Moderating Effects of Turnover in Software Teams

By
Lee Eason
December, 2026

Director of Thesis: Dr. Kanchan Das
Major Department: Department of Technology Systems

ABSTRACT

This study analyzed 685 external turnover events across 459 software development teams that occurred between 2020 and 2025 to examine how team size and temporal distribution moderate productivity recovery. Factorial ANOVA with Benjamini-Hochberg FDR correction tested three hypotheses regarding team configuration effects on recovery measured against departing engineer and team baseline productivity. An algorithm to quantify temporal distribution of a team was developed and documented. Team size and temporal distribution showed weak independent effects (66.7-83.3% robustness, $\eta^2 \approx 0.003$-0.004), but their interaction demonstrated 100% temporal robustness across all six recovery months studied ($\eta^2 \approx 0.013$). Small teams (3-4 members) achieved 131.4% recovery when temporally concentrated versus 106.4% when distributed, while large teams (8-10 members) showed the inverse: 129.3% recovery when distributed versus 98.6% when concentrated. The 8-member threshold represents a qualitative shift where coordination benefits transition from synchronous (smaller teams) to asynchronous mechanisms (larger teams). These findings translate to 5.5 hours per developer monthly, providing quantifiable guidance for team formation and turnover management in distributed software organizations.

Moderating effects of Turnover in Software Teams

A Thesis

Presented to the Faculty of the Department of Technology Systems

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Masters in Technology Management

By

Lee Eason

December, 2026

Director of Thesis: Dr. Kanchan Das, PhD
Thesis Committee Members:
Dr. Merwan Mehta, PhD
Dr. Erol Ozan, PhD

DEDICATION

This paper is dedicated to my son in the hope that my small accomplishment here may bolster his courage to brave his own education with all the fervor of his creative heart.

ACKNOWLEDGEMENTS

Thanks first to my thesis committee for their guidance and feedback, and with extra gratitude to Dr. Das for taking a chance on me and giving me so much support. I would also like to thank my wife and son for tolerating my many nights spent on the back porch working on this paper and other school related tasks while I juggled the rest of my work responsibilities. I thank my employer for both funding and supporting my continuing education. Finally, I thank both my parents for teaching me that *whatsoever thy hand findeth to do, do it with all thy might.* And especially my father, who never gave up on helping me get admitted to ECU and pursuing this crazy dream.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHAPTER 1: INTRODUCTION

The craft of software development, and therefore the craft of designing and running software development organizations, has been evolving for well over 50 years. Producing software code is a blend of art and science, more akin to translating a book into another language than solving a geometry proof. There are over 28 million people now classified as a software developer globally (Vailshery, 2024). In the US, their median salary of over $130k USD per year (Bureau of Labor Statistics, 2025) means software development is a massive industry with significant capital investment tied to the engineers. Software engineering also has one of the highest turnover rates of any industry (Barua et al., 2025).

Turnover is when an employee leaves their company for some reason and must be replaced by a new person. There are both short- and long-term impacts to turnover. The company loses the direct contributions of the person who left while a replacement is hired. Whatever specialty knowledge and experience that person had which was not recorded or shared is also permanently lost to the company.

This presents a serious challenge for leaders of engineering organizations. There are dozens of books and hundreds of podcasts and internet blogs focused on improving retention by increasing engagement and satisfaction of employees. There is also a plethora of qualitative academic studies on the many factors that lead a worker to leave their job for another company (Carvalho et al., 2024). Understanding these drivers is critical because the cost of losing expertise, loss of productivity, and the expense of finding a replacement and training them is incredibly high. One study in 2007 showed that the cost of replacing a technology employee exceeded all other fields (O’Connell & Kung, 2007).

Regardless, turnover is inevitable. For a manager of software engineering teams, this is a significant challenge. Their company applies funding towards software-based solutions to business challenges with an expectation that those solutions will be delivered on time and bring a positive return on investment. When the people doing that work resign to go to another company, the timelines and budget for the project are not typically changed. The expectation is that a good manager will plan for and handle situations like that in the normal course of their work. Therefore, it becomes vital that managers develop processes and a team culture that minimize turnover and the impact of turnover when it occurs as much as possible.

Making matters complicated is that the software engineering industry has been under constant transformation of the technology used to write software, the tools used to collaborate, and the so-called best practices used to manage the overall process. Software projects were primarily managed with a manufacturing approach in the 1970s through the 1990s. This meant a great deal of effort was spent before the project began, attempting to define every aspect of the software and what exactly it would do when finished. A combination of lessons learned from lean manufacturing principles shared by Toyota and extreme market pressure during the Dot Com boom of the 1990s led to a rethinking of software engineering. In 2001 a group of practitioners published what famously became known as the Agile Manifesto (Agile Alliance, 2001).

The manifesto sparked a shift in focus away from heavy pre-planning of work towards more rapid iteration of working software. This sparked a dramatic shift in process and led to the emergence of a whole new generation of process consulting and job titles. Experts in agile software development frequently publish blog posts and sell consulting services designed to improve organizational efficiency, but they rarely employ scientific methods. As a result, many so-called best practices today are based on anecdotal experiences and are not generalizable.

There are hundreds of internet articles from Agile coaches and consultants espousing the advantages of smaller teams with no data of any kind to back up the claims. Their claims seem to be based on theoretical frameworks such as coordination theory but without any real-world data to help us navigate the complex contexts of various work environments. Similarly, there are hundreds of opinion pieces that argue for and against co-located teams who work together face to face versus distributed teams, again with no data to create a fact-based perspective.

Interestingly, there is a good body of research on all three topics relevant to this study: turnover, team size, and distributed teams. What has not been adequately explored yet is the relationship between these things. There is no methodology in the literature to understand how team size and distribution moderate the impact of turnover. This brings us to the primary hypotheses of this study:

$H1_0$: There is no relationship between the size of a team and the time it takes the team to recover from a turnover event.

$H1_a$: Larger teams will be able to recover from turnover events faster than small teams.

$H2_0$: There is no relationship between temporal distribution of a team and the time it takes the team to recover from turnover events.

$H2_a$: Teams with high temporal alignment will recover from turnover events faster than teams that are more temporally distributed.

$H3_0$: There is no interaction effect between team size and temporal distribution when teams are recovering from a turnover event.

$H3_a$: The relationship between team size and turnover recovery is moderated by temporal distribution, such that the optimal team size for recovery depends on the team's temporal distribution pattern.

Having a scientific framework to understand these relationships will be valuable to software engineering leaders. Knowing which team size and temporal distribution are at higher risk of impact can help guide leaders to make structural and compositional changes to teams to lower risk. In cases where fitting teams to the guideline is not possible, it can guide leaders to more deeply employ other mitigating practices from the literature, such as improved documentation and more purposeful knowledge sharing among team members.

The data for this study came from a software-engineering analytics platform named BlueOptima. It was extensively analyzed to ensure suitability for study, and a thorough description of the tests and filters used are included in the methodology chapter. However, it is worth highlighting that this dataset appears to be unique in the literature in terms of its size and scope. The raw data covers 114 companies and over five years of daily coding activities for over 516k software engineers: over 98 million observations in total. Most research studies on software engineering are smaller studies or based on data from open-source projects that operate significantly differently than commercial settings (Duarte, 2022). This research will give a uniquely valuable and generalizable view of the interactions between turnover, team size, and team distribution in commercial setting.

CHAPTER 2: LITERATURE REVIEW

Measuring Productivity

One of the challenges with any study on software team productivity is that as of 2025 there is no clear consensus for how to define productivity for a software developer or a team. Stefan & Ruhe, (2018) encountered this challenge when compiling their list of studies on productivity as well. In fact, researchers have been wrestling with this problem since the 1970's. Early attempts to establish quantitative measures started with attempts to correlate Source Lines of Code (SLOC) as a measure of productivity (Walston & Felix, 1977). In 1979, Albrecht published a new approach that shifted focus away from counting lines of code to focusing more on the functionality produced (Abran & Robillard, 1994).

More recent studies have shifted to a qualitative approach. Some studies have leveraged qualitative surveys where productivity is defined in subjective terms based on stakeholder opinions (Berntsson-Svensson & Aurum, 2006). Another approach has been to rely on quantitative and objective metrics from project planning tools as a proxy of productivity, using data from tools like Jira, for example (Scott, Charkie, & Pfahl, 2020). And of course there are smaller scale, mixed methods studies that attempt to give more support for objective metrics by supplementing them with opinion-based surveys. Felderer and Travassos (2019) document this evolution well in their paper and show that researchers are converging on increasingly complex and multi-dimensional approaches in a search for trustworthy and accurate measures of productivity (Felderer & Travassos, 2019).

This has led to the emergence of several large publications from private and independent industry groups advancing more refined multidimensional frameworks. DevOps Research and Assessment (DORA) began publishing a comprehensive annual study in 2014 that examines many aspects of company practices, as well as the outcomes they lead to. For example, DORA found a connection between the maturity of monitoring tools and practices and the Mean Time to Resolution (MTTR) for incidents. When they pulled together all the different attributes of practices and outcomes, DORA was able to present an evidence-based maturity model that companies could follow that would improve their productivity and customer experience.

In 2021, Microsoft published a new framework for measuring productivity called SPACE which calls for using five dimensions: Satisfaction, Performance, Activity, Communication, and Efficiency (Forsgren et al., 2021). SPACE adds perceptual input to objective metrics and survey-based metrics to try and give an even more complete picture of developer productivity.

This study reverted to a simpler definition of productivity and an approach based on economics and supported in the literature. Since the goal of this study is not to give an absolute and accurate measurement of productivity the way that DORA and SPACE are striving to, but rather to compare the productivity of teams against one another as well as longitudinally, the more important factor is consistency of measure. The proxy metric we use must be applied in the same way across the entire study period.

In 1957, M.J. Farrell wrote a seminal paper that established a very simple economically focused understanding of productivity as the ratio of output to input (Farrell, 1957). The more output you could produce from a given input, the more productive a system was. Yadav and Merwah (2015) applied this concept to engineering, successfully applying an economic model to how engineering organizations create value (Yadav & Marwah, 2015).

This establishes the reasoning behind utilizing BlueOptima's Coding Effort (CE) metric for this comparative study. There is ample evidence in existing literature that objective measures based on source code contributions are correlated to perceptions of stakeholders (Lambiase et al., 2024), (Oliveira et al., 2020). Taken as a whole, the literature demonstrates that it is rational to use metrics related to software code changes to measure how productive a software engineer is. It also shows that multi-dimensional models such as CE are correlated to qualitative measures of business satisfaction with these outputs. CE provides us with a measurement suitable for a comparative study of software engineering productivity. More details on how CE works are provided in the methodology chapter.

Turnover

There is a good amount of research related to the topic of turnover, mostly in the general area of succession planning and learning theory. There is also a solid body of research on the topic of turnover in software engineering organizations, likely due to the higher than normal rate of turnover in software engineering.

That turnover affects productivity is both intuitively obvious and supported by the literature. It is repeatedly shown that increasing turnover rates lead to lower quality and worse outcomes (Park & Shaw, 2013). Prior research also shows that different types of turnover lead to variable impacts on productivity. Voluntary external departures, where the engineer chooses to leave the company for another firm, tend to have the highest impact as normally these are good performers and all their knowledge is lost when they depart (Pee et al., 2014). Park & Shaw also found that effects of turnover were not linear: when there were multiple departures over time, early departures had a higher impact than follow up departures likely because the team was so destabilized that the effect was lessened.

One of the profound consequences of turnover is the loss of knowledge. Software development is an inherently knowledge-based activity. Some of that knowledge is transferred into the source code, but much of the knowledge required to write software code is inherently tacit and based on the organization itself (Lucie & Hana, 2011). That knowledge extends to things related to future plans, customers, and rationale behind design decisions. This leads to a fundamental facet of software engineering organizations: that the organization develops a memory system maintained by the workers (Argote & Ingram, 2000). As Pee et al. (2014) found in their study, even losing a single key engineer can have devastating impacts on the organization as a whole.

This concept of “organizational forgetting” is crucial in understanding how turnover impacts productivity. Knowledge transfer and retention strategies help mitigate this effect, but in practice these strategies are not going to be consistently applied across organizations. Practices such as documentation, succession planning, and mentoring are shown to reduce impacts (Lucie & Hana, 2011), but they cannot completely offset them.

Another confounding factor is in the level of psychological safety in a team. One study found that when teams have high psychological safety, they are more comfortable sharing information, resulting in higher redundancy in knowledge (Edmondson, 1999). This would mitigate some of the effects of turnover by making the organizational memory more resilient. In contrast, a team culture that favors siloing of knowledge will be much more vulnerable to turnover.

In summary, there are two primary sources of impact to productivity during turnover: the direct impact of the loss of an individual’s contributions, and the bigger, harder to measure, and longer-term impact of the loss of their tacit knowledge.

Recovery

The question now turns to what research has been done to understand the impacts of turnover. Given that we know some immediate loss of productivity will occur, and loss of knowledge will have longer term effects, how do those things play out in the real world and what does that tell us about typical recovery timelines for an organization?

At the macro level, a comprehensive study of firms from 1997 to 2014 investigated the impacts of clusters of employees leaving to start another company, called a “spin-out (Chila et al., 2024).” This study showed that at the aggregate level, innovation was negatively impacted by these events for several years. However, they also showed that the level of impact was closely associated with how central the knowledge area of the departing experts were to the core business of the company. They also showed that the timeframe for recovery was moderated by how well the parent company could deploy human capital resources to backfill the departed experts.

At the team level, another theoretical framework can help us understand the mechanism of recovering from a turnover event. Marks et al. (2001) theorized that teams move between action and transition phases depending on context (Marks et al., 2001). We can consider the recovery period of a team after a departure and after the replacement is found and onboarded to be a transition period where time and energy is spent on that activity at the expense of the productivity focus during an action period. The less efficient a team is during this transition phase the less productive they will be.

In summary, the literature reflects a consensus that teams will typically return to or exceed prior levels of performance after a turnover event, but the time it takes to achieve that varies greatly and is highly context dependent. This study sought to clarify at least two elements of that context to understand their influence on the recovery time: team size and temporal distribution.

Team size and performance

The ideal size of a software team has been a hotly debated topic in the industry for many years. That size of team is related to productivity has been well established. Even in recent studies leveraging large datasets from open source projects, there is a clear correlation showing that after a certain point, as team size increases the performance of the team drops (Rodríguez-García et al., 2012). For our study we must understand the drivers behind that. The reason why team size is such an important factor lies in the intersection of four theoretical frameworks: socio-technical congruence, coordination theory, team cognition theory, and knowledge specialization frameworks. Taken together, these explain why team size effects are not linear or simple. The context of the organization will have a large impact on team size effects. Ultimately there is no one ideal team size, and many factors contribute to determining that ideal for any given company.

Socio-Technical Congruence Theory

Cataldo, et al (2008) introduced the socio-technical congruence theory as a foundational framework for understanding team size effects in software development. The theory posits that software developer productivity depends on the alignment between technical dependencies and social coordination patterns. When coordination patterns are congruent with their coordination needs, teams showed a 32% decrease in modification request resolution time (Cataldo et al., 2008). In other words, the architecture of the system should not be any more complex than is required by the team(s) supporting it.

This exposes two important levers in the discussion of productivity and team size: the application of technology and the size of the team. As team size increases, achieving socio-technical congruence becomes exponentially more difficult due to the quadratic growth in communication paths, following the formula N(N-1)/2. As the number of people on a team increases, the number of communication paths increases exponentially. This helps explain why productivity does not scale linearly with team additions.

Interestingly, the amount of the effect of this theory changes based on team distribution patterns and technical innovations (Kwan et al., 2011). This implies that researchers should regularly revisit the guidelines from studies that leverage this framework as technological innovation is incredibly fast, and it is likely that findings are quickly outdated. Given that this study is based on data through April of 2025, the findings should help give updated guidance based on real world experiences of developers using the latest tools available.

Coordination Theory

As team size and number of teams increase, more effort is required to keep developers working well together. Kraut and Streeter's (1995) work helped clarify the mechanisms behind that obvious statement. They identified four coordination methods: standards (methodologies and practices), plans (documentation and schedules), formal mutual adjustment (structured meetings), and informal mutual adjustment (ad-hoc communication) (Kraut & Streeter, 1995). Critically, they found that the effectiveness of these mechanisms deteriorates as team size increases, again confirming the quadratic scaling of communication pathways.

Studies of large-scale agile methodology implementations often show that the complexities in managing coordination and dependencies lead to failure (Bick et al., 2018). The key takeaway here is that as systems and teams scale, it becomes harder to run the organization. This has a direct implication on our study as we seek to understand the relationship between turnover and team size.

