An Exploratory Study of Productivity Perceptions in Software Teams

Ruvimova, Anastasia ; Lill, Alexander ; Gugler, Jan ; Howe, Lauren ; Huang, Elaine ; Murphy, Gail ; Fritz, Thomas

Abstract: Software development is a collaborative process requiring a careful balance of focused individual effort and team coordination. Though questions of individual productivity have been widely examined in past literature, less is known about the interplay between developers' perceptions of their own productivity as opposed to their team's. In this paper, we present an analysis of 624 daily surveys and 2899 self-reports from 25 individuals across five software teams in North America and Europe, collected over the course of three months. We found that developers tend to operate in fluid team constructs, which impacts team awareness and complicates gauging team productivity. We also found that perceived individual productivity most strongly predicted perceived team productivity, even more than the amount of team interactions, unplanned work, and time spent in meetings. Future research should explore how fluid team structures impact individual and organizational productivity.

DOI: https://doi.org/10.1145/3510003.3510081

Originally published at:
Ruvimova, Anastasia; Lill, Alexander; Gugler, Jan; Howe, Lauren; Huang, Elaine; Murphy, Gail; Fritz, Thomas (2022). An Exploratory Study of Productivity Perceptions in Software Teams. In: International Conference on Software Engineering (ICSE'22), Pittsburgh, PA, USA, 22 May 2022 - 27 May 2022. ACM, 1-13.
DOI: https://doi.org/10.1145/3510003.3510081

An Exploratory Study of Productivity Perceptions in Software Teams

Anastasia Ruvimova
University of Zurich
Zurich, Switzerland
ruvimova@ifi.uzh.ch

Alexander Lill
University of Zurich
Zurich, Switzerland
lill@ifi.uzh.ch

Jan Gugler
University of Zurich
Zurich, Switzerland
jan.gugler@gmail.com

Lauren Howe
University of Zurich
Zurich, Switzerland
lauren.howe@business.uzh.ch

Elaine Huang
University of Zurich
Zurich, Switzerland
huang@ifi.uzh.ch

Gail Murphy
University of British Columbia
Vancouver, Canada
murphy@cs.ubc.ca

Thomas Fritz
University of Zurich
Zurich, Switzerland
fritz@ifi.uzh.ch

ABSTRACT

Software development is a collaborative process requiring a careful balance of focused individual effort and team coordination. Though questions of individual productivity have been widely examined in past literature, less is known about the interplay between developers' perceptions of their own productivity as opposed to their team's. In this paper, we present an analysis of 624 daily surveys and 2899 self-reports from 25 individuals across five software teams in North America and Europe, collected over the course of three months. We found that developers tend to operate in fluid team constructs, which impacts team awareness and complicates gauging team productivity. We also found that perceived individual productivity most strongly predicted perceived team productivity, even more than the amount of team interactions, unplanned work, and time spent in meetings. Future research should explore how fluid team structures impact individual and organizational productivity.

CCS CONCEPTS

• **Software and its engineering → Programming teams**; Software development methods.

KEYWORDS

Team, Productivity, Software Developer, User Study

1 INTRODUCTION

Software development organizations face constant pressure to increase the speed, volume, and quality of their software production. As companies may not have the resources to address these goals by increasing team size, organizations must look to improving the *productivity* of those involved in producing the software. A desire to improve software development productivity is not new: Boehm wrote eloquently about the need for improved software productivity in the late 1980s [8]. Many have since studied appropriate measures for software development productivity [22, 27] and factors that impact software development productivity (e.g., [29]). Recognizing the significant impact individuals have on the productivity of a software development organization, recent research has placed more emphasis on the individuals involved in the development process, and in particular, on how they view their own productivity [31, 40]. Recent research also suggests that the measurement of productivity needs to include multiple dimensions of individual metrics, including such aspects as satisfaction, activity, and flow [29, 35].

However, individuals do not work alone on a software development project. Software development teams, comprising multiple individuals, are at the heart of most complex software development projects. Individuals constantly must make trade-offs to balance making progress on their individual work versus helping a teammate (e.g., [39]). Despite the necessary and central role of teams in software development, surprisingly few studies focus on team software development productivity, and those that do, study a snapshot of development. For instance, Lakhanpa et al., using a one-time survey, found team cohesion to be the dominating factor when investigating team performance [32]. These studies leave open many questions about the interplay between individual and team productivity. We focus our analysis on the following:

RQ1: What is the relationship between perceived team and individual productivity in software teams?
RQ2: Which team factors affect individual and team productivity perceptions?

To investigate these questions, we undertook a three-month multi-modal study in which we collected daily self-reports from 25 software developers within five companies in North America and Europe. These surveys, collected on a daily and bi-hourly basis, asked participants to report their individual and team productivity, and to reflect on a variety of factors which affected their productivity that day, such as team interactions, time spent in meetings and the amount of unplanned work they had done. Additionally, we held a series of three interviews with each developer, spaced one month apart, to capture any changes in their team or work, and to ask broader questions about their perceptions of productivity and teamwork. In total, we collected 624 daily surveys, and 2899 bi-hourly self-reports.

Applying both quantitative and qualitative analysis approaches, we found that developers tend to operate in fluid team constructs,

ICSE '22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9221-1/22/05.
https://doi.org/10.1145/3510003.3510081

which impacts team awareness and complicates measurements of team productivity. We also found that perceived individual productivity most strongly predicted perceived team productivity, even more than the amount of team interaction, unplanned work, and time spent in meetings. We discuss the implications of these insights and what they may suggest for future research about team productivity. This paper makes three contributions:

- It presents a methodology for undertaking a longitudinal study of software development team productivity that incorporates both individual and team observations.
- It presents novel findings about developers' perceptions of their team and the team's productivity.
- It provides a replication package with the surveys, anonymized daily and bi-hourly self-reported data, and analysis scripts.

Section 2 relates this study and its findings to earlier efforts. Section 3 presents the study method and data analysis approaches employed. Section 4 presents the results of the study, which are discussed in Section 5. Section 6 reports threats to validity and Section 7 summarizes the work.

2 RELATED WORK

Traditionally, productivity is defined as the ratio of outputs over inputs, where output is value produced and input is time or other costs invested [47]. While this definition can work well in some contexts, like manufacturing, it is not easily applied to software development as outputs and inputs tend to be difficult to quantify. We consider how our work builds on and compares to quantifying software development productivity, as well as individual developer and team productivity. As our study was conducted during COVID-19, which provides a unique work situation, we also consider how our study compares to other efforts to study software development during COVID-19.

Quantifying Software Development Productivity
Two primary approaches are employed in the effort to quantify development work: automated process metrics and self-reported perceived productivity. Automated process metrics typically rely on activity metrics, such as the lines of source code (SLOC) written [22, 55], function points [1, 27], or change requests fulfilled [14, 43]. These 'objective measures' may only be capturing a small part of a developer's work [52] and may not be commensurate with the actual value produced or progress made. Some question the very concept of measuring productivity [31], noting that people tend to optimize for whatever metric is being used—a phenomenon known as Goodhart's law [15, 24]. These metrics may underestimate work done for tasks that are essential, but that do not produce immediate tangible output, such as training for a new skill or mentoring a new hire. To overcome these challenges, recent efforts have considered perceived productivity and what impacts it [29, 40, 41]. In this paper, we use self-reported productivity to explore the interplay between individual and team productivity.

Individual Developer Productivity
Researchers have investigated how the perceived productivity of individual developers relates to a number of factors. For example, job aspects that motivate developers or bring enjoyment correlate with higher productivity and job satisfaction [5, 29]. Positive affect states, indicating happier moods, have been shown to boost developer problem-solving skills and productivity [2, 25, 40, 44]. Likewise, developer's moods were shown to influence performance on certain tasks, such as debugging [28]. Interruptions, on the other hand, are often cited as one of the most prominent factors influencing productivity, increasing anxiety and error rate, and reducing performance [3, 18, 34, 46]. Our study differs from these efforts in focusing on how perceptions of individual and team productivity relate.

