$	
Upper Branching	Branching factor at the root/first of level of file tree. Equal to files + directories directly below the root	Exploitative search
Lower Branching	Average branching factor for directories below root/first level of a given file hierarhcy	Exploratory search

TABLE 6: Overall Search Productivity. Models are OLS, with robust standard errors clustered at the firm level. *p*-values are displayed in brackets.

	Files		*Directories*		*Lower Descendants*	
	(6-1)	(6-2)	(6-3)	(6-4)	(6-5)	(6-6)
Standups × Post	0.373 [0.460]	0.567 [0.087]	0.443 [0.363]	0.613 [0.077]	0.426 [0.481]	0.652 [0.103]
ln(Lines + 1)		0.452 [0.000]		0.396 [0.000]		0.528 [0.000]
Firm FE	Yes	Yes	Yes	Yes	Yes	Yes
Minute FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	20520	20520	20520	20520	20520	20520

TABLE 7: Comparing Search Process Across File Hierarchies. Models are OLS, with robust standard errors clustered at the firm level.

	Overall Branching		*Upper Branching*		*Lower Branching*	
	(7-1)	(7-2)	(7-3)	(7-4)	(7-5)	(7-6)
Standups × Post	−0.007 [0.941]	0.036 [0.499]	0.185 [0.291]	0.258 [0.047]	−0.200 [0.139]	−0.169 [0.054]
ln(Lines + 1)		0.100 [0.000]		0.170 [0.000]		0.127 [0.000]
Firm FE	Yes	Yes	Yes	Yes	Yes	Yes
Minute FE	Yes	Yes	Yes	Yes	Yes	Yes
Observations	20520	20520	20520	20520	12454	12454

Firm Outcomes

Third party panel of experienced judges evaluated day-end firm apps for *Appeal*, *Creativity*, and *Completion*

Appeal as measured by attractiveness to customers as outcome of exploitation, whereas *Creativity* is that of exploration

Completion compared to rule out potential confound of judge perceptions

TABLE 3: Firm Judge Scores. Models are OLS, with robust standard errors clustered at the firm level. *p*-values are displayed in brackets. The *Forfeit* variable, which takes a value of 1 for firms that forfeited and did not undergo the judging process, is included in odd numbered models below

	(3-1) Appeal	(3-2) Appeal	(3-3) Appeal	(3-4) Appeal	(3-5) Creativity	(3-6) Creativity	(3-7) Creativity	(3-8) Creativity	(3-9) Completion	(3-10) Completion	(3-11) Completion	(3-12) Completion
Treatment Group	0.614 [0.009]	0.846 [0.007]	0.546 [0.014]	0.661 [0.042]	-0.499 [0.082]	-0.687 [0.079]	-0.692 [0.046]	-0.960 [0.018]	0.279 [0.435]	0.385 [0.434]	0.251 [0.562]	0.336 [0.578]
Current Student			0.725 [0.050]	0.991 [0.145]			0.145 [0.714]	0.638 [0.431]			0.336 [0.563]	0.917 [0.338]
Graduate Degree			0.430 [0.273]	0.365 [0.544]			-0.458 [0.386]	-0.877 [0.224]			-1.065 [0.112]	-1.640 [0.078]
GitHub			0.120 [0.881]	0.106 [0.914]			0.893 [0.378]	0.907 [0.423]			-0.223 [0.855]	0.013 [0.992]
Google Development			-0.207 [0.618]	-0.310 [0.707]			0.898 [0.082]	1.006 [0.165]			-0.127 [0.838]	-0.352 [0.707]
Software Development			-0.050 [0.080]	-0.039 [0.354]			0.007 [0.872]	-0.003 [0.957]			-0.042 [0.480]	-0.043 [0.578]
Prior Hackathons			-0.144 [0.108]	-0.182 [0.106]			-0.096 [0.444]	-0.046 [0.782]			-0.094 [0.529]	0.021 [0.944]
Firm Size			-0.113 [0.274]	-0.096 [0.527]			-0.222 [0.280]	-0.421 [0.075]			-0.169 [0.424]	-0.195 [0.524]
Forfeit	-3.497 [0.000]		-3.702 [0.000]		-3.336 [0.000]		-3.331 [0.000]		-2.772 [0.000]		-2.929 [0.000]	
Constant	3.274 [0.000]	3.154 [0.000]	3.590 [0.000]	3.456 [0.009]	3.518 [0.000]	3.615 [0.000]	3.284 [0.008]	3.782 [0.017]	2.670 [0.000]	2.615 [0.000]	4.012 [0.009]	3.640 [0.114]
Observations	38	27	38	27	38	27	38	27	38	27	38	27

Conclusion

We find that iterative coordination increases search productivity, while favoring exploitative search over exploratory search

We contribute to/empirically validate the **theoretical search literature**, introducing 1) **novel data collection methodology** via Git and 2) **the use of hackathons** as empirical setting for strategy and entrepreneurship research

Study limitations include group size, inability to capture communication mechanisms, junior engineers

Future research should focus on boundaries to our findings (organization size, problem-solving context), other outcomes (e.g. emergence of shadow hierarchy)

Method

Population (112): sophomore+/post-grad CS majors, professional developers, etc.

Participants registered as firms of 2-4

Hackathon as entrepreneurial setting, providing market competition with judges representing consumer choice

Task

Competing firms developed apps that achieved some prosocial goal

Firms required to 1) develop off of toolkits provided by sponsor, and 2) use Git for version control

Experiment

Mentors facilitate (but do not run) stand-ups

Treatment: every two hours, three questions:

"What have you accomplished since your last check-in?"

"What are your goals until the next check-in 2 hours from now?"

"What are your goals for the end of the day (and have they changed)?"

Control: every two hours, null interaction

Difference-in-differences design with 2.5 hour pre-treatment period (for search panel data):

$$ln(Y_{it}+1) = \beta(Standups_i \times Post_{it}) + ln(Lines_{it}) + \alpha_i + \delta_t + \epsilon_{it}$$

Discussion

How does an organization solve universal problems of task division and allocation when system-level goals are in flux?

Before task division and task allocation can occur, agents in organization must reach a working agreement over their system-level goal

We define iterative coordination in the context of organizational search as an explicit update to the organization's goal definition

Interpretations

Increased iteration may lead to imperfect shared mental representations across organization (Knudsen & Srikanth 2014)

Simplifications in mental representations can help reduce search effort (Csaszar & Levinthal 2016), potentially increasing search productivity (Ethiraj & Levinthal 2009)

Inaccurate representations preferable to counterfactual of one that doesn't distinguish between alternatives (Puranam & Swamy 2016)