Team Cognition Theory

As teams form and work together, they build two critical cognitive components: a shared understanding of the tasks to do, and a shared understanding of the team itself (He et al., 2007). This shared understanding enables the team to work together more effectively. It is intuitive that as team size increases, the cognitive load to build and maintain that shared understanding increases, meaning that team size will influence productivity. Interestingly, He et al.'s work implies that cognitive load due to the size of the team is a bottleneck that is directly related to productivity. When a team member leaves the team, rebuilding that understanding will impact productivity. This leads to a gap in the research literature. This study clarified the quantifiable size of this effect and how the size of the team is related to it.

Knowledge Specialization and Coordination Trade-offs

Faraj and Sproull's (2000) expertise coordination framework provides the final theoretical pillar for understanding team size effects. They analyzed 69 software teams and found that knowledge teams must manage three critical processes: expertise location (knowing where to find expertise), expertise recognition (knowing where expertise is needed), and expertise integration (bringing needed expertise to bear to solve problems) (Faraj & Sproull, 2000).

This framework exposes a fundamental trade-off in team sizing. Larger teams can provide broader and deeper expertise, but with a higher cost of coordination. Critically, their study found that expertise coordination had a stronger relationship with performance than expertise presence. In other words, having expertise on a team was not as important as being able to effectively leverage it.

This has clear implications for our study. Losing experts due to turnover will clearly damage productivity, but building the skill of expertise coordination with a replacement engineer may be directly related to the size of the team. Our study will help to clarify that relationship further.

Theoretical convergence

The size of a software team affects its maximal productivity, and these frameworks help reveal why. When considering the loss of a team member as a type of system shock event, the research gap that links recovery time after a turnover event to team size will help bring more clarity to how these theoretical frameworks apply to software teams.

Empirical Evidence on Optimal Team Sizes

Interestingly not all studies on team size reach the same conclusion. One cross industry study of over 200 software projects did not find a relationship between team size and the effort it takes to develop the software, implying that the addition of additional expertise balanced out the coordination overhead (Pendharkar & Rodger, 2007). Similarly, at least one prior study found that there is a non-linear relationship between team size and the actual cost to develop a software system, implying that the complex factors involved in real world software development paired with modern tools make predicting the impact of scaling a team up difficult (Pendharkar & Rodger, 2009).

However, most studies do suggest that there is evidence that there exists a general range of team sizes that will work best in most contexts. Studies have shown that smaller teams tend to have higher quality (Subramaniam & Nakkeeran, 2016). Again, Rodriguez et al (2012) found that teams of 9 or more were less productive than smaller teams in terms of function points successfully delivered.

In summary, team size is a factor many researchers have considered and the commercial industry wrestles with. There is a plethora of unscientific and opinion based articles shared by industry leaders based on their experiences. None of the available research has clarified the relationship between how team size affects the productivity impact of the team when dealing with turnover, however. This research will fill that important gap.

Geographic distribution and virtual teams

We shift our focus now to explore literature related to the study of geographic distribution in software teams. Since the 1980s, a shift away from Keynesian economics towards an open trade policy popularized by von Hayek led to companies adjusting their strategy for labor to take advantage of lower cost workers in markets like India. At first the focus was on lowering the cost of low-value work, but over time the labor markets of low cost locations increased in education and skill level, opening up the ability to shift higher value work as well (Lewin et al., 2009).

There is a wide range of research on the topic of globalization and offshoring. Many of these papers discuss the economics and global impacts of moving labor around the globe. Since the 1990's at least, researchers have been studying how globalization affects wages and equality (Feenstra & Hanson, 1996). However, our focus is not on politics or equitability, but rather on the way that globally distributed teams react to turnover versus co-located and time zone aligned teams.

Aside from the theoretical frameworks already discussed (socio-technical congruence theory, coordination theory, etc), there are two additional theoretical frameworks that can help us understand the relationship between global distribution and team productivity.

Media Richness Theory

In a foundational paper published in 1986, Daft and Lengel established a theoretical framework that helps explain how information is organized and communicated based on goals around reducing uncertainty and complexity (Daft & Lengel, 1986). The key finding for us in this research is that the composition of the audience is incredibly important when deciding how to communicate. Daft et al. proved this in a follow up study two years later by showing that high performing managers took that factor into account when deciding how to lead their teams (Daft et al., 1987).

The implication for us is that as teams grow more globally distributed, the cultural makeup of the team will grow more diverse, and the need to be more careful with information and communication will increase as well. This suggests that globally diverse teams may face more challenges when dealing with turnover. However, that finding is not in the literature yet, so this is another research gap our study filled.

Media Synchronicity Theory

Dennis and Valacich (2008) built on media richness theory to provide a slightly more nuanced view. They showed that the level of familiarity a worker has with their colleagues and with the work they are doing has significant impacts on which methods of communication of information are the most effective (Dennis et al., 2008). This again has significant implications for our study, as it is likely that the more distributed a team is, the harder it may be to gain familiarity with your coworkers when joining a team after a turnover event has occurred. This suggests that the more distributed a team is, the harder it may be to regain overall team productivity after a turnover event.

In fact, Kiely et al. called for further research on topics such as this study after constructing a theoretical framework that attempted to integrate coordination theory and media synchronicity theory (Kiely et al., 2021). They found that different methods of communication and tools had significant impacts on how effective they were at influencing globally distributed team performance.

Empirical Evidence on Productivity Impacts

In general, prior research that attempts to link geographic distribution with productivity has produced inconsistent results. Early studies on distribution were usually focused on the theoretical models already discussed here, keeping narrow views that often did not integrate those frameworks together or show causal connections (Kotlarsky, Oshri, van Hillegersberg, & Kumar, 2008). Many studies perform qualitative research that attempts to validate the ideas of a theory but fail to connect them to quantitative outcomes (Kotlarsky & Oshri, 2005). Often what appears to be a causal connection between geographic distribution and productivity turns out to be caused by more nuanced factors than just distribution. That was the case with a seminal study on the effects of globalization of software teams done in 2003.

Mockus (2003) showed that tasks done by distributed teams took 2.5 times longer than similarly sized tasks done by co-located teams (Herbsleb & Mockus, 2003). This paper has been cited over 100 times (Herbsleb & Mockus, 2025). However, when they investigated what was causing the increased time to complete tasks it was not simply distribution of the team, but rather the number of people involved in completing a task, even after correcting for team size and complexity of tasks. This raises a critical question: how can we better model the relationship between team size and distribution of teams? This study helped model that relationship using the shock events of turnover.

An in depth study of the development of Windows Vista published in 2009 also showed that distribution of teams had little to no effect on the quality or delivery of the software (Bird et al., 2009). As Bird himself pointed out in his paper, this was contrary to the popular thinking of the day. The most likely explanation for this is that the software development process that Microsoft used for Vista mitigated the common coordination challenges that plague distributed software teams.

However, a longitudinal field study conducted in 2007 found a strong relationship between team dispersion and lower performance (Ramasubbu & Balan, 2007). They also found that with careful application of engineering process improvement, many of the negative effects they observed could be mitigated, although not eliminated. Like other studies, Ramasubbu controlled for team size.

Practical Considerations of Globally Distributed Teams

Aside from the theoretical frameworks, there are other considerations we need to consider when thinking about distributed teams and how productivity and turnover may interact. Technological innovations in collaboration software are continuous, so the context in which teams we study operate are constantly shifting and changing. As many of the studies examined for this paper found, use of tools and processes can have a large mitigating effect on the downsides of distributed teams. This is supported by media richness theory - tools that improve how information is conveyed will improve synchronicity. This implies that studies dealing with distribution of teams, or any factor of software development that is subject to media richness theory or coordination theory constraints, will age quickly as new tools are adopted by the industry that may radically change the conditions.

Another factor to consider is physical versus temporal distance. Studies have shown that teams with high temporal distance have significantly higher difficulty communicating and scheduling meetings leading to higher challenges in coordination (O'Leary & Mortensen, 2009). Therefore, it will be important for us to distinguish between geographically distributed teams as defined by their city and country, and their temporal distance which is defined by their time zone.

Research Gaps

No study has modeled the relationship between team distribution and turnover on software teams. This will be the first study to quantify that relationship. There is also inconsistency in the literature about how to quantify temporal distance of team members. This study formalized a methodology that other studies can adopt.

CHAPTER 3: METHODOLOGY

Research Instrument

Comprehensive statistical analysis was performed on the research data to establish that it contained data appropriate for study. The data used was an anonymized dataset provided by BlueOptima as part of an academic research support program. The East Carolina University Institutional Review Board (IRB) reviewed the study protocol and the nature of the data and determined that the research did not constitute human subjects research as defined under 45 CFR 46. As such, formal IRB oversight and approval were not required.

BlueOptima measures developer productivity by gathering data from software code repositories and connecting them to various organizational information, such as which team the developer is on, and data from project collaboration tools such as Jira. The dataset spanned the period of January 1st, 2020, to April 1st, 2025. The unfiltered data spans 114 enterprises, 29,200 teams, and 516,250 software developers committing code to 726,553 code repositories. The developers were globally distributed across 1,856 cities and 108 countries that span 22 time zones. The size and scope of the research data made this study unique in academic literature.

However, because BlueOptima customers have high flexibility in how they deploy and configure the tool, there existed high variability in how "teams" were defined in the data. Companies were not required to supply information about the software engineer's location. It was therefore necessary to trim down the data to study only teams that fit reasonable criteria of what could define a valid team for the purposes of studying team level productivity. The criteria used for this study were as follows:

1. Teams must have at least 12 months of data
2. Teams must never exceed 30 members in a single month

3. Teams must have at least 3 members at some point in time
4. All team members must have a valid location with a defined time zone

After this first phase of filtering, the data ended up with daily activity records spanning 82 enterprises, 3,300 teams, and 27,896 developers committing code across 124,761 repositories. The developers in our filtered list of teams spanned 596 cities across 71 countries and 18 time zones. The time span of the data remained the same.

Description of Coding Effort and Defense of its Usage

BlueOptima's tool uses a proprietary method for measuring the level of effort that a single code commit reflects. They call this metric "coding effort" (CE). CE is derived based on a weighted model that uses 36 different metrics related to an individual software change. Those metrics include Lines of Code (LOC) added and removed, code comments, McCabe and Halstead complexity, fan-out metrics, and Data Abstraction Coupling – all well researched and well-known software engineering metrics. These metrics were selected to capture the volume, complexity, and inter-relatedness of the software change. Usage of combinatorial and weighted models such as this are well established in the literature of software productivity (Oliveira, Viana, Cristo, & Conte, 2017). BlueOptima uses these metrics in a proprietary model they built based on research initially done at Cambridge University to generate a number which represents the number of hours a typical software engineer would need to create the code change. That number is then pro-rated backwards to the last commit from that developer, giving a per day estimate of how many hours of positive productivity the developer was able to contribute.

Given that there is high variability in how teams of engineers work together, and activities such as knowledge transfer, planning, and other coordination tasks will take away from the time an engineer has available to contribute to code changes, CE gives us a way to compare how effectively a team is managing those tasks.

It is important to note that *the objective of this study is not to establish CE as an industry standard*. Nor is this study intending to validate whether CE is an accurate measure of productivity in isolation. This study leverages CE merely as a proxy metric for comparative study of productivity variance between individuals and teams of software engineers. The goal is to understand the effects of team size and temporal distribution, and any interaction effect those factors have on how well teams deal with turnover. So, while this paper cannot disclose the specific implementation of BlueOptima's algorithm for CE as it is their intellectual property, those specifics are irrelevant for this study. We can be confident in the validity of the framework they used, and that is enough to leverage CE as a comparative instrument for studying productivity variance.

Establishing Variability

Before we can study the relationship between team productivity and other factors, we must first establish that the research data displayed appropriate variability in team productivity. We must look at both team level and individual level productivity variance. Was there variability between teams in the same enterprise? We also need to know if teams display variability across enterprises.

For individual developer productivity, this study examined whether developers on the same team display variance, whether developers across teams but in the same enterprise vary, and finally whether developers across enterprises vary in productivity.

Variability Conclusions

Statistical analysis revealed that the per developer productivity displays high variance: standard deviation of productivity is 19.56 hours per month for an individual developer. 28.8% of that variance can be explained by which team the developer was on. This translates to roughly 5.8 hours of productivity differences per developer per month due to factors related to team configuration and processes.

Interestingly, 62.4% of individual productivity variance, or 12.5 hours per month, was found to be due to personal differences between individuals. This variance was likely related to years of experience, level of expertise, or the specialty skills people have and how well they apply them.

Only 8.8% of variance, or 1.8 hours per month, was related to which enterprise the person worked for. This is interesting because it suggests that when it comes to matters of company culture, how each team applies those things matters much more than what the high-level company messaging or intent is.

Examining team level productivity across enterprises shows that most variance came from team specific factors, not which company they were part of. 87% of across enterprise variance was related to team factors and only 13% were due to enterprise factors.

All these findings supported that our research data demonstrated strong internal validity and contained information that will help us understand where the factors that drive that 28.8% team level productivity variance comes from. The effect size of that variance was strong, so findings from this study will have positive commercial impacts for teams that incorporate them. Teams that implement these suggestions could see improvement of 7.4 hours of productivity per developer per month, or about 19.2%. See Appendix B for plots from these tests.

Research Design

Given our stated hypotheses and the theoretical frameworks reviewed, our variable interaction models for each hypothesis are revealed in the figures below.

Figure 1: H1 Variable Interaction Model

Figure 2: H2 Variable Interaction Model

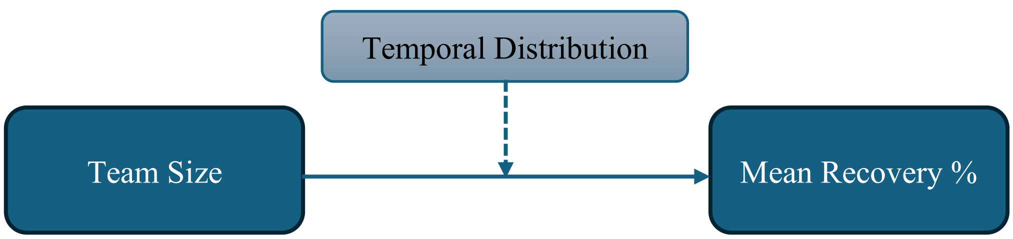

Figure 3: H3 Variable Interaction Model

Defining Turnover

Turnover is constituted of discrete but connected events in which a person leaves a company and is replaced by someone else. These events may take place over the course of many months, as the leaver is usually not immediately replaced. Our data shows a typical gap of 2.7 months between the leave and join events. Once the replacement joins there is a period of time in which they must learn how to operate successfully in the company and on their new team. For the purposes of this study, the term *turnover* will refer to the entire process.

It is common to make changes to team size for strategic reasons. Companies may choose to grow a team to increase their throughput or shrink a team to reduce spending. The data will have instances where teams are grown or shrunk over time that are not related to turnover. This study identified these situations by measuring the team size and ensuring only events where team membership changed but team size did not are included. This way, strategic growth of a team or shrinking of a team due to events like re-allocation of investment did not confound the results.

There was also a confounding possibility of internal VS external hires. Internal mobility is common in software engineering organizations; the data showed that 89% of team membership changes were some type of internal mobility. It is logical to assume that moving an engineer between teams will have very different impacts than hiring a completely new person into a company. This study controlled for internal VS external hires by checking to see if a replacement had already contributed code to another team at the company before joining the new team. The tests also checked to see if the departing engineer continued contributing with a different team. In either case the tests ignored the event as this study only focused on recovery time when losing expertise completely and replacing with a new engineer. A future study with this same data may examine internal mobility.

Next it was important to determine when turnover occurred on a team. Because the study data did not have explicit records from organizational design or human capital management tools, only daily CE records, the tests had to derive turnover based on the coding activity of the individuals on a team. By creating some rules it was possible to determine where there are records of team turnover.

This study identified a backfill event by finding where an engineer that had been contributing to a team ceased to have CE contributions for at least six months. This allowed for most normal vacation and leave of absence scenarios. The first month of no CE contribution was called the Turnover Start. If a new engineer begins contributing CE to that team within six months, and the total team size based on a rolling 6-month average from before Turnover Start stays the same, this study regarded this as a turnover event. If the team size grew or shrank significantly between Turnover Start and End then the tests concluded the team was undergoing other structural changes, confounding the turnover situation, and therefore ignored those events.

The month that the replacement engineer began contributing CE is referred to as month 1 in all charts and tables below, and this event signaled the end of the turnover event. We refer to this as Turnover End. We analyzed the productivity of the new engineer to see how well they achieved a level on par with both the per-developer average of their team from before the turnover start, and the average of the person they are replacing. Both were based on a six-month average from the time prior to Turnover Start. This allowed us to consider situations where the departing engineer was a low or high performer relative to the team.