Team Productivity
Other research has looked into software development team productivity. Several aspects have been shown to potentially predict the success of software teams, including organization, teamwork structures, and cohesion. Early research by Boehm [9] indicated that productivity on a software development project is most affected by who develops the system and how well they are organized and managed as a team. Bendifallah and Scacchi [6] found that variation in teamwork productivity and quality could best be explained in terms of recurring teamwork structures, categorized by patterns of interaction. Lakhanpa et al. found team cohesion to be the dominating factor when investigating the influence of team cohesion, experience, and capability on team performance [32]. A recent literature review [12] revealed 37 factors to be pertinent to team productivity in software development, including individual characteristics such as work experience [21] and skill/competence [36, 45]. Among interaction factors, collaboration [16, 17], team member availability [21, 36] and ease of communication [54, 56] were found to be predictive of team productivity. Ko notes that there are four lenses through which to think about productivity in software development: the individual, team, market, and the full-spectrum, and suggest that a single measure may not suffice for capturing performance [30]. Though past work has looked at individual and team productivity separately, we aim to bridge this gap by exploring how individuals perceive their team's productivity and how they understand their own productivity in the team context.

Remote Work and the Pandemic
Our study took place during the COVID-19 pandemic, with the result that participants worked primarily from their home offices. While working from home is often claimed to improve productivity [13, 20, 37], a study by Russell et al. found that working from home is associated with greater levels of both work pressure and work-life conflict [49, p. 92] because work intrudes into developers' home lives through working unpaid overtime, thinking about work in off hours, exhaustion, and sleeplessness [26]. It should be noted that working from home in a pandemic is not representative of regular remote work situations in which the employee might work from a home office or elsewhere outside the office. A recent study explored how human and organizational factors influence software team productivity in the pandemic, finding that the main factors

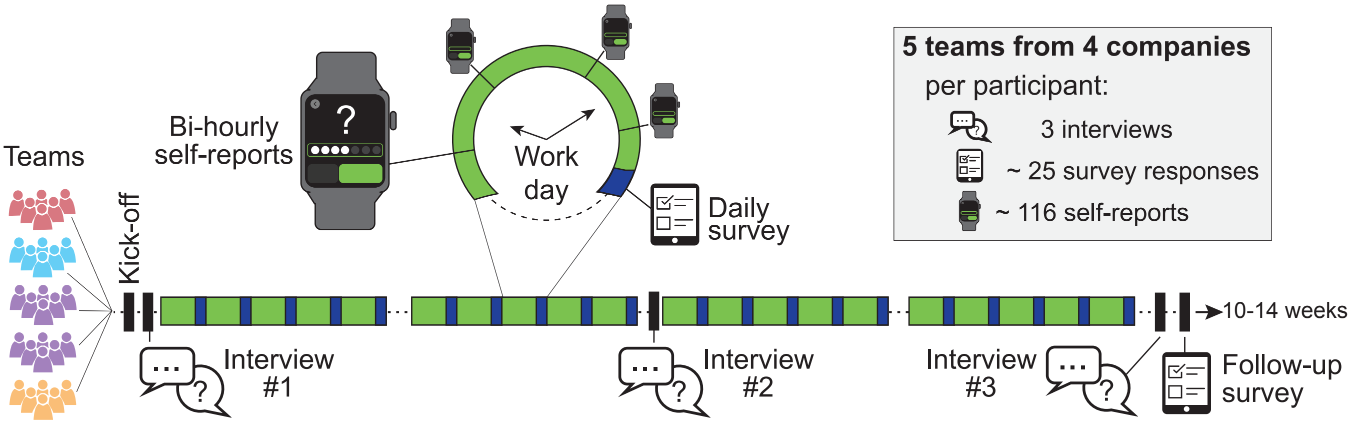


Figure 1: Study timeline

are external interruptions, environment adaptation, and emotional issues [7]. Ralph et al. investigated how working from home in the pandemic affected software developer well-being and productivity [48]. Miller et al. conducted a study at a large software company to understand what team culture factors have been affected by the pandemic [42] and found that the ability to brainstorm with colleagues, difficulty communicating with colleagues, and satisfaction with interactions from social activities were important factors. Rather than looking at productivity from a single perspective (individual level, team level, or organizational), we explore the nuanced interaction between individual and team productivity perceptions, and the complex balance and trade-off of individual work and collaborative efforts.

3 STUDY DESIGN

As the activities related to software development can vary substantially week to week [38], a major objective for our study was to consider software developers and software development teams across several weeks. The study timeline is presented in Figure 1. We targeted a study duration of 8 weeks to capture a longer period of development (all teams participated for 10 to 14 weeks); as most teams were using an agile approach, this time span worked out to a few sprints. Given the length of the study and the need to gather input from the software developers, we needed to choose study instruments that allowed collection of data frequently; we used bi-hourly self-reports, daily surveys, and monthly interviews to gather data frequently while not causing our participants to disengage from survey fatigue. We describe the participants in our study, the survey instruments used and collected data, and report on the quantitative and qualitative analysis methods we applied to the data.

In the interest of transparency and reproducibility, we have posted the survey materials, deidentified quantitative data (from the daily surveys and bi-hourly self-reports), and the data analysis scripts on the Open Science Framework at [50]. We have not made public the qualitative data from the daily surveys or interviews due to the risk of deanonymization, but include the corresponding survey and interview material for replication.

3.1 Participants

We recruited 25 software developers from 5 teams of 4 companies across North America and Europe. All teams were agile and worked in various domains, ranging from time-tracking software to real-estate platforms. Team size ranged from 5 to 15 (mean=9.40), and on average, 53% of the developers of each team participated. We recruited these participants through word-of-mouth, social media, and other personal and professional contacts. These teams were considered to be software development teams if they identified their industry to be Software Development and if individual contributors spent at least half of their work time developing software. The amount of time spent on development work was later verified with the activity metrics given in the bi-hourly self-reports. Participants received a smart watch as compensation for their time and effort.

Out of the 25 software developers, 21 identified as male and 4 identified as female. They had a mean age of 36.22 years with a standard deviation (SD) of 10.26 years. Table 2 shows an overview of the participants and their respective teams, the participant's role and the amount of collected data for each participant. For simplicity, we categorized the 25 participating software developers into two roles: individual contributor (IC) and manager. A manager was considered to be anyone who plays a leadership or managerial role in the team.

3.2 Survey Instrument & Collected Data

To gain a broader perspective of productivity, this multi-modal study combined bi-hourly self-reports, daily surveys, monthly interviews, and post-study follow-up surveys. Though the minimum requirement for participation was 8 weeks, all five teams participated for 10 to 14 weeks (mean = 12.4, SD = 1.625). We collected 624 daily surveys and 2899 bi-hourly self-reports, with an average of 25 daily surveys and 116 bi-hourly self-reports per participant.

Bi-Hourly self-reports were submitted on a smart fitness watch by the participants. They were asked to report their individual and team productivity, stress level, and activity categories worked on in the last hour. Table 1 provides an overview of the questions and answer options, and Figure 2 shows the data entry screens of the smart watch application.

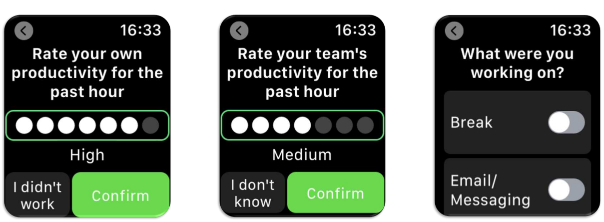
Table 1: Survey & Interview Questions

Question	**Answer Type**
Bi-Hourly Self-Reports	
Rate your **own productivity** for the past hour	Very low (1) to Very high (7)*
Rate your **team's productivity** for the past hour	Very low (1) to Very high (7)**
Rate your own **stress** level for the past hour	Very low (1) to Very high (7)***
What were you working on?	Activity Categories****
Daily Survey	
Individual Productivity	
How would you rate your **overall** productivity today?	Not Productive At All (1) to Extremely Productive (7)
Which **factors** may have had an effect on your productivity today?	Open-Ended
(Graph) How would you explain the **pattern** of your productivity ratings for today?	Open-Ended (Response to Graph)
Team Productivity	
How would you rate your team's **overall** productivity today?	Not Productive At All (1) to Extremely Productive (7)
(Graph) How would you explain the **pattern** of your team productivity ratings for today?	Open-Ended (Response to Graph)
Individual vs. Team Productivity	
If there are **difference** in your own and your team's productivity, how would you explain it?	Open-Ended
Interactions	
How would you rate the amount/frequency of interaction you had with your **supervisor** today?	No Interaction At All (1) to Very High (7)
How much of the interaction with your supervisor today was **related to work**?	Almost Entirely Not About Work (1) to Almost Entirely About Work (5)
How much did you interact with your **team** today?	Almost not at all (1) to Unusually Much (5)
How much of the interaction with your team today was **related to work**?	Almost Entirely Not About Work (1) to Almost Entirely About Work (5)
Other Factors	
How much time did you spend in **meetings** today?	Hour Buckets
How much time did you spend doing **unplanned work** today?	Time Entry
Describe the **nature** of the unplanned work - what was it, how did it get assigned to you?	Open-Ended
What task did you spend the most time on today?	Open-Ended
Did your tasks today require more **individual effort or team effort?**	Almost Entirely Individual Effort (1) to Almost Entirely Team Effort (5)
Was the majority of your time today spent working **remotely** (out of the office)?	Yes/No
How (positively) do feel about your **interruptions** today?	Very Negative (1) to Very Positive (7)
Would you like to share **anything else** about your day?	Open-Ended
Monthly Interviews	
Could you describe your **team?**	Open-Ended
Has your team context **changed** since you started the study?	
Could you describe the **nature of your work**?	
Has the nature of your work **changed** since you started the study?	
Do you work more individually or actively collaborate with others?	
How often do you communicate with your team and what does this communication look like?	
What would you say productivity means to you?	
How do you go about assessing your own productivity? For your team's?	
What factors do you feel have the biggest effect on your own productivity?	
What factors do you feel have the biggest effect on your team's productivity?	
What factors have the biggest effect on your stress?	
Are you currently working more remotely or from the office?	

* I didn't work, No Answer; ** I don't know, No Answer; *** Not sure, No Answer; **** Activity Categories: Break, Email/Messaging, Meeting, Planning, Software Development Work, Browsing, Read/Write Documents, Other Individual Tasks, Other Team Tasks; Some questions were shortened and reordered for clarity.

Table 2: Study Participants (IC = Individual Contributor)

Team / Participant	Developer Role	# Self Reports	# Daily Surveys
Team 1 – 13 weeks			
S1	IC	117	18
S2	IC	90	13
S3	IC	129	26
S4	Manager	114	19
S5	Manager	41	28
S6	Manager	91	29
Team 2 – 14 weeks			
S7	IC	185	9
S8	IC	29	12
S9	IC	177	45
S10	Manager	122	40
S11	Manager	92	20
S12	Manager	157	50
Team 3 – 10 weeks			
S13	IC	169	27
S14	IC	126	15
S15	IC	117	31
S16	IC	62	19
S17	IC	71	11
S18	Manager	157	39
S19	Manager	70	21
Team 4 – 11 weeks			
S20	IC	135	10
S21	IC	128	33
S22	IC	138	18
S23	Manager	46	22
Team 5 – 14 weeks			
S24	IC	200	50
S25	IC	136	19
∑ 25		2899	624



(a) **(b)** **(c)**

Figure 2: Screens of the smart watch application for entering (a) individual productivity, (b) team productivity, and (c) activities done in the last hour

Daily surveys were filled out via an online portal at the end of each workday. These surveys asked participants about the individual and team productivity they had reported throughout the day

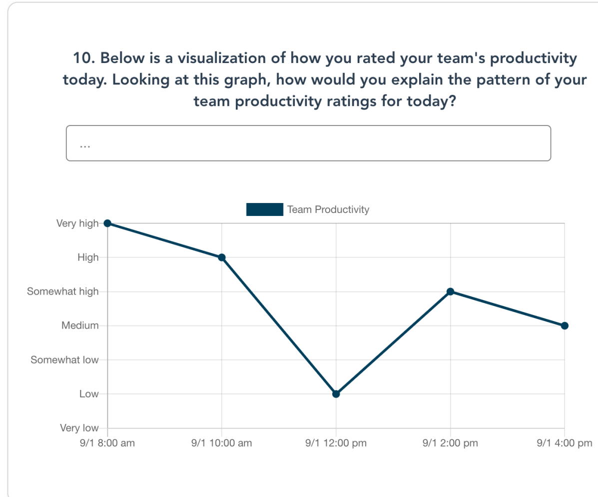


Figure 3: Team Productivity Graph in Daily Survey

(using a visualization generated from their bi-hourly self-reports) and to reflect on any patterns or differences. Further questions centered around several teamwork-related factors, such as how many team interactions they had experienced that day and how related these interactions were to work; time spent in meetings and time spent in unplanned work; how negatively they felt about interruptions that day; how collaborative their tasks were, and whether anything else had influenced their work (open question). Table 1 provides an overview of the questions, and Figure 3 shows an example of the graphs the survey presented for reflection.

Audio-recorded **Interviews** were conducted monthly (at the start, middle, and end of the study). These were primarily used to check in with participants and monitor for any major changes in their work or team. Additionally, we used them to question participants about their perceptions of productivity and their team, including how they assess productivity, which factors influence their productivity, and how they define their team. An overview of the questions asked in these interviews can be found in Table 1.

Follow-up surveys were employed via an online form to ask participants several additional questions, including demographics and the criteria by which they believe their supervisor assesses them. The full survey is included in the replication package [50].

3.3 Quantitative Analysis

We performed statistical analysis to investigate correlations between different factors (presented in Table 3), visualize ratings over time (Figure 4), and generate a regression model of factors affecting team productivity (Table 4). The data source included Likert-style ratings from both the daily surveys and bi-hourly self-reports. We included in the analysis participants who submitted at least two bi-hourly self-reports and one end-of-day survey per day for a minimum of two weeks (i.e., ten daily surveys and 20 bi-hourly self-reports in total), as we considered two weeks a minimum baseline for balancing out possible day-to-day variation. This resulted in 24 of 25 participants being included in the analysis. All values were normalized per participant using the standardization protocol: subtracting mean and dividing by the standard deviation, where the mean and deviation were calculated per participant. This normalization step eases comparison between participants. Quantitative analyses were conducted with R (version 4.0.5, https://www.R-project.org/). Linear regression models were created using the `lme4` package [4]. Tables were created in part using the `apaTables` package [51]. To facilitate transparency, we have posted the survey materials, deidentified quantitative data, and scripts for data analysis here [50].

Means, standard deviations, and correlations with confidence intervals are shown in Table 3.

Figure 4 shows the weekly average individual productivity (dotted) and team productivity (solid) for Teams 3 and 5. It can be seen that both lines are highly individual for each team.

Quantitative scores were further used to give context to qualitative (open-text) answers from the same daily survey response. For example, when a participant talked about "meetings" when responding to the factors that had an effect on their productivity, we referenced the rating they gave for individual productivity and whether it was lower, higher, or equal to the participant's average to understand the sentiment behind the response.