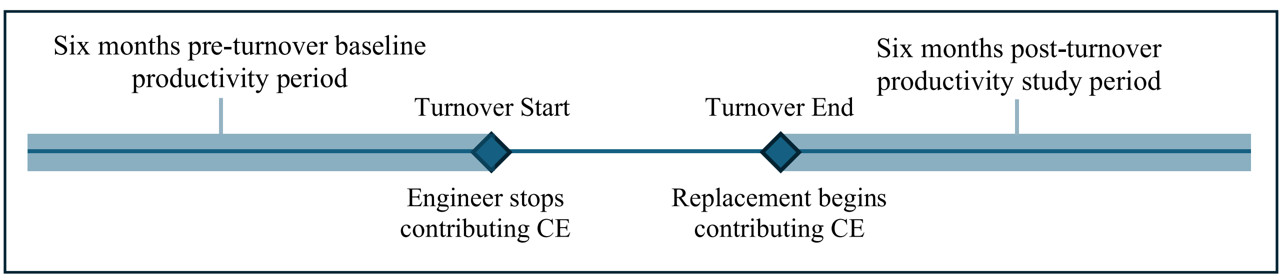

Figure 4: Visualizing the Turnover Process and Period of Study

There were 91,883 total team membership change events in the raw data, which included all leave (45,666) and join (46,217) events. Using the filtering rules described above, this study identified 7,707 turnover events. After filtering out internal mobility there were 1,603 events. This represents 48,192 daily observations of productivity before, during, and after turnover, but there was still one more layer of filtering to do.

There were some extreme outliers in the recovery data due to a lack of productivity data for some individuals and teams. Filtering further to ensure that the team, the engineer being replaced, and the replacement engineer all had at least 0.5 hours of CE per month as a baseline, and that the recovery median did not exceed 500%, retained 92.6% of observations and ensured the study data was fully clean and free of confounding events. This resulted in a final sample of 685 turnover events across 459 teams, 50 enterprises, and 44,603 daily CE observations spanning March of 2020 to April of 2025. This was the sample used for all tests and what will be described in the results chapter.

Team Sizes

The total, unfiltered research data showed a large variance in team sizes ($\tilde{x}$=8.73, σ=74.8). Our definition of a valid team filtered out teams that never have at least 3 members and never more than 30. This gave us a more valid distribution of team sizes in sample dataset ($\tilde{x}$=6, σ=3.96).

This study segmented teams into bins based on their mean size: small (3-4), medium (5-7), large (8-10), and very large (11+). The bin limits were selected based on balancing meaningful statistical robustness with prior literature on software team sizes. The data for each segment in the sample breaks out as follows:

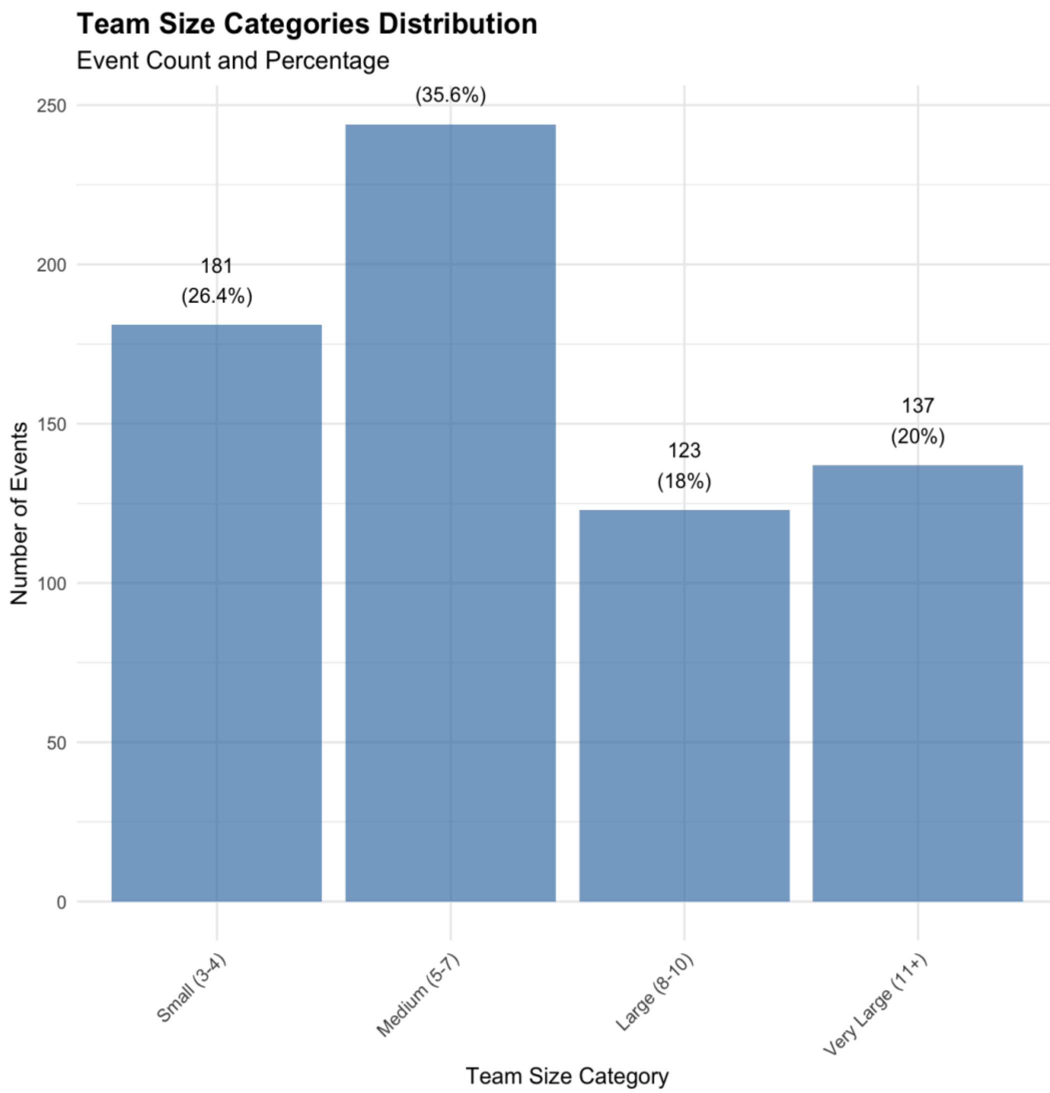

Figure 5: Team Size Segment Counts

Note that team sizes are not evenly distributed (Shapiro-Wilk W = 0.9132 , p-value = 2.358883e-19), so our tests analysis will need to take this into account.

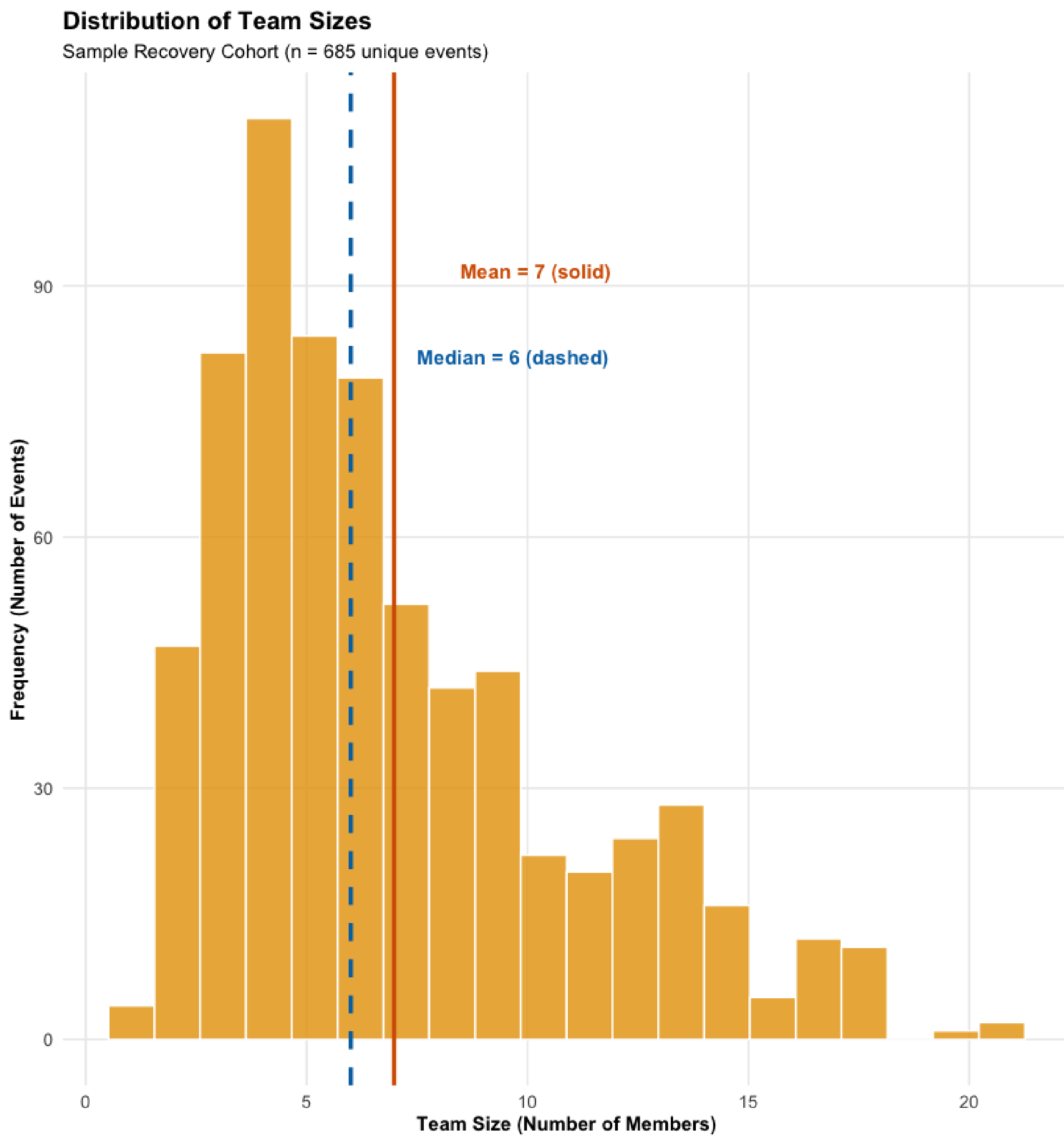

Figure 6: Team Size Distributions

Temporal Distribution

The literature review chapter already pointed out that teams can be distributed both geographically and temporally. While there are implications for teams that span multiple nations and cultures, this study focused on the effects of temporal distribution where team members are spread across multiple time zones. However, measuring that was not straight forward.

Because time zones are circular, it was not possible to measure temporal dispersion linearly. Someone living in GMT -10 (French Polynesia) and someone in GMT +12 (New Zealand) are not 22 hours apart in time zone as linear math would suggest; they live two hours apart and would have 6 overlapping working hours. Fortunately, Fisher (1993) developed an approach we can leverage. His work allows us to measure the circular variance of points dispersed on a 2d circle. This study applies Fisher's work in a novel way by mapping the time zone of each team member to a point on a 2d circle and using Fisher's methods to compute the circular variance of the team. When referring to the application of circular variance to the temporal distribution of a team, this study will notate it throughout as *temporal variance*.

Below is the algorithmic approach that was adapted from Fisher's work for this research study. Let *n* be the number of team members, and let each member's time zone be converted into an angle on the unit circle:

$$\theta_i = 2\pi * \left(\frac{UTC\ offset_i}{24}\right)$$

Where θ_i is the angle (in radians) corresponding to the time zone of the *i*-th team member. As Fisher showed, we now compute the average cosine and sine components:

$$\bar{C} = \frac{1}{n}\sum_{i=1}^{n} \cos(\theta_i)$$

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} \sin(\theta_i)$$

Then compute the mean resultant length:

$$R = \sqrt{\bar{C}^2 + \bar{S}^2}$$

Finally, we compute the temporal variance:

$$Temporal\ Variance = 1 - R$$

The resulting value lies in the interval [0,1] where:

- 0 indicates all team members are in the same time zone (full temporal alignment)
- 1 indicates maximum dispersion across time zones (uniform spread around the 24-hour clock)

When applied to a team, this formula produces a single metric that reflects how distributed the members of the team are across time zones. To illustrate this, let's look at three example teams from the study sample data and visualize their temporal variance.

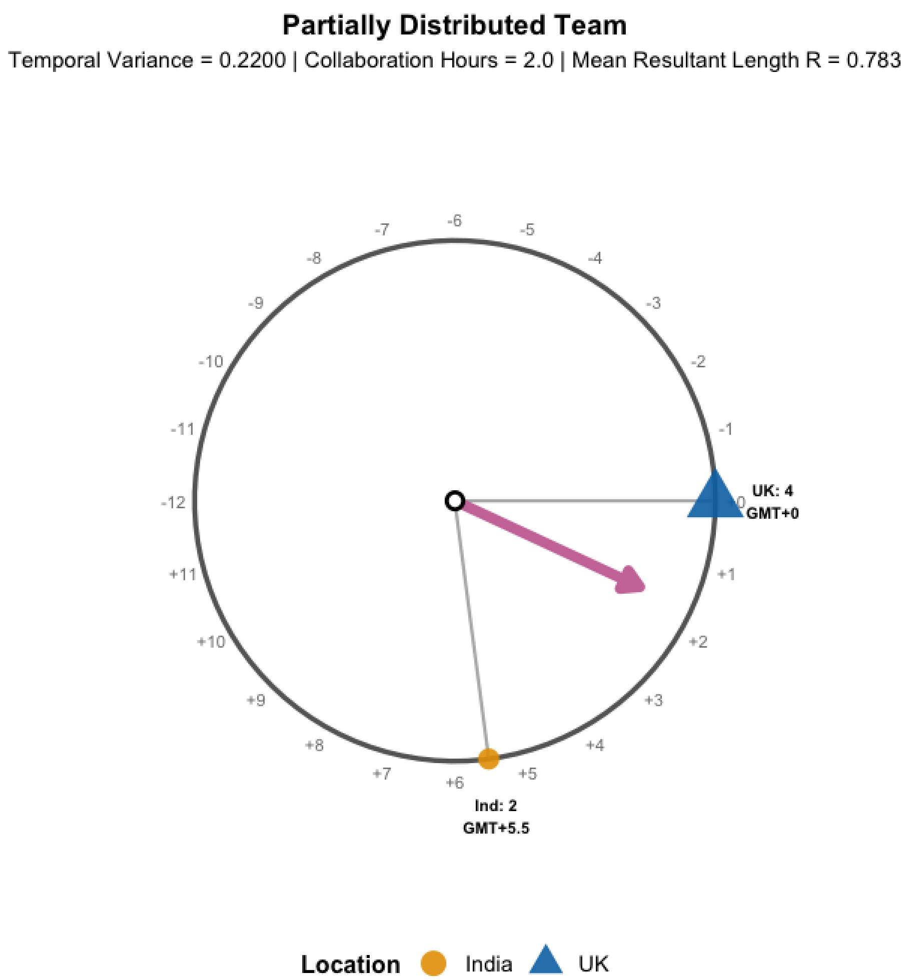

Figure 7: Temporal Variance of a Partially Distributed Team

This team has six members: two in India (GMT+5.5) and four in the UK (GMT+0). The dots represent team member locations around a 360-degree radial representing time zones around the world. Larger dots denote more team members. The team members have only two hours per day where every person's local working hours overlap. Observe that the blue vector arrow is showing the most balanced mean point between the dots, and the length of the arrow represents how well that mean fits all the points. In this example, that mean resultant length is 0.78, so the temporal variance is 0.22.

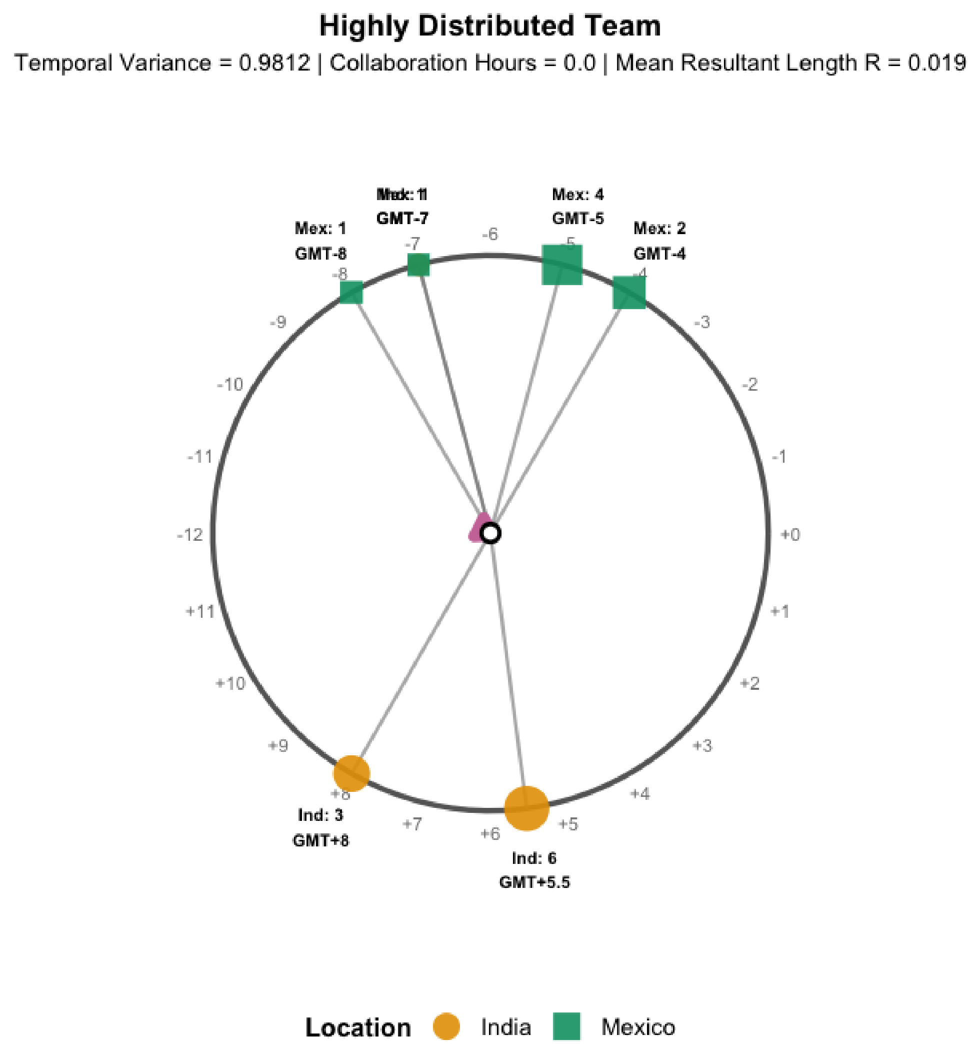

Figure 8: Temporal Variance of a Highly Distributed Team

Note that the example team in Figure 8 is primarily clustered in two groups across Mexico and India. While those two groups may operate as sub-teams, this study regarded these situations as being highly distributed because some level of coordination is required across all team members. Temporal variance for this team is 0.98, very close to total distribution, and the resultant length is so small you cannot see the vector at all. This team has 0 overlapping collaboration hours across the entire team, so they are almost certainly working in clusters that must collaborate asynchronously.

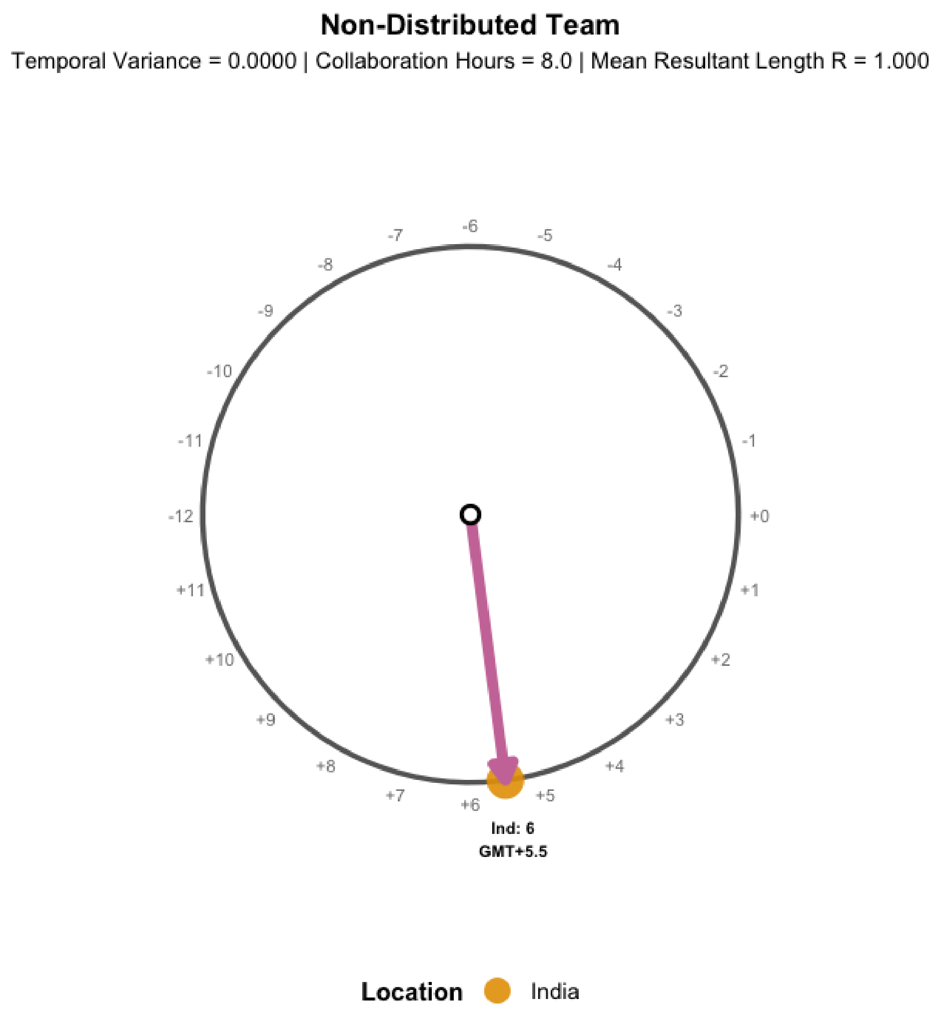

Figure 9: Temporal Variance of a Non-Distributed Team

Lastly, you see an example of a team with all members in the same time zone with Figure 9. Temporal variance is 0 and collaboration hours is 8.

Correlation between temporal variance and collaboration hours is very strong (Spearman correlation: r = -0.930), but temporal variance offers a more exact measurement that captures some of the nuance of temporal distribution as the diagrams above illustrate.

Through statistical analysis and examining the correlation of collaboration hours and temporal variance, bins were established for the distribution segments as follows. Note that for each bin you can see how many collaboration hours the teams in those bins typically have.

Segment	*Min Temporal Variance*	*Max Temporal Variance*	*Mean Collaboration Hours*	*Sample Turnover Events*
None/Not Distributed	0	0.0076	7.6	118
Some	0.0076	0.1166	3.7	321
High	0.1166	1	1.3	246

Table 1: Temporal Variance Segment Bins

Now that we have a method of quantifying temporal distribution, the next challenge is that temporal distribution for teams is often not static. As teams grow, shrink, or experience turnover, new or lost members may increase or decrease the temporal distribution. In the research data, 78% of backfills changed the temporal variance of the team. Very large teams logically showed more of a tendency to change their variance with backfills than small teams; 36.4% of small team backfills introduced no change to variance, while very large teams only kept the same variance 8.6% of the time. Overall, in the unfiltered data, 41.4% of backfills increased variance, 36.7% decreased variance, and 22% didn’t change it.

Since the goal is to understand how temporal distribution affects turnover recovery, this study used the temporal variance of the team after the backfill joined as the independent variable in the tests. This allowed the measures to accurately test for relationships between recovery and temporal variance, since it is the variance of the team with the new team member present that impacted how they handle tasks like knowledge transfer and training of the new person. If the tests were to use the prior variance value, it would be testing the wrong measure 78% of the time.

As with team sizes, the distribution of temporal variance across teams in our sample was not normal (Shapiro-Wilk W = 0.7795 , p-value = 2.280982e-29), so the tests employed corrections to make it possible to safely look for relationships.

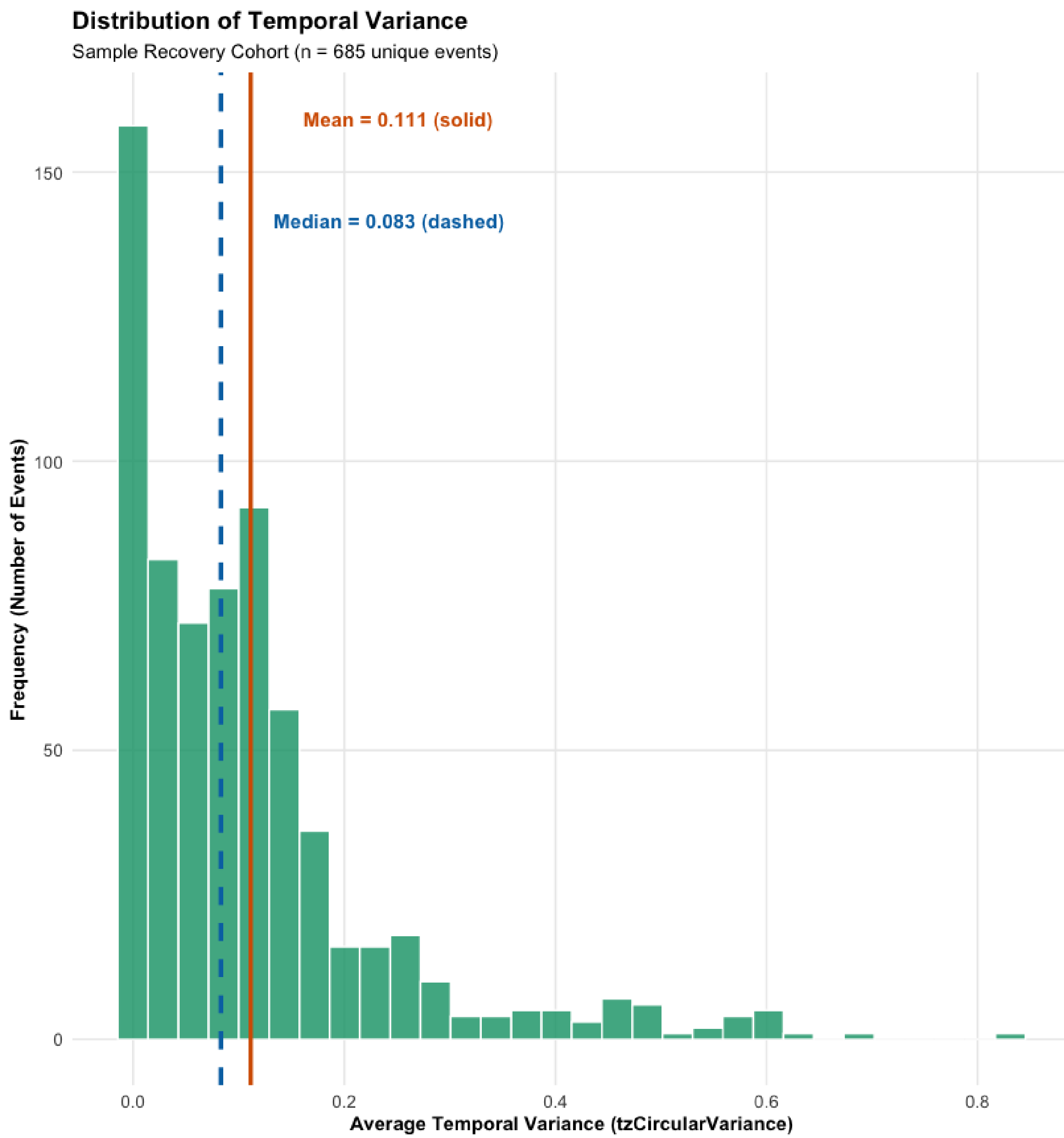

Figure 10: Distribution of Temporal Variance

Looking at the fully filtered sample you can see how many qualified turnover events there are for each combination of team size and distribution.

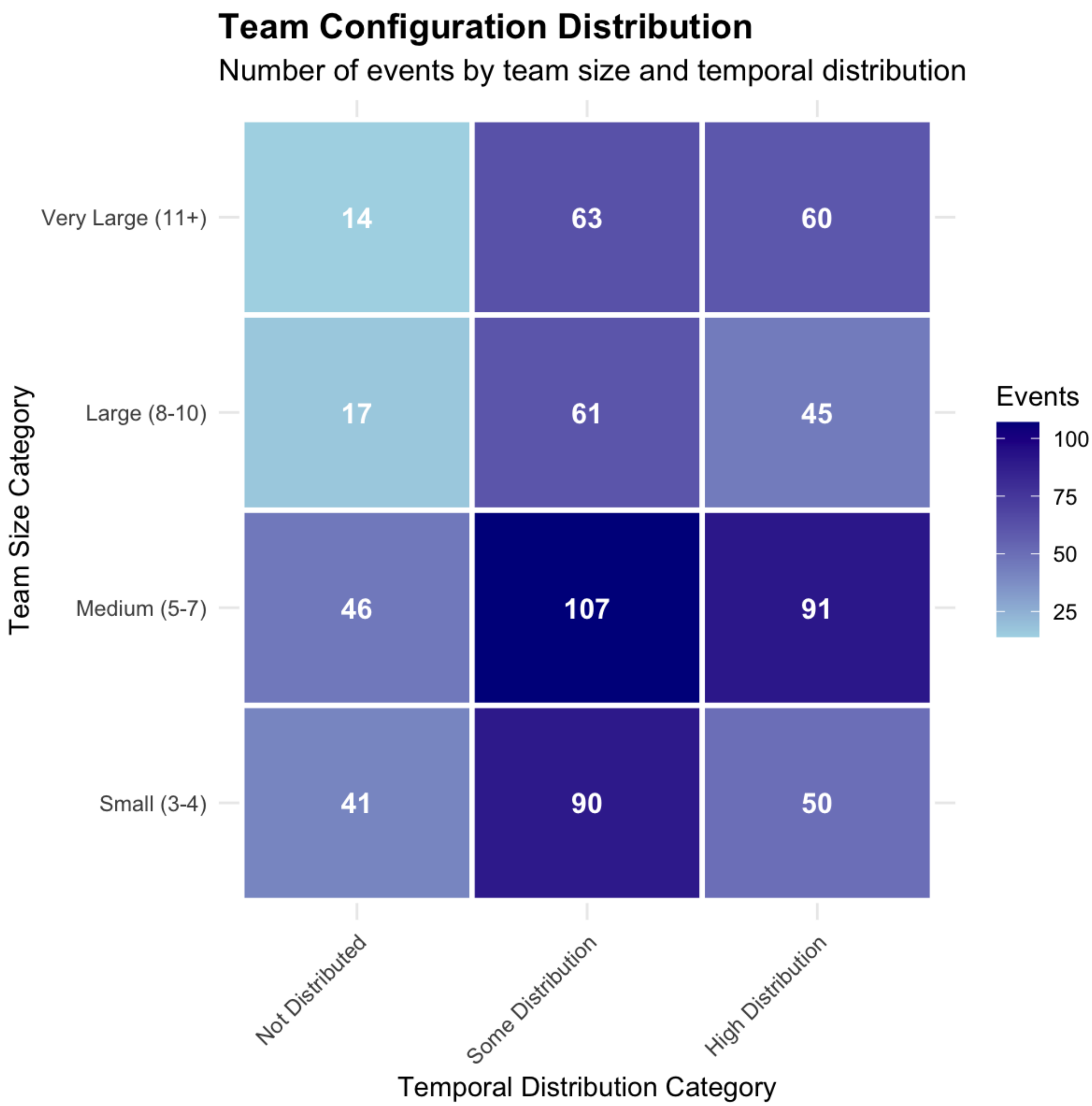

Figure 11: Events for each team configuration

While it may appear at first that some of our cells show low data counts, there were still at least 145 CE observations for each cell, offering excellent statistical robustness.

CHAPTER 4: RESULTS

Summary of Results

A factorial ANOVA approach was used to test for significant differences in recovery outcomes each month following the replacement engineer's start date. Recovery was measured against two baselines: the departing engineer's productivity and the team's prior average productivity. The analysis examined main effects of team size and temporal distribution, as well as their interaction effect. Type II ANOVA was conducted as a robustness validation to confirm that cell size imbalance did not affect conclusions; both approaches yielded highly consistent results. Benjamini-Hochberg false discovery rate correction was applied to control for multiple testing across the six-month follow-up period.

Hypothesis decisions integrated three criteria: statistical significance ($p < 0.05$ after FDR correction), temporal robustness (percentage of months showing significant effects across the six-month follow-up), and practical significance (partial η^2 effect sizes). Evidence strength was classified as strong (≥80% robustness, $\eta^2 > 0.01$), moderate (≥67% robustness, $\eta^2 > 0.006$), or weak (≥50% robustness). While Cohen's conventions for effect sizes were used for contextualization, hypothesis acceptance or rejection was based on the combination of significance, robustness, and effect size magnitude rather than conventional thresholds alone. Complete statistical methodology is detailed in Appendix C; all p-values are reported in Appendix D.