3.4 Qualitative Analysis

Qualitative analysis focused on open-ended questions and was performed on the following data:

Interview transcriptions: questions on team definition (informing results in 4.1) and assessing productivity (4.2 and 4.3).

Follow-up survey: question on assessing productivity (4.3).

Daily surveys: questions on team-related activities, nature of unplanned work, and factors affecting productivity (4.4).

For all three, we employed the reflexive thematic analysis introduced by Braun and Clarke [10] from a critical-realist perspective. The 6-step process included gaining familiarity with the data, inductively generating codes, searching for themes/clusters of codes, reviewing these themes, defining and naming themes, and producing a final summary. One author led the process, with two other authors involved in the initial coding and the final theme clustering. For both the interviews and free-form survey answers, we began by coding half of the sample set and continuing until no novel codes were identified, at which point we considered convergence and saturation to be reached. In the end, we reached saturation after fully coding the interviews of 14 participants (56%) and the daily surveys of 20 participants (80%). Below, we discuss the most frequently occurring themes. For the qualitative data, we use the × sign to indicate how many developers mentioned the theme.

4 RESULTS

In the following, we present the key themes of our analysis: (1) the fluid notion of teams, (2) the ways and difficulties of gauging team productivity by individual developers, (3) the discrepancy between managers' and developers' views on an individual's productivity within a team, and (4) the interplay between team-related factors and productivity perceptions. We focus primarily on the team aspect of software development, though the collected data also corroborates many earlier findings on individual productivity, such as the positive effects of having clear goals, or the negative effects of frequent context switches.

4.1 The Fluid Notion of Teams

In the three monthly interviews, we asked participants how they define their team and whether this team has changed since the start of the study. We found that the concept of team is more fluid than we initially expected. In the literature, a team in software development is often treated as a well-defined and fixed concept. For instance, in Scrum–a common agile process model–a team is defined as "small" (7±2 people) with "no sub-teams" and people working "full time in the team".[1,2] However, in practice, the concept of a team may be fluid, making the assessment of team productivity more difficult.

All participants in our study stated that they are part of a larger team with up to 15 people. The participants described that these larger teams sometimes change due to reorganizations; over the course of our study, fluctuations in the makeup of these larger teams happened in several cases. Participants also explained that there are ***teams or sub-teams within these larger teams*** (6×, 43%). These sub-teams were typically defined in terms of the people which participants interact with on a regular basis, whom they have the most knowledge about, and with whom they share tasks and goals.

We do have our daily meetings with my smaller team where we talk about what we're working on. [...] And then every week we have a meeting that the entire team [...] and there we find out what the other half of the team is working on (S1)

Yeah, that's a little interesting. Because when I would talk about my team, I would always be talking about my entire team, like the complete team. But in the short term, like, while I'm on a particular [..] smaller team, [..] our goals would be aligned and shared. So we'd be doing each other's code reviews, we'd be doing verification for our tasks and all that. So yeah, I think if you talk about a team, in the sense of like, having shared goals and working together towards them, that would be like this smaller team that I'm a part of (S1)

Sub-teams are ***dynamic and can change frequently*** (6×, 43%), often due to the tasks, projects, or the roles a developer takes on, and can be quite ***fuzzy*** in terms of who is on the team or not (3×, 21%).

I moved from like one small team to a different small team. But that's expected because our smaller teams are supposed to be very flexible (S1)

We're split into like subgroups or squads, or whatever you want to call it. So our squad is actually particularly small. So I have one other person in my squad. [..] The actual team like the full team is nine engineers; actually it's 13 all together, but when I think of my team, I think of it more along lines of the engineers. So that instead of that 13 I think of the nine engineers that we have, those would be the people I talk to the most and share the common goals with (S7)

At the same time there is often ***no single one definition*** of who is part of a team, with different people defining teams differently. Leads of (sub-)teams, for instance, have in contrast to the common fuzzy notion a very clear picture of their team, defining them as the people that report to them (3×, 15%).

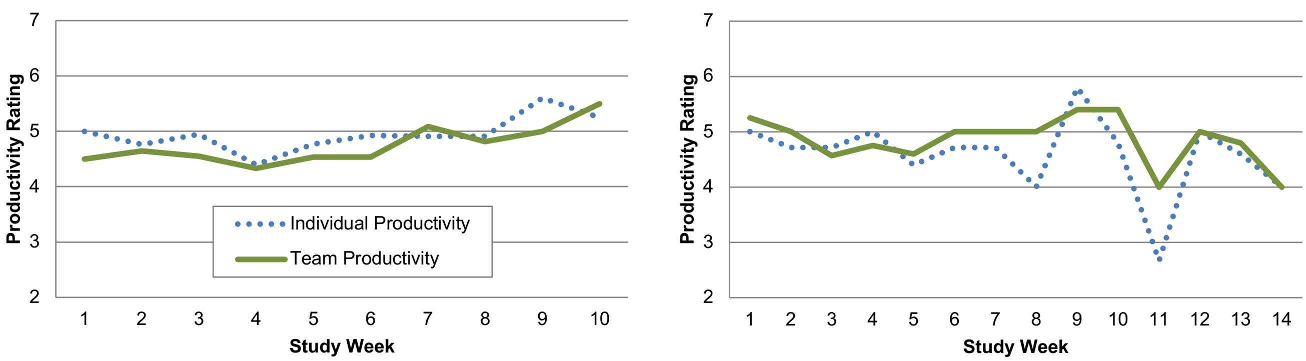
People who report up to me [..] that's who I think of as my team (S12)

[1] www.scrum-institute.org/Scrum_Roles_The_Scrum_Team.php
[2] www.scrum.org/resources/what-is-scrum

Table 3: Correlations

Variable	M	SD	1	2	3	4	5	6
1. Team Productivity	0	1						
2. Individual Productivity	0	1	0.41**					
3. Interruptions	0	1	0.28**	0.35**				
4. Team Interaction	0	1	0.13**	0.10*	0.06			
5. Unplanned Work	0	1	−0.09 *	−0.15 **	−0.24 **	0.11**		
6. Meeting Time	0	1	0.02	0.06	0.02	0.42**	0.10*	
7. Task Collaborativeness	0	1	0.05	0.07	0.08	0.53**	0.08	0.37**

M and SD are used to represent mean and standard deviation, respectively. Values in square brackets indicate the 95% confidence interval for each correlation. The confidence interval is a plausible range of population correlations that could have caused the sample correlation (Cumming, 2014). * indicates p < .05. ** indicates p < .01.



(a) Team 3 (7 participants, 10 weeks)

(b) Team 5 (2 participants, 14 Weeks)

Figure 4: Averaged Weekly Individual and Team Productivity Ratings

Finally, developers can be ***part of multiple teams*** and/or interact frequently with outside teams (7×, 35%), often more than with some members of their own (bigger) team (6×, 30%).

I myself work on this project as project lead and partially like, as a developer. I also work in the web based department as project lead and developer (S23)

[..] one of the UX people [..], our product marketing person and technical writer [..] who I don't technically consider on our team, but I do work with them a decent amount on various projects (S10)

This fuzzy and fluid notion of teams, combined with a lack of interaction and knowledge of the people on the bigger teams, makes the assessment of *team* productivity difficult. It also raises questions of whether there can be a single, or a few, definitions of what a team is and how to determine what makes up a good team. For the self-reporting in our study, we asked participants to focus on the "people that you interact with on a regular basis, and with which you share responsibilities for outcomes"; all participants accepted this as an appropriate definition, even if that meant the definition of a team was fluid.

4.2 Gauging Team Productivity by Individuals

How do developers gauge the productivity of their team? Various themes were identified in the final interviews regarding team productivity, including that assessing team productivity is difficult even when activity metrics are available. As a result, developers use several proxies to gauge team productivity.