The primary discovery of this study is that ideal team size and temporal distribution depend on each other. The test results show that as team size increases, recovery from turnover benefits from temporal distribution. As team size decreases, temporal distribution hurts recovery. The breakpoint for this phenomenon is a team size of 8 people. There is a clear intersection in the data showing interaction effects between team size and temporal distribution. These interaction effects are strong in every month of recovery against both prior engineer and prior team average productivity. We can see this pattern by looking first at month 1 recovery in Figure 12, then examining the full trend across all months in Figure 13.

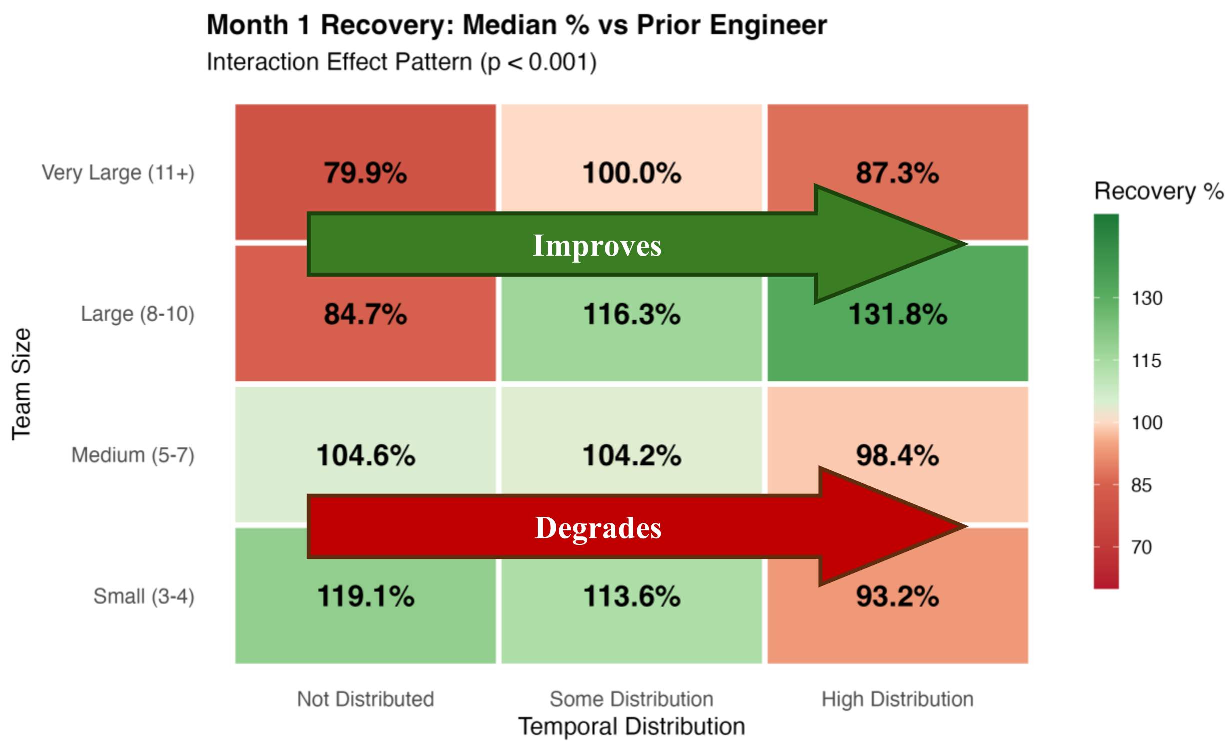

Figure 12: Month 1 Recovery vs Prior Engineer

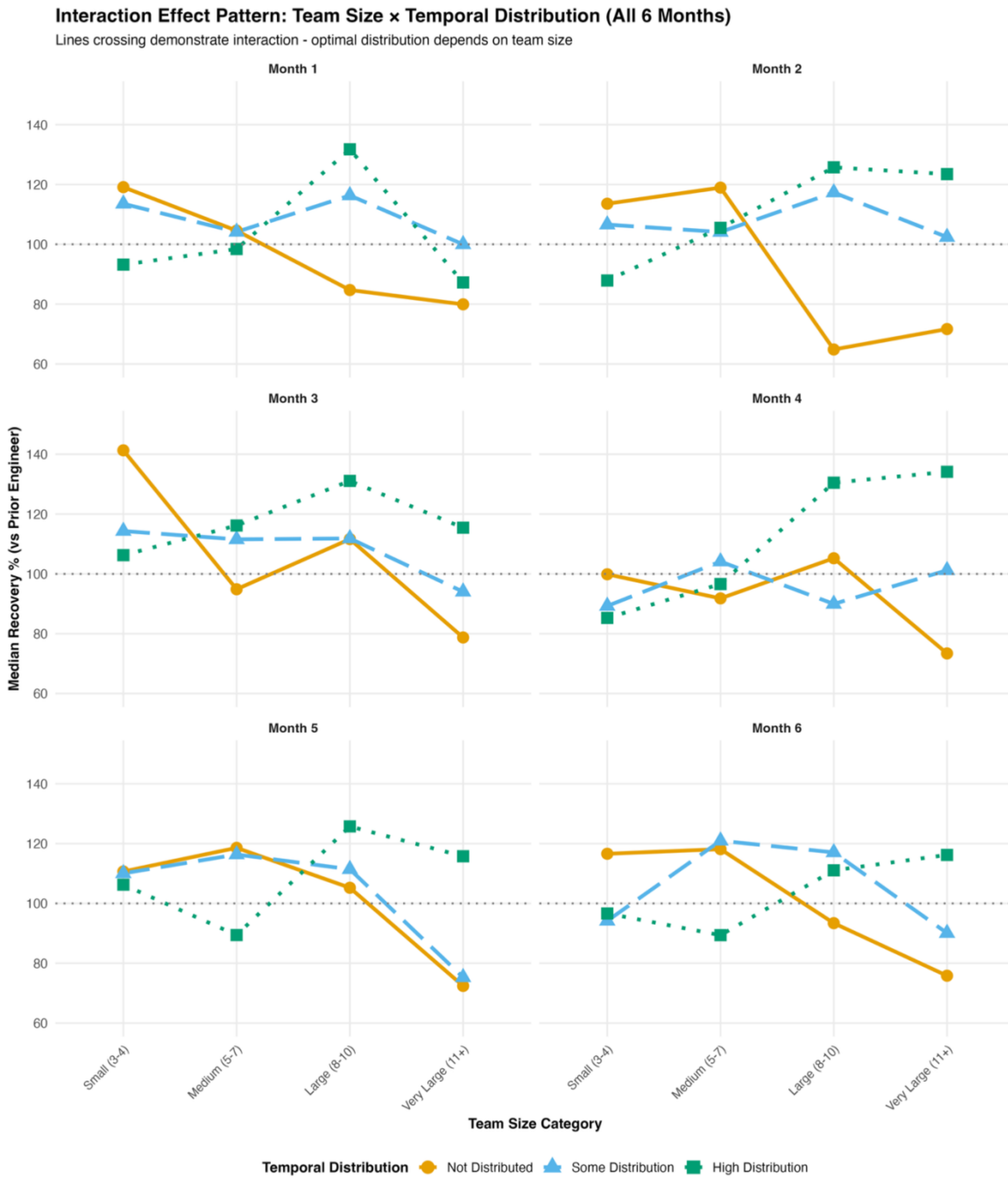

Figure 13: Interaction Effects on Turnover Recovery

Note in Figure 13 that every month shows that as team size increases, the effect of temporal distribution flips from hurtful (small teams) to helpful (for large teams). This demonstrates the consistency of the phenomenon Figure 12 showed.

Figure 14 below shows another way to visualize this. Note that the red line showing non-distributed team recovery over time moves from the top performing line for small teams to the bottom performing line for very large teams. The reverse occurs with the blue line indicating highly distributed teams.

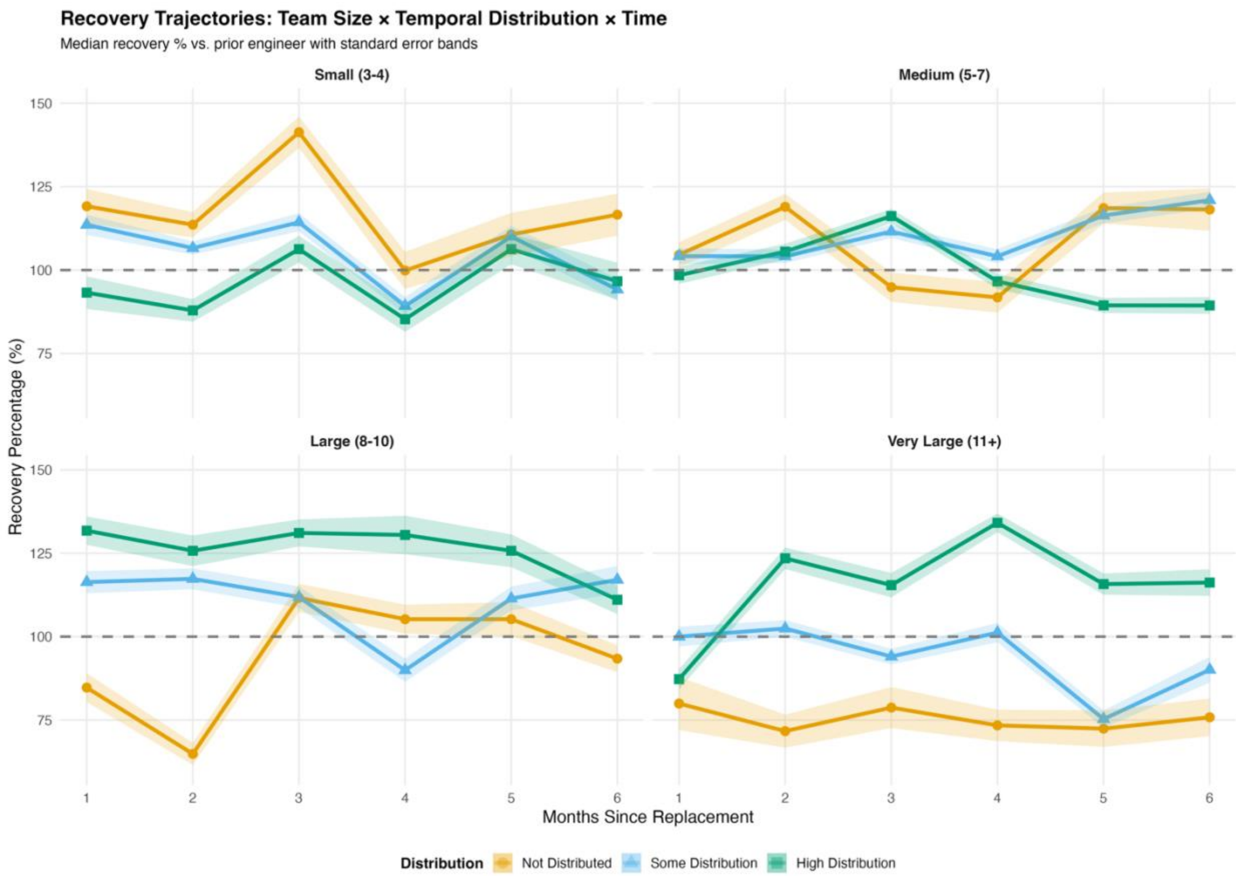

Figure 14: Recovery Trajectories by Team Size and Distribution

The main effects of team size and temporal distribution were inconsistent and weaker in their results on their own. See Appendix D for full details on F and P values. The interaction effect was significant in all months and tests and represents the most important finding, demonstrating that team configuration effects of size and temporal distribution cannot be understood in isolation. Optimal recovery depends on the alignment between team size and temporal distribution, with different configurations showing dramatically different recovery patterns ranging from 80% to 132% in the first month post-replacement.

1. Strong Support: Team size × temporal distribution interactions (100% robustness, $\eta^2 \approx 0.013$)
2. Weak Support: Team size main effects (67-83% robustness, $\eta^2 \approx 0.004$)
3. Weak Support: Temporal distribution main effects (50-67% robustness, $\eta^2 \approx 0.002$)

We then examined each pre-registered hypothesis to examine their specific findings.

Hypothesis 1: Team Size Effects on Recovery

$H1_0$: There is no relationship between the size of a team and the time it takes the team to recover from a turnover event.

$H1_a$: Larger teams will be able to recover from turnover events faster than smaller teams.

We **reject the null hypothesis $H1_0$** based on significant team size effects observed in 4 of 6 months (66.7% robustness) for recovery vs. prior engineer performance, and 5 of 6 months (83.3% robustness) for recovery vs. team baseline performance, with average effect sizes of $\eta^2 = 0.0035$ and $\eta^2 = 0.0038$ respectively.

However, the results do not support the alternative hypothesis $H1_a$ regarding the direction of effects. Contrary to the prediction that larger teams would recover faster, our findings revealed the opposite pattern. Marginal means analysis for Month 1 demonstrated that smaller teams achieved superior recovery rates across multiple configurations. Small teams (3-4 members) in Not Distributed settings achieved 131.4% recovery (95% CI: 123.0-139.7%), while Large teams (8-10 members) in the same configuration achieved only 98.6% recovery (95% CI: 89.3-107.8%).

This contradictory finding suggests that coordination complexity costs outweigh redundancy benefits during turnover recovery, supporting coordination theory predictions about communication overhead scaling quadratically with team size rather than simple resource availability models.

Hypothesis 2: Temporal Distribution Effects on Recovery

$H2_0$: There is no relationship between temporal distribution of a team and the time it takes the team to recover from turnover events.

$H2_a$: Teams with high temporal alignment will recover from turnover events faster than teams that are more temporally distributed.

We **reject the null hypothesis $H2_0$** based on significant temporal distribution effects in 4 of 6 months (66.7% robustness) for recovery vs. prior engineer performance, though evidence was weaker for recovery vs. team baseline (3 of 6 months, 50.0% robustness). Average effect sizes were small ($\eta^2 = 0.0028$ and $\eta^2 = 0.0013$ respectively).

However, we do not support the alternative hypothesis $H2_a$ in its simple form. The relationship between temporal alignment and recovery proved more complex than predicted. Month 1 marginal means revealed that moderate distribution ("Some Distribution") often yielded superior outcomes compared to both high temporal alignment ("Not Distributed") and high distribution ("High Distribution"). For example, Large teams with Some Distribution achieved 125.5% recovery (95% CI: 119.5-131.5%), outperforming the same teams when Not Distributed (98.6%) and performing comparably to High Distribution (129.3%).

This finding challenges simple co-location assumptions and suggests that moderate geographic dispersion may develop beneficial asynchronous coordination capabilities, supporting media richness theory predictions about communication adaptation rather than pure synchronous coordination advantages. The tools that allow distributed teams to work well together likely benefit new engineers joining the team.

Hypothesis 3: Interaction Effects Between Team Size and Temporal Distribution

$H3_0$: There is no interaction effect between team size and temporal distribution when teams are recovering from a turnover event.

$H3_a$: The relationship between team size and turnover recovery is moderated by temporal distribution, such that the optimal team size for recovery depends on the team's temporal distribution pattern.

The results **strongly reject the null hypothesis $H3_0$** and **fully support** the alternative hypothesis $H3_a$. Interaction effects demonstrated robustness with 100% significance across all monthly analyses for both outcome measures, with substantial effect sizes averaging $\eta^2 = 0.0132$ for recovery vs. prior engineer and $\eta^2 = 0.0121$ for recovery vs. team baseline.

The interaction pattern provides strong evidence that optimal team size for recovery indeed depends on temporal distribution patterns. Small teams showed a 25.0 percentage point recovery range across temporal distribution categories (131.4% to 106.4%), while Large teams demonstrated an even more dramatic 30.7 percentage point range (98.6% to 129.3%). This represents substantial practical significance, with the average interaction effect of 27.9 percentage points translating to meaningful differences in productivity restoration.

Critically, the data revealed distinct optimal configurations contradicting simple linear relationships:

- Small teams: Performed best when Not Distributed (131.4% recovery)
- Large teams: Performed best when Highly Distributed (129.3% recovery)

- Medium teams: Showed relatively stable performance across temporal distributions (109.8-115.1%)

This pattern strongly supports H3a by demonstrating that the relationship between team size and recovery is fundamentally moderated by temporal distribution. The findings align with socio-technical congruence theory, suggesting that optimal team configuration depends on alignment between coordination complexity and available coordination mechanisms.

Effect Size Interpretation and Practical Significance

While our effect sizes appear modest by Cohen's conventions ($\eta^2 = 0.003\text{-}0.013$), they represent substantial practical significance in organizational contexts. For this study, Cohen's conventions were only used to understand the effect sizes and were not used as part the hypothesis testing. The 27.9 percentage point average interaction effect translates to approximately 5.5 hours of productivity difference per developer per month, representing 658 hours annually for a 10-person team—equivalent to roughly $65,000 in value assuming standard billing rates.