Assessing team productivity is difficult, especially due to limited team awareness. Many participants explicitly stated that it was difficult for them to assess the productivity of their team, in large part because of a lack of awareness of what their team was working on. Almost half of the coded participants (9×, 45%) mentioned at least once that they have no awareness about their team, regardless of them being an individual contributor or a team manager.

I just have no means or mechanism to easily determine how much work they've done (S25)

I don't have insight into my team's daily productivity (S12)

The prevalence of remote work contributed strongly to this lack of awareness and thus the difficulty in gauging team productivity. From the 20 developers coded, only one participant worked mostly in the office, four (20%) experienced remote and office work during the study period, and the other 15 (75%) worked remotely most of the time. All survey responses for which participants reported having 'no awareness' of their team stem from days on which the participants worked remotely.

I think it's a bit harder to see, like, the productivity when, like, we're doing everything remotely (S5)

Since I can't see my team, I can only guess their productivity (S10)

Engagement with team is often used to gauge team productivity. The importance of team awareness in gauging team productivity further manifests in participants often referring to their team interactions when explaining their team's productivity in the daily survey. More than half of the coded participants (11×, 55%) explicitly mentioned the engagement with their team in these answers, talking about the meetings (11×), more general interactions with teammates (7×), or working together (5×).

I was having productive discussions with my co-workers, so I knew my teammates are also being productive (S1)

Team seemed consistently productive - had some meetings with them and worked with them via slack/jira tickets/email on other items (S10)

The engagement theme was also very prominent for participants that spent time in their office. Three of the four participants that spent time collocated with their teammates explicitly used the interaction as an indicator for the team's productivity, even if it was brief and did not necessarily cover the work performed during the day.

As I have seen them working and what they said, it made me the impression they were productive (S21)

Team members walking into my room with questions and issues [...] As far as I saw, they looked quite productive today (S23)

Availability of others used as indicator for productivity. One factor that was mentioned several times as an indicator for the team's productivity is the availability and responsiveness of teammates. The more responsive teammates were, the more productive the team was perceived. For instance, several participants explicitly mentioned the responsiveness of their teammates in chat applications or on code review discussions as an explanation for a high or low productivity rating (4×, 20%).

My team was responsive when I needed them to be (S24)

Everyone I worked with was fairly responsive and we accomplished a lot together (S24)

A few participants also related the responsiveness of teammates not just to their team productivity, but also to their own productivity when the lack of responsiveness might block them (3×, 15%).

[I was] able to communicate to different departments in a timely manner so I was not blocked on my work (S5)

Automated metrics sparsely used. All participating teams used issue trackers, version control systems, and other tools that provide metrics on team performance. Yet only a few participants (4×, 20%) referred to automatically collected metrics, such as issue tracker activity or code review progress, for their team productivity ratings.

I didn't see many tickets moving forward on the kanban board, and didn't get any request for peer review of code either, so it looks like the team did not feel very productive (S1)

On the contrary, some participants referred to a 'feeling' of how productive their team was rather than concrete metrics that they used to gauge team productivity (2×, 10%).

I was not able to have a feeling about the output of the team today (S16)

In the morning I heard that many things were already done. In the afternoon I did not hear that much anymore so it's more a stomach feeling (S23)

One team even explicitly stated at the beginning of the study that the progress tracked in their issue tracker is not being used for assessing their team's productivity due to the variance in task complexity and size.

Individual productivity as a base. To gauge the team's productivity, participants also used the perception of their own productivity as a basis and assumed that the team's productivity was similar (3×, 15%).

I tend to assume my team is as productive as I am (S10)

Based on their perception of their own productivity and how they thought their activities differ from a usual day, participants adjusted the team's productivity. For instance, when a participant was more distracted, had more meetings than usual, or more context switches, they rate their team's productivity higher than their own productivity.

Since I can't see my team, I assume they are being productive. Though I objectively think I got a fair amount done today. When I look back at the day holistically, there were moments when I felt more distracted, and therefore rated my productivity lower [than my team] in that moment (S10)

I was having to context-switch a lot today [more than usual], but I think other team members probably didn't have to context-switch as much (S24)

The close relation between individual and team productivity perception can also be seen in Figure 4, which illustrates the averaged productivity ratings for the team and individual over the course of the study. The small difference in mean team and individual productivity for the 2899 bi-hourly self-reports ($Mean_{Team}$= 4.61 ±0.78; $Mean_{Ind}$=4.47 ±0.70) as well as the 624 daily surveys ($Mean_{Team}$= 4.98 ±0.79); $Mean_{Ind}$=4.96 ±0.66) for the 25 participants further illustrate the close relation.

This close relation is expected, since the more productive each individual developer is, the more productive the team can be. However, participants frequently delineated themselves from their team, referring to the team as 'the others' and excluding themselves. For instance, participant S24, a member of the test team, stated:

The test team was blocked by the test environment being down, but I was not effected and in fact it probably allowed me to be more productive on my individual tasks (S24)

Only in rare instances did participants talk about the team as a 'we'.

We have been able to finish some task successful in the end of day (S16)

This differentiation between team productivity and individual productivity can also be seen in how solving team issues might be considered detrimental to the individual productivity in some cases, even though the individual helps the team.

Mostly productive until I was interrupted to help with a testing issue and then I ran into the problem with a shared team resource that I had to fix (S24)

4.3 Gauging Individual Productivity in Teams

How is individual productivity assessed in a team context? We were curious to know whether developers are assessed in their teams in the way they perceive they are. To explore this question, we posed the following assessment questions:

For individual contributors: (asked in the follow-up survey) *In a few words, what would you say are the criteria your manager uses to assess your productivity?*
For managers: (asked in the interview and the follow-up survey) *What criteria do you use to assess the productivity of your team?*

Expectations from individuals differ from the ones of managers. Participating individual contributors expect managers to predominantly assess them by taking into account a more traditional productivity notion based on input and output metrics such as

- *how long I am spending on my tasks (S24)*
- *units of work completed per unit of time (S7)*
- *projects completed (S25)*

Two participants further expected the manager to actively take into account feedback or reviews from their team members (S2, S9).

Participating managers, on the other hand, often referred to a more holistic notion of productivity when assessing developers, one which explicitly recognizes that automatic metrics such as velocity do not show the whole picture.

- *Metrics such as bug fixes and flow stream help, but they are not the end-all-be-all (S6)*
- *How much toil was put in: the actual and time and effort, rather than just ticking off accomplishments (S11)*

Other managers adopted an even broader notion of productivity focusing on the commitments and impact:

- *Achieving something close to what we committed with (S4)*
- *Accomplishing the right things which will have impact (S10)*

One possible explanation for this discrepancy is an unequal interpretation of the question: perhaps "assessing" and "being assessed" are not quite mirrored. The responses of individual contributors suggest an interpretation of "how does my manager know I am being productive?", with answers mentioning indicators such as peer feedback and velocity metrics. In other words, participants listed the indicators which they believe their managers use to gain awareness of their productivity. The managers, when asked how they assess the productivity of their team, rather answered "what does it mean for my team to be productive?", listing such factors as commitments and accomplishing the right tasks.

4.4 Interplay between Team-Related Factors and Productivity Perceptions

Working together to create a product is a necessity when developing large and complex software. As a result, individuals often have to balance making progress on individual goals and working with the team. We elaborate on these trade-offs by presenting quantitative and qualitative data collected in the 624 daily surveys. First, we report the statistical analysis of Likert-scale ratings collected in the survey. Second, we present several corresponding themes we found in the open-ended qualitative responses of these surveys.