The consistency of interaction effects across all time periods strengthens confidence in these findings. Unlike main effects that showed temporal variation, interaction patterns remained remarkably stable, suggesting fundamental differences in coordination mechanisms rather than transient adaptation effects. This consistency is particularly notable given the complex organizational dynamics surrounding turnover events.

Temporal Robustness Analysis

The six-month longitudinal design enabled assessment of effect persistence across the recovery timeline. Interaction effects maintained significance and substantial magnitude throughout all follow-up periods, with effect sizes ranging from $\eta^2 = 0.0063$ in Month 1 to $\eta^2 =$

0.0183 in Month 2. This temporal consistency suggests that team configuration effects represent stable coordination mechanisms rather than short-term disruption patterns.

Statistical Validation and Robustness Checks

Multiple validation approaches confirmed the reliability of our findings. The primary analysis used factorial ANOVA (Type I sum of squares), which is appropriate for balanced factorial designs testing main effects and interactions. To address the cell size imbalance in the observational data, Type II ANOVA was performed as a robustness validation. Results between Type I and Type II approaches remained highly consistent (difference < 0.01 for 83% of tests), confirming that cell size imbalance did not compromise statistical validity or alter substantive conclusions. Non-parametric Aligned Rank Transform (ARTool) validation yielded comparable significance patterns, supporting conclusions despite assumption violations.

Benjamini-Hochberg FDR correction proved conservative but appropriate, with 89% of raw significant findings surviving multiple testing correction. The high survival rate reflects the genuine strength of effects rather than type I error inflation, particularly for interaction effects where all tests remained significant after correction.

Effect size confidence intervals demonstrated adequate precision for organizational research, with the narrowest intervals for interaction effects (our primary finding) and wider intervals for main effects reflecting their weaker evidence base. The convergent evidence across statistical approaches strengthens confidence in the practical recommendations derived from these analyses.

CONCLUSION

This study provided strong empirical evidence that the effects of team size and temporal distribution on turnover recovery cannot be understood in isolation. The results demonstrated a robust interaction between these two software engineering team configuration variables, indicating that recovery outcomes following a turnover event depended on their alignment. Across the six-month recovery period, smaller teams exhibited faster recovery when temporally concentrated, while larger teams recovered more effectively when temporally distributed. Conversely, smaller and highly distributed teams showed consistently slower recovery, and larger, temporally concentrated teams exhibited reduced recovery performance. These differences were substantial in magnitude, with optimal team configurations demonstrating recovery rates more than 30 percentage points higher than the least effective configurations.

These findings are consistent with and extend several established theoretical frameworks in software engineering and organizational literature. The observed interaction between team size and temporal distribution aligns with coordination theory and socio-technical congruence theory, which emphasize the increasing coordination demands and communication complexity associated with larger teams. In this context, temporal distribution appears to offer coordination mechanisms that mitigate these costs in larger teams, while introducing additional friction for smaller teams where coordination overhead is otherwise minimal. Similarly, the results are consistent with media richness and media synchronicity theories, suggesting that teams adapt their communication practices to available media and temporal constraints in ways that influence recovery outcomes. Rather than indicating a uniformly beneficial or detrimental effect of temporal distribution, the findings highlight how its impact depends on the structural context of the team.

This study also presents a general approach for measuring temporal distribution of teams. The formulas shared can be re-used using data available in most organizations via a Human Capital Management (HCM) system. HCM providers should consider exposing temporal variance in their tools and helping organizational leaders have a better understanding of how their teams are distributed. The formulas presented in this study could be used in an organizational health summary to grade how well teams conform to the guidelines our conclusions present.

Future studies should incorporate these interaction effects as control variables when examining other aspects of productivity. For example, there are many studies being done now to examine the productivity impact of AI based software assistance tools. If those tools are used by engineers on full teams, understanding whether the configuration of the team they are on is influencing their productivity could help reduce confounding variables in those studies.

Another opportunity for future studies is to include qualitative data that helps us understand whether different team configuration attributes result in better productivity, rather than just more or less. A software engineer may write copious amounts of source code, but if that code doesn't meet end user expectations, or deliver good commercial results, then higher productivity may still be a net loss. This study also had no insight into the tools being used by the engineering teams being measured. Future studies should incorporate that as another potential differentiator of productivity. AI based tools are likely going to be a significant driver of variance as well and should be accounted for in future turnover studies. It's possible that highly distributed teams could use AI based tools to help them collaborate just as effectively as a more centralized team regardless of size.

REFERENCES

2014 State of DevOps Report. Retrieved from https://dora.dev/research/2014/

Abran, A., & Robillard, P. (1994). Function points: A study of their measurement processes and scale transformations. *J. Syst. Softw.*, *25*, 171-184. doi:10.1016/0164-1212(94)90004-3

Agile Manifesto. Retrieved from https://agilealliance.org/agile101/the-agile-manifesto/

Argote, L., & Ingram, P. (2000). KNOWLEDGE TRANSFER: A BASIS FOR COMPETITIVE ADVANTAGE IN FIRMS. *Organizational Behavior and Human Decision Processes*, *82*, 150-169. doi:10.1006/OBHD.2000.2893

Barua, R., Ghosh, S. K., Rahman, M. N., Biswas, K. C., & Haque, M. S. (2025). OPTIMIZING SOFTWARE ENGINEERING CAREERS: HIRING, RETENTION, AND WORKFORCE DEVELOPMENT. *Journal Of Creative Writing (ISSN-2410-6259)*, *9*(1), 49-65. Retrieved from https://jocw.ittc.edu.bd/index.php/jocw/article/download/148/103

Berntsson-Svensson, R., & Aurum, A. (2006). Successful software project and products: An empirical investigation. *Proc. ISESE*, *06*, 144-153.

Bick, S., Spohrer, K., Hoda, R., Scheerer, A., & Heinzl, A. (2018). Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings. *IEEE Transactions on Software Engineering*, *44*, 932-950. doi:10.1109/TSE.2017.2730870

Bird, C., Nagappan, N., Devanbu, P. T., Gall, H., & Murphy, B. (2009). Does distributed development affect software quality? *Communications of the ACM*, *52*, 85 - 93. doi:10.1145/1536616.1536639

Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, Software Developers, Quality Assurance Analysts, and Testers, at https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm (visited *August 04, 2025*).

Carvalho, M. L. L., Cruz, P. D. S., de Almeida, E. S., Neto, P. A. D. M. S., & Prikladnicki, R. (2024). Please do not go: understanding turnover of software engineers from different perspectives. arXiv preprint arXiv:2407.00273. Retrieved from https://arxiv.org/pdf/2407.00273

Cataldo, M., Herbsleb, J., & Carley, K. M. (2008). Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. doi:10.1145/1414004.1414008

Chila, V., Devarakonda, S. V., & Martin, X. (2024). EXPRESS: From loss to recovery: Spin-outs and parents' innovation. *Strategic Organization*. doi:10.1177/14761270241270971

Daft, R., & Lengel, R. (1986). Organizational information requirements, media richness and structural design. *Management Science*, *32*, 554-571. doi:10.1287/MNSC.32.5.554

Daft, R., Lengel, R., & Treviño, L. (1987). Message Equivocality, Media Selection, and Manager Performance: Implications for Information Systems. *MIS Q.*, *11*, 355-366. doi:10.2307/248682

Dennis, A., Fuller, R. M., & Valacich, J. (2008). Media, Tasks, and Communication Processes: A Theory of Media Synchronicity. *MIS Q.*, *32*, 575-600. doi:10.2307/25148857

Duarte, C. H. C. (2022). Software productivity in practice: A systematic mapping study. *Software*, *1*(2), 164-214. Retrieved from https://www.mdpi.com/2674-113X/1/2/8

Edmondson, A. (1999). Psychological Safety and Learning Behavior in Work Teams. *Administrative Science Quarterly*, *44*, 350 - 383. doi:10.2307/2666999

Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science*, *46*, 1554-1568. doi:10.1287/MNSC.46.12.1554.12072

Farrell, M. (1957). The Measurement of Productive Efficiency. *120*, 253-281. doi:10.2307/2343100

Feenstra, R., & Hanson, G. H. (1996). Globalization, Outsourcing, and Wage Inequality. *Sustainability & Economics eJournal*. doi:10.3386/W5424

Felderer, M., & Travassos, G. (2019). The Evolution of Empirical Methods in Software Engineering., 1-24. doi:10.1007/978-3-030-32489-6_1

Fisher, N. I. (1993). Statistical Analysis of Circular Data. Cambridge: Cambridge University Press.

Forsgren, N., Storey, M.-A., Maddila, C., Zimmermann, T., Houck, B., & Butler, J. (2021). The SPACE of Developer Productivity: There's more to it than you think. ACM Queue, 19(1), 20-48. Retrieved from https://www.microsoft.com/en-us/research/publication/the-space-of-developer-productivity-theres-more-to-it-than-you-think/

He, J., Butler, B., & King, W. R. (2007). Team Cognition: Development and Evolution in Software Project Teams. Journal of Management Information Systems, 24, 261 - 292. doi:10.2753/MIS0742-1222240210

Herbsleb, J., & Mockus, A. (2003). An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Trans. Software Eng.*, *29*, 481-494. doi:10.1109/TSE.2003.1205177

Herbsleb, J., & Mockus, A. (2025). Retrospective: An Empirical Study of Speed and Communication in Globally Distributed Software Development. *IEEE Transactions on Software Engineering*, *51*, 833-835. doi:10.1109/TSE.2025.3533977

Kiely, G., Butler, T., & Finnegan, P. (2021). Global virtual teams coordination mechanisms: building theory from research in software development. *Behaviour & Information Technology*, *41*, 1952 - 1972. doi:10.1080/0144929X.2021.1909141

Kotlarsky, J., and Oshri, I., and van Hillegersberg, J., & and Kumar, K. (2008). Globally distributed component-based software development: An exploratory study of knowledge management and work division Knowledge Processes in Globally Distributed Contexts. In J. Kotlarsky, I. and Oshri, & P. C. and van Fenema (Eds.), (pp. 159-184). London: Palgrave Macmillan UK. doi:10.1057/9780230582408_7

Kotlarsky, J., & Oshri, I. (2005). Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, *14*, 37-48. doi:10.1057/palgrave.ejis.3000520

Kraut, R., & Streeter, L. (1995). Coordination in software development. *Commun. ACM*, *38*, 69-81. doi:10.1145/203330.203345

Kwan, I., Schröter, A., & Damian, D. (2011). Does Socio-Technical Congruence Have an Effect on Software Build Success? A Study of Coordination in a Software Project. *IEEE Transactions on Software Engineering*, *37*, 307-324. doi:10.1109/TSE.2011.29

Lambiase, S., Catolino, G., Pecorelli, F., Tamburri, D., Palomba, F., Heuvel, W.-j. V. D., & Ferrucci, F. (2024). An Empirical Investigation Into the Influence of Software Communities' Cultural and Geographical Dispersion on Productivity. *J. Syst. Softw.*, *208*, 111878. doi:10.1016/j.jss.2023.111878

Lewin, A., Massini, S., & Peeters, C. (2009). Why are companies offshoring innovation? The emerging global race for talent. *Journal of International Business Studies*, *40*, 901-925. doi:10.1057/JIBS.2009.64

Lucie, L., & Hana, U. (2011). Staff Turnover as a Possible Threat to Knowledge Loss. *2011*, 84-98. Retrieved from https://www.semanticscholar.org/paper/b9eeab93d7a9855c8e08f75efc7a5717c4d1e5b6

Marks, M., Mathieu, J., & Zaccaro, S. (2001). A Temporally Based Framework and Taxonomy of Team Processes. *Academy of Management Review*, *26*, 356-376. doi:10.5465/AMR.2001.4845785

Vailshery, L. S. (2024). Number of Software Developers worldwide 2018-2024. Retrieved from https://www.statista.com/statistics/627312/worldwide-developer-population/

O'Connell, M., & Kung, M.-C. (2007). The cost of employee turnover. *Industrial management*, *49*(1). Retrieved from https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=00198471&AN=25205085&h=dLS+vIuzQeQUTS2J1HVd7EjIS6cDzkKwlK+D89+QhfV1sfZfBbaJgn8gU/xTnbOQKCr9DYYx4WDk2UvQIPC0/Q==&crl=c

O'Leary, M., & Mortensen, M. (2009). Go (Con)Figure: Subgroups, Imbalance, and Isolates in Geographically Dispersed Teams. *Decision Making*. doi:10.1287/orsc.1090.0434

Oliveira, E., Fernandes, E., Steinmacher, I., Cristo, M., Conte, T., & Garcia, A. F. (2020). Code and commit metrics of developer productivity: a study on team leaders perceptions. *Empirical Software Engineering*, *25*, 2519 - 2549. doi:10.1007/s10664-020-09820-z

Oliveira, E., Viana, D., Cristo, M., & Conte, T. (2017). *How have software engineering researchers been measuring software productivity?-a systematic mapping study*.

Park, T., & Shaw, J. (2013). Turnover rates and organizational performance: a meta-analysis. *The Journal of applied psychology*, *98 2*, 268-309. doi:10.1037/a0030723

Pee, L. G., Kankanhalli, A., Tan, G., & Tham, Z.-C. (2014). Mitigating the Impact of Member Turnover in Information Systems Development Projects. *IEEE Transactions on Engineering Management*, *61*, 702-716. doi:10.1109/TEM.2014.2332339

Pendharkar, P., & Rodger, J. (2007). An empirical study of the impact of team size on software development effort. *Information Technology and Management*, *8*, 253-262. doi:10.1007/s10799-006-0005-3

Pendharkar, P., & Rodger, J. (2009). The relationship between software development team size and software development cost. *Commun. ACM*, *52*, 141-144. doi:10.1145/1435417.1435449

Ramasubbu, N., & Balan, R. (2007). Globally distributed software development project performance: an empirical analysis., 125-134. doi:10.1145/1287624.1287643

Rodríguez-García, D., Sicilia, M., García-Barriocanal, E., & Harrison, R. (2012). Empirical findings on team size and productivity in software development. *J. Syst. Softw.*, *85*, 562-570. doi:10.1016/J.JSS.2011.09.009

Scott, E., Charkie, K. N., & Pfahl, D. (2020). *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) Productivity, Turnover, and Team Stability of Agile Teams in Open-Source Software Projects*.

Stefan, W. A., & Ruhe, M. (2018). A Systematic Review of Productivity Factors in Software Development. *CoRR*, *abs/1801.06475*. Retrieved from http://arxiv.org/abs/1801.06475

Subramaniam, R., & Nakkeeran, S. (2016). Identifying the Impact of Team Size on Team Work Quality (TWQ) in Software Teams through an Empirical Approach. *Asian Journal of Research in Social Sciences and Humanities*, *6*, 151-171. doi:10.5958/2249-7315.2016.00051.4

VanVoorhis, C. R. W., & Morgan, B. L. (2007). Understanding power and rules of thumb for determining sample sizes. *Tutorials in quantitative methods for psychology*, *3*(2), 43-50.

Walston, C. E., & Felix, C. P. (1977). A method of programming measurement and estimation. *IBM Sys. J.*, *16*, 54-73.

Yadav, P., & Marwah, C. (2015). The Concept of Productivity. Retrieved from https://www.semanticscholar.org/paper/1514f4aae0207bd2cfe5305604662b7f1f2e34a5

APPENDIX A: IRB NHRSC LETTER

University and Medical Center Institutional Review Board
Office of Research
300 E. 1st Street
East Carolina University | Greenville, NC 27834
252-744-2914 office | umcirb@ecu.edu

Not Human Subject Research Certification

From: Social Behavioral IRB
To: Lee Eason
Date: 02/11/2026

To Whom It May Concern:

This activity was reviewed on July 22, 2025, by the UMCIRB Director. Lee Eason, a student in the Department of Technology Systems at East Carolina University, was involved in a study to offer practical guidelines for optimal team sizes in geographically distributed software teams to reduce productivity and knowledge loss caused by turnover using data obtained from a software engineering company, BlueOptima. At the time of the review, Mr. Eason confirmed there would be no interaction or intervention with human subjects and no identifiable data was provided in the dataset. As such, Mr. Eason's activity did not meet the definition of "human subject" research and no local IRB approval was necessary.

If you have questions or require additional information, please feel free to contact the UMCIRB office at 252-744-2914.

Sincerely,

Neil Gilbird

Date 2/12/2026

Neil Gilbird, MPH, CIP
Director, Human Research Protections
University & Medical Center Institutional Review Board
East Carolina University
Willis Building, 300 E. 1st Street, Mailstop 682
Greenville, NC 27858
(252)744-3515
gilbirda@ecu.edu

APPENDIX B: VALIDITY OF SAMPLE PLOTS

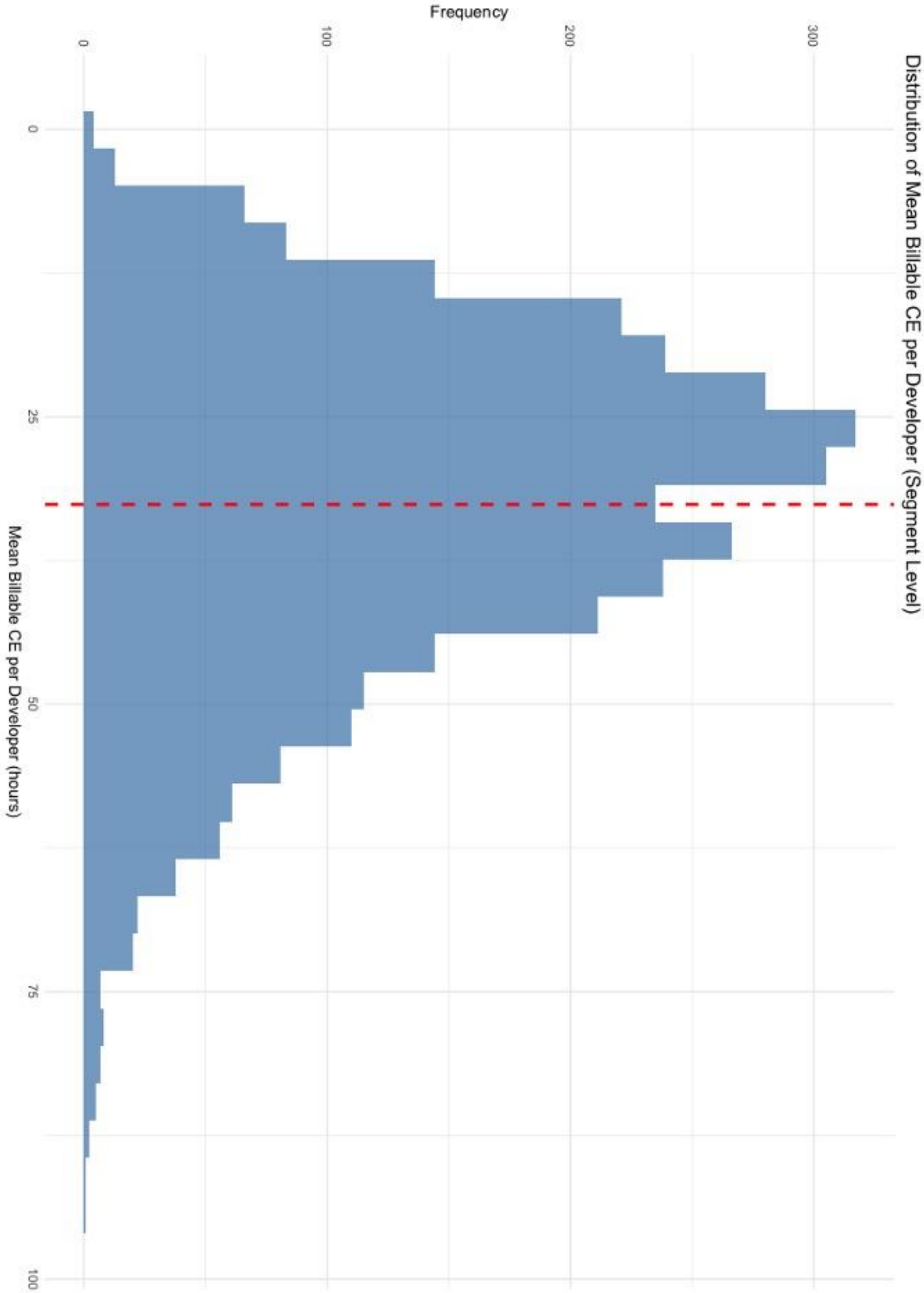

Figure A1: Distribution of Mean CE per Developer

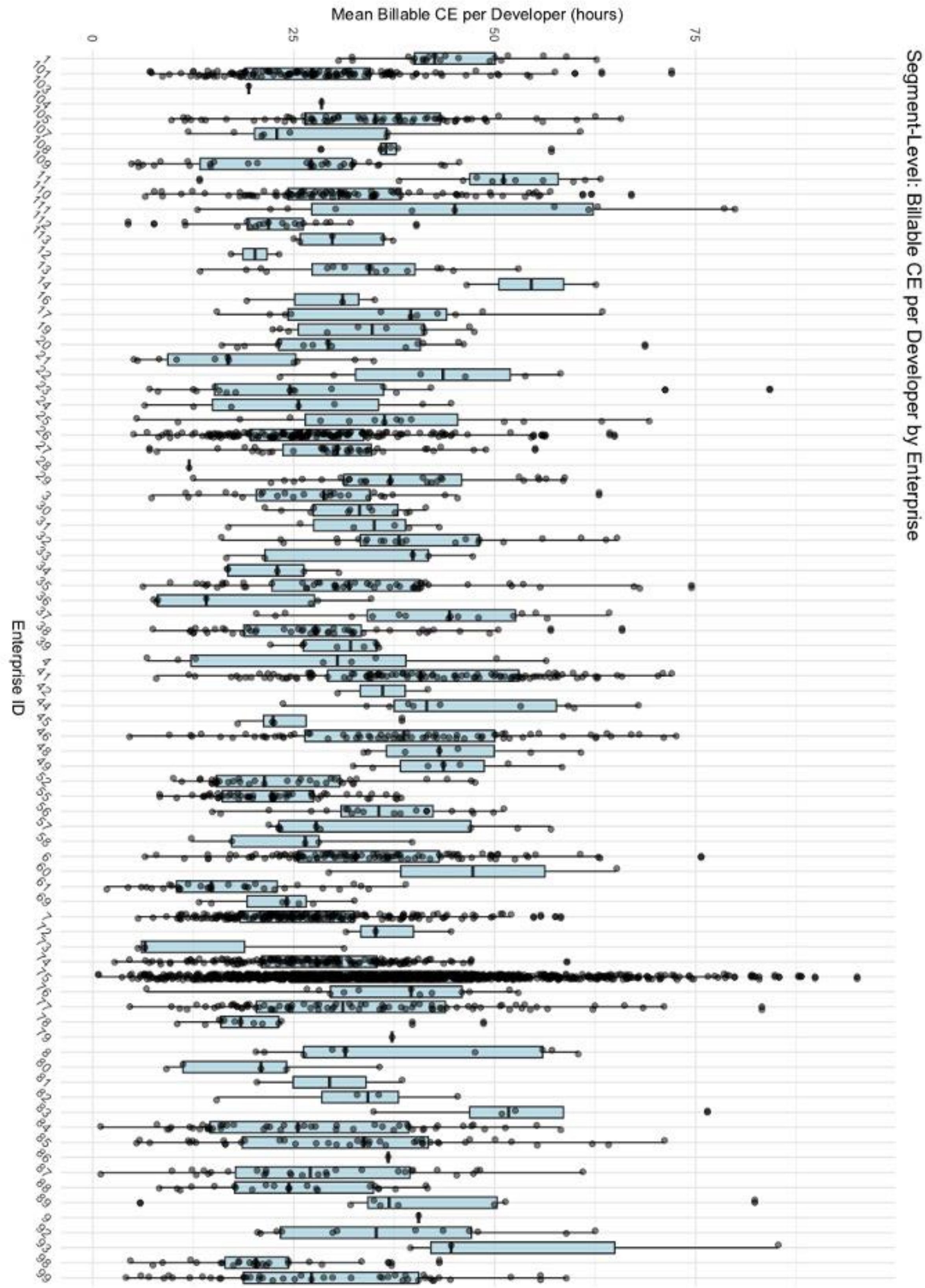

Figure A2: Team Level CE per Developer by Enterprise

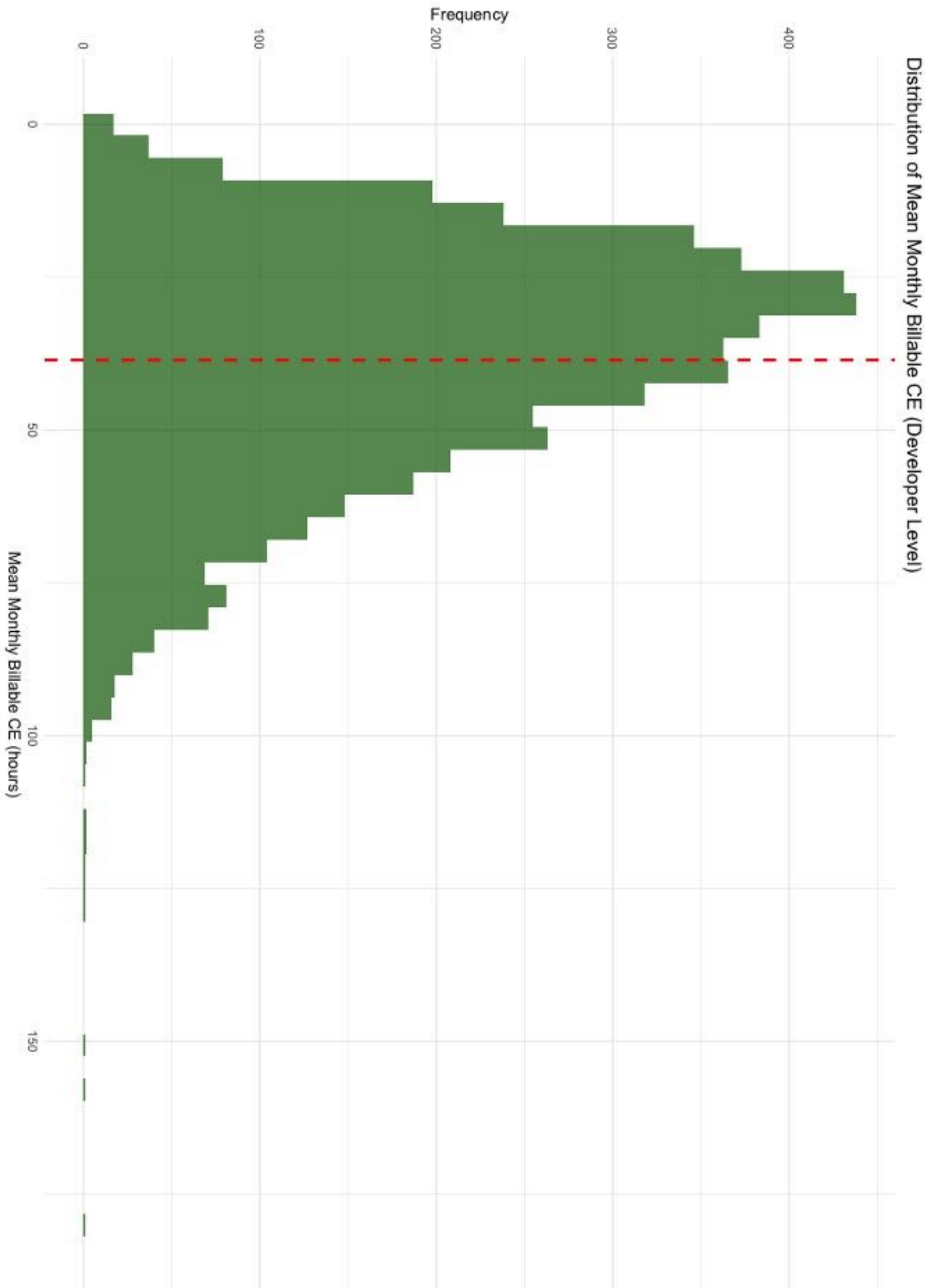

Figure A3: Team level Aberrant CE per Developer by Enterprise

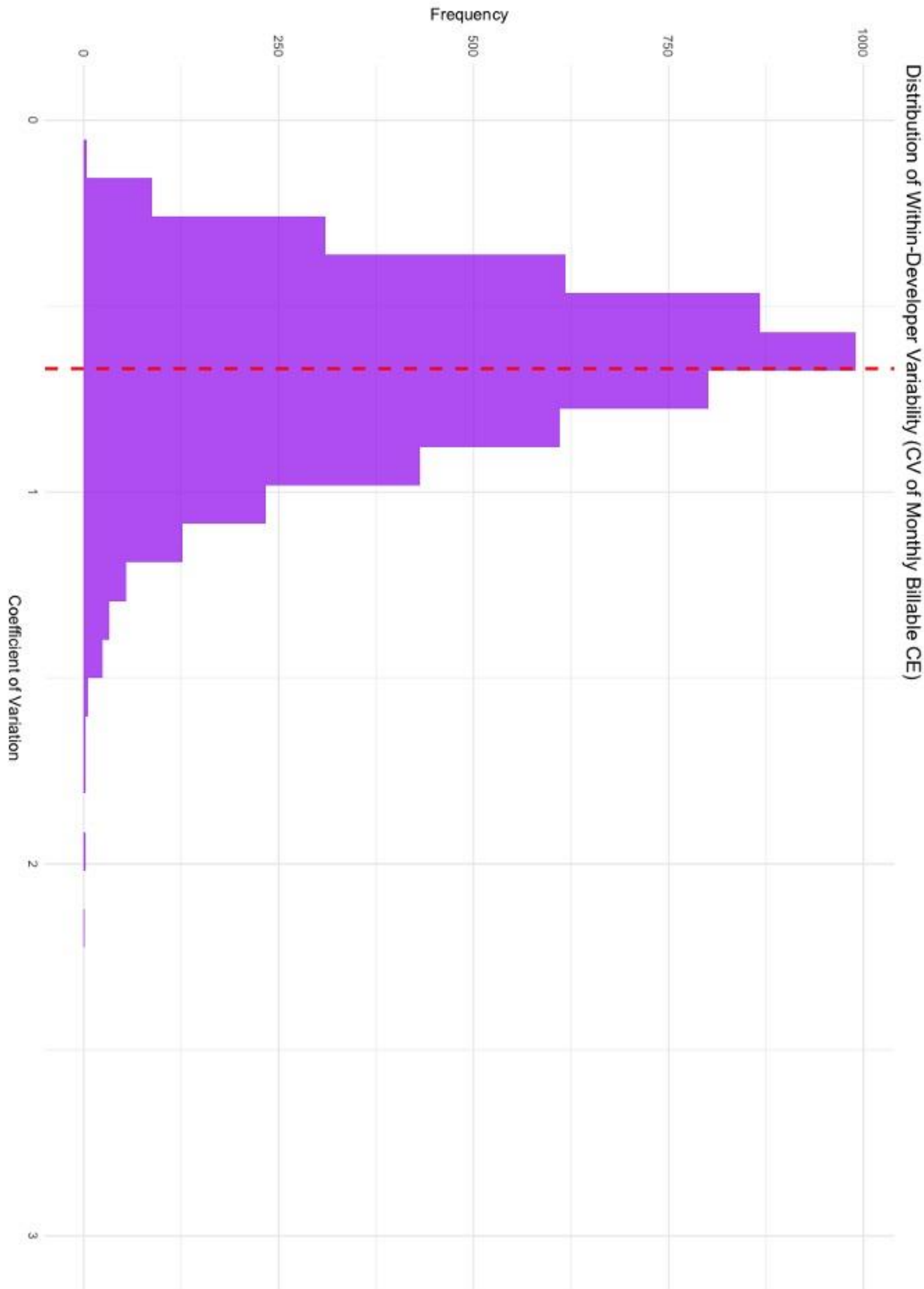

Figure A4: Distribution of Within-Developer Variability

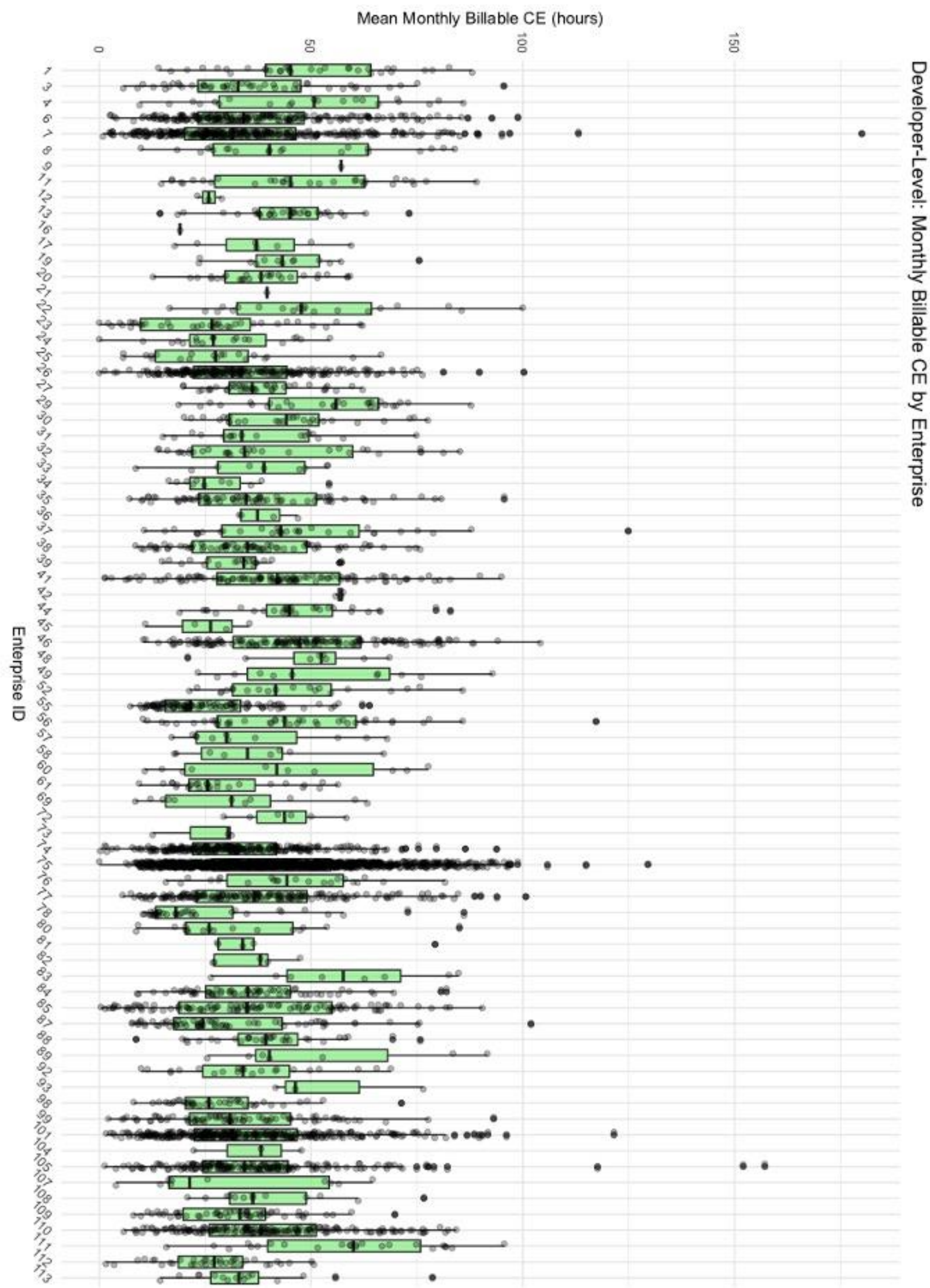

Figure A5: Developer Monthly CE by Enterprise

APPENDIX C: DESCRIPTION OF MEASURES

Recovery Outcomes

Two recovery metrics were computed monthly for each backfill engineer over the first six months of joining their new team.

Recovery VS Prior Engineer: This measures the new engineer against the average productivity of the engineer they are replacing for the six months prior to their departure.

$$\text{RecPrior}i,m = 100 * \left(\frac{\text{CE}i,m}{\text{CEPriorEng}}\right)$$

Where *CEi,m* is the Coding Effort of the replacement engineer *i* in month *m,* and *CEPriorEng* is the leaver's six-month pre-departure average.

Recovery VS Prior Team Average: This captures the per-developer average monthly CE and compares the new engineer's CE to it. This test helps ensure that if the prior engineer was an outlier (very high or very low performer), the new engineer can be fairly measured.

$$\text{RecTeam}i,m = 100 * \left(\frac{\text{CE}i,m}{\text{CEPriorTeam}}\right)$$

Where the denominator is the team's six month per-developer average before Turnover Start.

Predictors

Team Size in four categories: Small (3-4), Medium (5-7), Large (8-10), and Very Large (11+)

These cut points were chosen to balance statistical power across cells while preserving meaningful organizational distinctions. All categories exceeded the minimum recommended cell size of 30 observations, with the smallest containing 145 events in Month 1

Temporal Distribution in three categories based on temporal variance (range = 0–0.831, $\tilde{x} = 0.1113$, $s = 0.1255$):

Not Distributed (low variance), Some Distribution (moderate variance), and Highly Distributed (high variance). These classifications were based on practical distinctions in working-hour overlap.

Month Since Join: factor variable with six levels (1-6)

Control and Quality Checks.

- All included turnover events maintained stable team size during the leave/join transition to avoid confounding with strategic resizing.
- Only external turnover events were included; internal transfers were excluded.
- Key variables had no missing data; data completeness was 100%.
- All team size × temporal distribution combinations exceeded minimum recommended sample sizes and displayed excellent statistical power for medium and large effects.

Statistical Approach

Mixed-Effects Framework

The analysis employed mixed-effects models to account for the nested data structure where monthly observations are clustered within turnover events, which are clustered within teams and enterprises. The base model specification was:

log(Recovery) ~ TeamSize × TemporalDistribution × Month + (1|EventID) + (1|TeamID)

Recovery ratios were log-transformed to address skewness and heteroscedasticity identified through residual analysis. Random intercepts for Event ID and Team ID account for unobserved heterogeneity at these clustering levels.

Statistical Methods

Diagnostic testing revealed violations of normality (100% of month-specific tests) and homoscedasticity (100% of month-specific tests) assumptions. This study addressed these violations through multiple approaches which will be expanded upon in the following pages:

1. Log transformation of outcome variables to reduce skewness
2. Factorial ANOVA (Type I sum of squares) as primary analysis method
3. Type II ANOVA conducted as robustness validation for cell size imbalance
4. HC3 robust standard errors to address heteroscedasticity
5. ARTool aligned rank transformation as non-parametric alternative
6. Benjamini-Hochberg false discovery rate (FDR) correction for multiple testing across 6 months

Hypothesis Testing Framework

Three pre-registered hypotheses were tested using a systematic month-by-month analysis framework with appropriate corrections for multiple testing. Each hypothesis was evaluated using factorial ANOVA, with Type II ANOVA performed as a robustness validation to confirm cell size imbalance did not affect conclusions. Benjamini-Hochberg false discovery rate correction was applied within each hypothesis-outcome combination across the six-month follow-up period. This type of correction ensures that positive results are not due to a buildup of error rates across sequential tests.

Robustness was assessed as the percentage of months showing statistically significant effects after FDR correction. Evidence levels were established based on both statistical robustness and effect size magnitude:

- **Strong evidence**: ≥80% robustness, $\eta^2 > 0.01$

- **Moderate evidence**: ≥67% robustness, $\eta^2 > 0.006$
- **Weak evidence**: ≥50% robustness
- **Insufficient evidence**: <50% robustness

Sample Validation

The final analytical sample consisted of 44,603 monthly observations from 685 external backfill events across 459 teams and 50 enterprises. This provides exceptional statistical power to detect small-to-medium effect sizes (Cohen's $f \geq 0.06$) with >80% power in all team size × temporal distribution combinations.

FDR correction was used because of the possibility of error rates compounding as we tested across the six-month timeframes. This way we can be confident that the results are not simply the result of accumulating errors.

All cell sizes exceeded 145 observations, well above the minimum threshold of 30 required for robust factorial analysis. The 7:1 ratio between largest and smallest cells reflects natural organizational patterns in observational data and remains within acceptable bounds for factorial ANOVA analysis. To address potential concerns about this imbalance, Type II ANOVA was conducted as a robustness check (see validation results), confirming that conclusions remained consistent across both analytical approaches (VanVoorhis & Morgan, 2007).

APPENDIX D: HYPOTHESIS TESTING P VALUES

month	comparison	team_size_p	team_size_f	team_size_eta2	temporal_p	temporal_f	temporal_eta2	interaction_p	interaction_f	interaction_eta2	n
1	vs_prior	0.00000	11.0795	0.00503	0.28800	1.2450	0.0004	0.000000280	6.8652	0.0062	6538
1	vs_team	0.00000	11.5612	0.00524	0.00409	5.5050	0.0017	0.000000008	8.1839	0.0074	6538
2	vs_prior	0.32836	1.1475	0.00036	0.00917	4.6940	0.0010	0.000000000	27.7150	0.0176	9272
2	vs_team	0.00231	4.8318	0.00154	0.04639	3.0718	0.0007	0.000000000	25.1378	0.0160	9272
3	vs_prior	0.00000	12.1795	0.00420	0.00000	22.7084	0.0052	0.000000000	14.8081	0.0102	8543
3	vs_team	0.00760	3.9821	0.00138	0.10039	2.2993	0.0005	0.000000000	14.2948	0.0099	8543
4	vs_prior	0.00000	23.5755	0.00894	0.00000	32.2440	0.0082	0.000000000	15.4820	0.0117	7694
4	vs_team	0.00000	28.7753	0.01101	0.17424	1.7477	0.0004	0.000000000	11.3586	0.0087	7694