A Model for Team Productivity

To investigate the interplay between team and individual productivity, we collected data in the 624 daily surveys on team-related factors and activities, in addition to the team and individual productivity ratings. In particular, we were interested in team-related activities, such as meetings, interactions, and interruptions, as well as the focus of the task work on team or individual effort, since all of these target the interplay between the team and the individual and they have often been mentioned in the past in context of individual productivity. We also included a variable on time spent on unplanned work, since we hypothesized that such work mostly stems from team issues in the collaborative setting of software development. The answers in the daily survey on the nature of the unplanned work further confirmed our hypothesis. For instance, one participant reported the source of unplanned work as:

- *A co-worker reached out to me about a problem they were facing, and we both debugged it until we found a solution (S1)*

We created a linear mixed regression model with the team productivity rating as the dependent variable, the variables mentioned above and listed in Table 4 as fixed effects factors, and the participants as random effects factor.[3]

Table 4: Factors Affecting Team Productivity

Variable	Factor
Individual Productivity	0.34^{***}
Feeling about Interruptions - *how positive or negative a person felt about interruptions that day*	0.10^{*}
Team Interaction Amount	0.07
Team Interaction Work-Relevance - *were the team interactions more work- or non work-related?*	0.00
Time Spent in Unplanned Work	−0.06
Time Spent in Meetings	−0.02
Collaborativeness of Task - *whether a task required more individual effort or team work*	−0.01

* indicates p < .05; ** indicates p < .01; *** indicates p < .005

We saw that among these factors (see Table 4), the strongest predictor of team productivity was the individual productivity reported that day ($r=0.34^{***}$), with all other factors being significantly less strong as predictors. One possible explanation for the strong predictive capability of individual productivity is that developers identify closely with the team and assume their team's productivity as their own, or vice versa: in essence, they project their own personal productivity onto the team, which we also found in the qualitative data presented in Section 4.2. We conjecture that this is more likely to occur in the absence of team awareness. This correlation between individual and team productivity is important to consider when measuring team productivity for future studies; it may be wise to at the same time capture individual productivity. Though meetings were often mentioned as a factor affecting productivity in the qualitative data (see below), we did not see a significant effect in the model. This may be because meetings both drive progress forward while concurrently fragmenting the workday and hindering focus.

Meetings

A majority of the participants (15×, 75%) stated that meetings have a substantial impact on their individual productivity as well as their team's productivity, and that this impact can be either positive/productive (11×, 55%) or negative/unproductive (13×, 65%).

- *Team meetings are generally productive for us but our meetings are scheduled anywhere from 30m to 2h apart from one another. We are meeting with 90% of the same attendees in every meeting so*

[3]Note that the correlations can be found in Table 3.

there is little reason for the time gaps in between. I was still mildly productive because I was able to fit in non-thought intensive tasks in between (S7)

The start of the day had a few meetings for me and my team, but the rest of the day was free of interruptions. The start of the day felt a little unproductive due to multiple meetings, but later I found my stride and kept working (S1)

These mixed perceptions of the impact of meetings on productivity echo the findings by Meyer et al. [40]. Meetings were generally seen as productive when there are concrete results or actionable decisions, or when they create a sense of accomplishment for the participants. Meetings were generally seen as unproductive if there is no clear meeting agenda, key people are missing, there are too many people, or if they are not needed. Especially when there are too many meetings and/or the meetings are not necessary, the meetings were mentioned to have a negative impact on productivity and can reduce the time to work and the participant's focus (13×, 65%).

Most of my team was in at least one meeting today that they probably didn't really need to be in (S24)

Supporting Others

More than half of the developers (11×, 55%) mentioned supporting or helping other team members when they compared their individual with the team productivity ratings. Overall, helping others lead to an even distribution of being more, equally, or less productive than the team, resulting in no clear or visible trend. Eight participants mentioned that efficient collaboration, e.g. discussing solutions, mentoring, helping others or generally working together had an impact on their own as well as their team productivity.

Even though I was interrupted to teach a teammate how to do something, I know that teaching them how to do the task will be valuable for the whole team in the future, so I feel pretty good about doing it (S24)

Communicating and collaborating with other team members to help them troubleshoot issues, give them information they needed (S25)

Availability of Colleagues

Availability of others is not only used as an indicator for gauging team productivity, but as already mentioned in Section 4.2, can have an impact on the individual as well as the team's productivity. Multiple developers (6×, 30%) explicitly mentioned the availability of their colleagues as a productivity factor. Especially in cases when participants or the team is blocked due to the unresponsiveness of others, it can severely impact both individual and team productivity.

I had one teammate who was late to a meeting when he was the most important person to be in the meeting, and I had another teammate who was unresponsive for 30 minutes when several people were waiting on him to do something for the release deployment (S24)

Negative: Key person not available (S16)

5 DISCUSSION

Insights from this study can help inform approaches to measure productivity as well as how to enhance productivity.

5.1 Measuring Productivity

Gauging Team Productivity

How to measure productivity for individual software developers and software development teams remains a subject of inquiry and debate. Within the last year, some have advocated for multi-dimensional approaches to assessing individual productivity [23], whereas others believe "team productivity can be measured at an organizational scale with simple, holistic observations [...and...] any serious failings in individual productivity can be discovered by means of good organizational habits" [33]. Our results indicate how tricky it is to measure, assess, or gauge team productivity, whether as a projection of individual productivity or through direct measurement approaches. The fluidity of teams that exist in software development today would need to be incorporated into any assessment approaches. It may be that it is more important from an organizational level to consider how software is actually being produced, rather than how the organizational chart indicates the software is being produced. This kind of shift is occurring in software development as organizations begin focusing on the streams of value being produced [11].

Effects of Team Factors

An avenue for future work would be to investigate how team characteristics (e.g. size, workflow) or roles (manager, contributor) affect productivity perceptions. In this work, we chose not to perform a broad analysis of many teams, favoring instead to conduct an in-depth longitudinal case study of a few teams to examine how team and individual productivity relate over time. A future study could engage a wider sample of teams that vary systematically on certain characteristics to investigate these questions.

5.2 Enhancing Productivity

Our results suggest possible changes to tools and processes that may help enhance productivity in software development. These improvements relate either to team awareness or to engagement.

Improving Team Awareness

The results of our study suggest that developers often lack awareness of their team's work and progress, corroborating previous findings [53]. Given that our study was conducted during the unprecedented remote work situation of the COVID-19 pandemic, this lack of awareness may have been more pronounced as team members did not often have the chance to meet in person. Our participants used a number of digital collaboration tools, for example, Microsoft Teams, Slack, and Zoom for communication; Screen for para-programming; and Atlassian Jira and other digital Scrum and Kanban boards for progress tracking. Though the communication tools help to gauge presence, and progress trackers help to provide quantitative metrics, these tools only give a surface-level measure of productivity, and the process of combining these data can be tedious. There is a need for improving awareness by better communication and collaboration tool design. It is, however, impressive that despite this lack of team awareness, coordination seems to be effective enough for complex projects to succeed. It may be interesting to investigate how much team awareness software teams actually require to operate well, as such a threshold could help to optimize for minimal interruptions.

Improving Engagement

Participants in our study recognized the need for engagement with other team members. However, they struggled with how to engage most effectively. Well-run and structured meetings have been shown to support engagement [40], but it remains challenging for meetings to successfully engage people in projects despite advice and protocols being available on how to best structure and run such meetings. Future research should delve into the nature of the different kinds of engagement required within software development teams. Identifying a taxonomy of engagement types would provide a foundation for assessing different approaches to increase engagement. Engagement could be increased through adopting processes such as meeting protocols, using an appropriate technological solution, or improving the structure of communication, for example discussions based on issues inside the issue tracker, compared to chat channels.