5	vs_prior	0.02524	3.1109	0.00130	0.00168	6.3920	0.0018	0.000000000	19.5358	0.0163	7078
5	vs_team	0.08666	2.1933	0.00092	0.00454	5.3993	0.0015	0.000000000	14.5850	0.0122	7078
6	vs_prior	0.22722	1.4464	0.00078	0.66443	0.4089	0.0001	0.000000000	15.3877	0.0166	5478
6	vs_team	0.00303	4.6425	0.00249	0.00063	7.3827	0.0026	0.000000000	17.0101	0.0182	5478

Table 2: ANOVA Type I P Values

Month	Test Description	Hypothesis	Type II p	BH q	Partial eta2	n	Sig. p	Sig. q
1	recovery_vs_prior	H1: Size	<0.00001	0	0.005067	6538	TRUE	TRUE
1	recovery_vs_prior	H2: Distribution	0.288001	0.345601	0.000381	6538	FALSE	FALSE
1	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.006272**	**6538**	**TRUE**	**TRUE**
2	recovery_vs_prior	H1: Size	0.556927	0.556927	0.000372	9272	FALSE	FALSE
2	recovery_vs_prior	H2: Distribution	0.009172	0.013758	0.001013	9272	TRUE	TRUE
2	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.017641**	**9272**	**TRUE**	**TRUE**
3	recovery_vs_prior	H1: Size	<0.00001	0	0.004265	8543	TRUE	TRUE
3	recovery_vs_prior	H2: Distribution	<0.00001	0	0.005296	8543	TRUE	TRUE
3	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.010307**	**8543**	**TRUE**	**TRUE**
4	recovery_vs_prior	H1: Size	<0.00001	0	0.009123	7694	TRUE	TRUE
4	recovery_vs_prior	H2: Distribution	<0.00001	0	0.008325	7694	TRUE	TRUE
4	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.011948**	**7694**	**TRUE**	**TRUE**
5	recovery_vs_prior	H1: Size	0.003613	0.005419	0.001319	7078	TRUE	TRUE
5	recovery_vs_prior	H2: Distribution	0.001685	0.00337	0.001806	7078	TRUE	TRUE
5	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.016318**	**7078**	**TRUE**	**TRUE**
6	recovery_vs_prior	H1: Size	0.209008	0.25081	0.000793	5478	FALSE	FALSE
6	recovery_vs_prior	H2: Distribution	0.664427	0.664427	0.00015	5478	FALSE	FALSE
6	**recovery_vs_prior**	**H3: Interaction**	**<0.00001**	**0**	**0.01661**	**5478**	**TRUE**	**TRUE**
1	recovery_vs_team	H1: Size	1.00E-06	3.00E-06	0.005287	6538	TRUE	TRUE
1	recovery_vs_team	H2: Distribution	0.004085	0.009076	0.001684	6538	TRUE	TRUE
1	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.007468**	**6538**	**TRUE**	**TRUE**
2	recovery_vs_team	H1: Size	0.000624	0.001248	0.001563	9272	TRUE	TRUE
2	recovery_vs_team	H2: Distribution	0.046386	0.069579	0.000663	9272	TRUE	FALSE
2	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.016027**	**9272**	**TRUE**	**TRUE**
3	recovery_vs_team	H1: Size	0.013318	0.015982	0.001398	8543	TRUE	TRUE
3	recovery_vs_team	H2: Distribution	0.100392	0.12047	0.000539	8543	FALSE	FALSE
3	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.009954**	**8543**	**TRUE**	**TRUE**
4	recovery_vs_team	H1: Size	<0.00001	0	0.011113	7694	TRUE	TRUE
4	recovery_vs_team	H2: Distribution	0.174238	0.174238	0.000455	7694	FALSE	FALSE
4	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.008794**	**7694**	**TRUE**	**TRUE**
5	recovery_vs_team	H1: Size	0.135799	0.135799	0.00093	7078	FALSE	FALSE
5	recovery_vs_team	H2: Distribution	0.004538	0.009076	0.001526	7078	TRUE	TRUE
5	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.012233**	**7078**	**TRUE**	**TRUE**
6	recovery_vs_team	H1: Size	0.000878	0.001317	0.002542	5478	TRUE	TRUE
6	recovery_vs_team	H2: Distribution	0.000628	0.003768	0.002694	5478	TRUE	TRUE
6	**recovery_vs_team**	**H3: Interaction**	**<0.00001**	**0**	**0.01833**	**5478**	**TRUE**	**TRUE**

Table 3: ANOVA Type II with FDR Correction Results

APPENDIX E: GEOGRAPHIC VS TEMPORAL DISTRIBUTION

Supplementary Investigation of Physical Proximity Effects

Following the finding that temporal distribution (time zone alignment) moderates team size effects on turnover recovery, the data was analyzed to investigate whether geographic distribution (number of cities spanning a team) provided additional explanatory power beyond temporal alignment. This analysis tested whether physical or regional proximity within the same time zone affected recovery outcomes, which would have implications for remote work policies and team formation decisions. It also ensures that the results of the study were not confounded by variables like co-location.

Methodology

The subset of teams classified as "Not Distributed" or "Some Distribution" (n=437 events, 27,945 monthly observations) was analyzed to isolate geographic effects from temporal effects. Geographic distribution was measured using city location data at the replacement month and operationalized three ways: (1) geographic dispersion index (cities per team member), (2) binary classification (single city vs. multi-city), and (3) ordinal classification (single, two, or 3+ cities).

Mixed-effects models tested whether geographic distribution and its interaction with team size predicted Month 1 recovery outcomes, controlling for temporal distribution and enterprise clustering:

```
log(Recovery) ~ Geographic × TeamSize × Temporal + (1|EnterpriseID)
```

To address the COVID-19 remote work confound, we conducted temporal validation by testing the same models separately for each year (2020-2025) and for three periods: Peak COVID (2020-2021), Transition (2022), and post-COVID (2023+).

Results

Initially, pooled analysis (2020-2025 combined) suggested interaction effects between geographic distribution and team size (categorical model: 100% robustness across 6 months, $p < 0.001$), with small teams appearing to benefit from single-city concentration and large teams from multi-city distribution.

However, year-by-year temporal validation revealed the effect was unstable and likely artifactual as demonstrated in Table 3.

Year	Sample Size	Interaction p-value	Small Team Recovery Difference
2020	44 events	$p < 0.001$	Single city +101 points
2021	128 events	$p = 0.002$	Multi-city +37 points (reversed)
2022	128 events	$p < 0.001$	Single city +71 points
2023	53 events	$p < 0.001$	Tied (+4 points)
2024	50 events	$p = 0.343$ (NS)	Multi-city +13 points
2025	9 events	$p = 0.782$ (NS)	Insufficient data

Table 4: Geographic effects on Turnover by Year

Period Analysis:

- Peak COVID (2020-2021): Significant main effect ($p = 0.022$), but multi-city small teams recovered better (144% vs. 115%), opposite the expected direction
- Transition (2022): Strongest interaction ($p < 0.001$), single-city advantage present
- Post-COVID (2023+): Weak interaction ($p = 0.030$), no small team difference (121% vs. 120%, $p = 0.947$)

Temporal Trend: Effect significance declined from 100% of years in 2020-2021 to 33% in 2023-2024.

Conclusion

The geographic distribution effect failed temporal robustness testing. The effect was temporally unstable, reversed direction between years, and disappeared in recent data (2024-2025). This pattern is inconsistent with either physical proximity mechanisms (which should strengthen post-COVID as offices reopened) or cultural/regional mechanisms (which should remain stable). The most rational explanation is sample composition artifacts related to COVID-era hiring disruptions, where multi-city teams during 2020-2022 may have represented struggling teams forced to hire remotely while healthy teams concentrated hiring locally.

Geographic distribution was excluded from main analyses due to lack of temporal robustness. The study focuses on temporal distribution (time zone alignment), which demonstrated consistent effects across all time periods and represents the primary coordination mechanism moderating team size effects on turnover recovery.