6 THREATS TO VALIDITY

External Validity

As our sample set featured only five teams, generalizability to other development companies and developers might be limited. These companies were recruited through word-of-mouth, social media, and other personal and professional contacts, and may therefore exhibit a sampling bias. We tried to mitigate this bias by recruiting teams of four companies with different product focuses, in North America and Europe. Furthermore our sample set does not exhibit an equal gender balance. In recruiting participants, our primary aim was to include full teams (as many members from a team as possible), and we therefore did not optimize for demographic representation. Our gender distribution was 21 self-reporting males and 4 self-reporting females (16%). This percentage of females comes close to the 20% participation of females reported in software development in the USA [19]. It should also be noted that since the study took place during the COVID-19 pandemic, with most (21 of 25) participants working remotely, the survey responses may not reflect normalcy and should be interpreted with this in mind.

Construct Validity

We performed a Thematic Analysis to analyze the participant responses to the daily surveys and interviews. One potential threat could be that the open coding step for the majority of interviews was performed by one author only. To reduce bias, three authors worked together to create an initial code set and identify and discuss possible themes.

Internal Validity

The daily question about patterns in team and individual productivity were accompanied by a graph displaying the participant's bi-hourly self-reports for the day, to help with reflection. However, five participants reported issues with the graph: four saw issues intermittently and one regularly. This may have affected the accuracy with which participants recalled their productivity for the day.

One question that arises is whether measures of perceived team productivity have value, considering that participants have limited awareness of their team. In this work, we did not aim to capture objective team productivity, but rather to understand how developers perceive their team's productivity even in the absence of concrete knowledge. At times, some participants did not answer the team productivity question or might have also guessed. However, the thoughtful answers to the open-ended daily questions on the productivity patterns and differences (Table 1), which often referred to teammates' activities and meetings, give us cause to believe that the received responses reflect the developers' perception of team productivity given the information available to them.

7 CONCLUSION

This work aims to bridge the gap between individual and team productivity in software development through a longitudinal user study that explores individual perceptions of productivity in a team context. Among other questions, we investigated developers' awareness of their team, definitions and measures of productivity from individual and managerial perspectives, and factors predicting perceived team productivity. We present a methodology for undertaking a longitudinal study of software teams, findings about developer perceptions of their team's productivity, and a replication package with the study material and anonymized quantitative data. We saw several themes emerge from the 624 daily survey responses collected from 25 developers in 5 teams. The findings suggest that developers operate in a fluid team construct with fuzzy, frequently changing boundaries. Furthermore, developers often lack awareness of their team's productivity. Finally, the most significant predictor of perceived team productivity was the reported individual productivity, above factors such as team interaction, unplanned work and time spent in meetings. We suggest that future research should further explore the intricacies of development in a broader spectrum of collaboration (e.g., as an individual, between collaborators, in teams, and on the organizational level) and to consider what the fluidity and dynamism of these collaborative structures mean for the success of software projects.

ACKNOWLEDGMENTS

The authors would like to thank all study participants and companies involved, and Joanna Triscott for support with visualizations.

REFERENCES

[1] Allan J. Albrecht. 1979. Measuring Application Development Productivity. In *Proceedings of IBM Applications Development Symposium*. Monterey, 83.

[2] Teresa M. Amabile and Steven J. Kramer. 2011. *The Progress Principle: Using Small Wins to Ignite Joy, Engagement, and Creativity at Work.* Harvard Business Press.

[3] B. Bailey, J. Konstan, and J. Carlis. 2001. The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. In *INTERACT*.

[4] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* 67, 1 (2015), 1–48. https://doi.org/10.18637/jss.v067.i01

[5] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson, and Helen Sharp. 2008. Motivation in Software Engineering: A systematic literature review. *Information and Software Technology* 50 (08 2008), 860–878. https://doi.org/10.1016/j.infsof.2007.09.004

[6] Salah Bendifallah and Walt Scacchi. 1989. Work Structures and Shifts: An Empirical Analysis of Software Specification Teamwork. In *Proceedings of the 11th International Conference on Software Engineering* (Pittsburgh, Pennsylvania, USA) *(ICSE '89)*. Association for Computing Machinery, New York, NY, USA, 260–270. https://doi.org/10.1145/74587.74624

[7] Carla I. M. Bezerra, José Cezar de Souza Filho, Emanuel F. Coutinho, Alice Gama, Ana Lívia Ferreira, Gabriel Leitão de Andrade, and Carlos Eduardo Feitosa. 2020. How Human and Organizational Factors Influence Software Teams Productivity in COVID-19 Pandemic: A Brazilian Survey. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 606–615. https://doi.org/10.1145/3422392.3422417

[8] B. Boehm. 1987. Improving Software Productivity. *Computer* 20 (1987), 43–57.

[9] Barry W. Boehm. 1987. Improving Software Productivity. *Computer* 20, 9 (Sept. 1987), 43–57. https://doi.org/10.1109/MC.1987.1663694

[10] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative research in psychology* 3 (Jan 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[11] Flint Brenton. 2019. What Is Value Stream Management? A Primer For Enterprise Leadership. *https://www.forbes.com/sites/forbestechcouncil/2019/07/08/what-is-value-stream-management-a-primer-for-enterprise-leadership/?sh=3fbc9cc17b67* (July 2019).

[12] Edna Dias Canedo and Giovanni Almeida Santos. 2019. Factors Affecting Software Development Productivity: An Empirical Study. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (Salvador, Brazil) *(SBES 2019)*. Association for Computing Machinery, New York, NY, USA, 307–316. https://doi.org/10.1145/3350768.3352491

[13] Wayne Cascio. 2000. Managing A Virtual Workplace. *Academy of Management Perspectives* 14 (08 2000). https://doi.org/10.5465/AME.2000.4468068

[14] Marcelo Cataldo, James D. Herbsleb, and Kathleen M. Carley. 2008. Socio-Technical Congruence: A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (Kaiserslautern, Germany) *(ESEM '08)*. Association for Computing Machinery, New York, NY, USA, 2–11. https://doi.org/10.1145/1414004.1414008

[15] Alec Chrystal and Paul Mizen. 2003. Goodhart's Law: Its Origins, Meaning and Implications for Monetary Policy. *Cent Bank Monet Theory Pract Essays Honour Charles Goodhart* 1 (01 2003). https://doi.org/10.4337/9781781950777.00022

[16] Victor Clincy. 2003. Software Development Productivity and Cycle Time Reduction. *Journal of Computing Sciences in Colleges* 19 (12 2003), 278–287.

[17] Broderick Crawford, Ricardo Soto, Claudio León de la Barra, Kathleen Crawford, and Eduardo Olguín. 2014. The Influence of Emotions on Productivity in Software Engineering. In *HCI International 2014 - Posters' Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 307–310.

[18] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 175–182. https://doi.org/10.1145/985692.985715

[19] datausa.io. [n.d.]. Software developers, applications and systems software. https://datausa.io/profile/soc/software-developers-applications-systems-software

[20] Thomas H. Davenport and Keri E. Pearlson. 1998. Two Cheers for the Virtual Office. *Sloan Management Review* 39 (1998), 51–65.

[21] Claudia De O. Melo, Daniela S. Cruzes, Fabio Kon, and Reidar Conradi. 2013. Interpretative Case Studies on Agile Team Productivity and Management. *Inf. Softw. Technol.* 55, 2 (Feb. 2013), 412–427. https://doi.org/10.1016/j.infsof.2012.09.004

[22] Prem Devanbu, Sakke Karstu, Walcélio Melo, and William Thomas. 1996. Analytical and Empirical Evaluation of Software Reuse Metrics. In *Proceedings of the 18th International Conference on Software Engineering* (Berlin, Germany) *(ICSE '96)*. IEEE Computer Society, USA, 189–199.

[23] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's More to It than You Think. *Queue* 19, 1 (Feb. 2021), 20–48. https://doi.org/10.1145/3454122.3454124

[24] C. A. E. Goodhart. 1984. *Monetary Theory and Practice: The UK Experience*. Macmillan Press.

[25] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2014. Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. *Journal of Software: Evolution and Process* 27 (08 2014). https://doi.org/10.1002/smr.1673

[26] Jeffrey Hyman, Chris Baldry, Dora Scholarios, and Dirk Bunzel. 2003. Work-Life Imbalance in Call Centres and Software Development. *British Journal of Industrial Relations* 41 (02 2003), 215–239. https://doi.org/10.1111/1467-8543.00270

[27] C. Jones. 1994. Software metrics: good, bad and missing. *Computer* 27, 9 (1994), 98–100. https://doi.org/10.1109/2.312055

[28] I. Khan, W. Brinkman, and R. Hierons. 2010. Do moods affect programmers' debug performance? *Cognition, Technology & Work* 13 (2010), 245–258.

[29] Young-Ho Kim, Eun Kyoung Choe, Bongshin Lee, and Jinwook Seo. 2019. Understanding Personal Productivity: How Knowledge Workers Define, Evaluate, and Reflect on Their Productivity *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300845

[30] Amy Ko. 2019. *Individual, Team, Organization, and Market: Four Lenses of Productivity*. Apress, 49–55. https://doi.org/10.1007/978-1-4842-4221-6_6

[31] Amy Ko. 2019. *Why We Should Not Measure Productivity*. Apress, 21–26. https://doi.org/10.1007/978-1-4842-4221-6_3

[32] B. Lakhanpal. 1993. Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis. *Information and Software Technology* 35, 8 (1993), 468–473. https://doi.org/10.1016/0950-5849(93)90044-4

[33] Isaac Lyman. 2020. Can developer productivity be measured? *https://stackoverflow.blog/2020/12/07/measuring-developer-productivity/* (Dec. 2020).

[34] Gloria Mark, Daniela Gudith, and Ulrich Klocke. 2008. The cost of interrupted work: More speed and stress. *Conference on Human Factors in Computing Systems - Proceedings*, 107–110. https://doi.org/10.1145/1357054.1357072

[35] Gloria Mark, Shamsi T. Iqbal, Mary Czerwinski, Paul Johns, and Akane Sano. 2016. *Neurotics Can't Focus: An in Situ Study of Online Multitasking in the Workplace*. Association for Computing Machinery, New York, NY, USA, 1739–1744. https://doi.org/10.1145/2858036.2858202

[36] Katrina D. Maxwell and Pekka Forselius. 2000. Benchmarking Software-Development Productivity. *IEEE Softw.* 17, 1 (Jan. 2000), 80–88. https://doi.org/10.1109/52.820015

[37] Claire R. McInerney. 1999. Working in the virtual office: Providing information and knowledge to remote workers. *Library & Information Science Research* 21, 1 (1999), 69–89. https://doi.org/10.1016/S0740-8188(99)80006-1

[38] André Meyer, Earl T. Barr, Christian Bird, and Thomas Zimmerman. 2019. Today was a Good Day: The Daily Life of Software Developers. In *Proceedings of the IEEE Transactions on Software Engineering (2019)*.

[39] André Meyer, Laura Barton, Gail Murphy, Thomas Zimmermann, and Thomas Fritz. 2017. The Work Life of Developers: Activities, Switches and Perceived Productivity. *IEEE Transactions on Software Engineering* PP (01 2017), 1–1. https://doi.org/10.1109/TSE.2017.2656886

[40] André Meyer, Thomas Fritz, Gail Murphy, and Thomas Zimmermann. 2014. Software Developers' Perceptions of Productivity. (11 2014). https://doi.org/10.5167/uzh-98324

[41] André N. Meyer, Gail C. Murphy, Thomas Fritz, and Thomas Zimmermann. 2019. *Developers' Diverging Perceptions of Productivity*. Apress, 137–146. https://doi.org/10.1007/978-1-4842-4221-6_12

[42] Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. "How Was Your Weekend?" Software Development Teams Working From Home During COVID-19. (01 2021).

[43] Audris Mockus, Roy T. Fielding, and James D. Herbsleb. 2002. Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Trans. Softw. Eng. Methodol.* 11, 3 (July 2002), 309–346. https://doi.org/10.1145/567793.567795

[44] S. C. Müller and T. Fritz. 2015. Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. 688–699. https://doi.org/10.1109/ICSE.2015.334

[45] Edson Oliveira, Tayana Conte, Marco Cristo, and Emilia Mendes. 2016. Software Project Managers' Perceptions of Productivity Factors: Findings from a Qualitative Study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Ciudad Real, Spain) *(ESEM '16)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. https://doi.org/10.1145/2961111.2962626

[46] Chris Parnin and Spencer Rugaber. 2011. Resumption Strategies for Interrupted Programming Tasks. *Software Quality Control* 19 (03 2011), 5–34. https://doi.org/10.1109/ICPC.2009.5090030

[47] Robert D. Pritchard. 1995. Productivity measurement and improvement: Organizational case studies. Greenwood Publishing Group.

[48] Paul Ralph, Sebastian Baltes, Gianisa Adisaputri, Richard Torkar, Vladimir Kovalenko, Marcos Kalinowski, Nicole Novielli, Shin Yoo, Xavier Devroey, Xin Tan, Minghui Zhou, Burak Turhan, Rashina Hoda, Hideaki Hata, Gregorio Robles, Amin Milani Fard, and Rana Alkadhi. 2020. Pandemic programming: How COVID-19 affects software developers and how their organizations can help. *Empirical software engineering* 25 (11 2020), 1–35. https://doi.org/10.1007/s10664-020-09875-y

[49] Helen Russell, Philip O'Connell, and Frances Mcginnity. 2007. The Impact of Flexible Working Arrangements on Work-Life Conflict and Work Pressure in Ireland. *Gender, Work & Organization* 16 (05 2007). https://doi.org/10.1111/j.1468-0432.2008.00431.x

[50] Anastasia Ruvimova, Alexander Lill, Jan Gugler, Lauren Howe, Elaine Huang, Gail Murphy, and Thomas Fritz. 2022. Study Materials. In *Open Science Framework*. https://osf.io/28gbx/?view_only=b65543a1d53b457aa09e708aa210fe72

[51] David J. Stanley and Jeffrey R. Spence. 2018. Reproducible Tables in Psychology Using the apaTables Package. *Advances in Methods and Practices in Psychological Science* 1, 3 (2018), 415–431. https://doi.org/10.1177/2515245918773743 arXiv:https://doi.org/10.1177/2515245918773743

[52] Christoph Treude, Fernando Figueira Filho, and Uirá Kulesza. 2015. Summarizing and Measuring Development Activity. In *Proceedings of the 2015 10th

Joint Meeting on Foundations of Software Engineering (Bergamo, Italy) *(ESEC/FSE 2015)*. Association for Computing Machinery, New York, NY, USA, 625–636. https://doi.org/10.1145/2786805.2786827

[53] Christoph Treude and Margaret-Anne Storey. 2010. Awareness 2.0: Staying Aware of Projects, Developers and Tasks Using Dashboards and Feeds. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1* (Cape Town, South Africa) *(ICSE '10)*. Association for Computing Machinery, New York, NY, USA, 365–374. https://doi.org/10.1145/1806799.1806854

[54] S. Wagner and M. Ruhe. 2018. A Systematic Review of Productivity Factors in Software Development. *ArXiv* abs/1801.06475 (2018).

[55] C. E. Walston and C. P. Felix. 1977. A Method of Programming Measurement and Estimation. *IBM Syst. J.* 16, 1 (March 1977), 54–73. https://doi.org/10.1147/sj.161.0054

[56] Murat Yilmaz, Rory O'Connor, and Paul Clarke. 2016. Effective Social Productivity Measurements During Software Development: An Empirical Study. *International Journal of Software Engineering and Knowledge Engineering* 26 (04 2016). https://doi.org/10.1142/S0218194016500194