



Co-Creation in Fully Remote Software Teams

Victoria Jackson
André van der Hoek

vfjackso@uci.edu

andre@ics.uci.edu

University of California, Irvine

Irvine, CA, USA

Rafael Prikladnicki

rafaelp@pucrs.br

Pontificia Universidade do Rio Grande do Sul

Porto Alegre, RS, Brazil

ABSTRACT

In this paper, we use the lens of co-creation—a concept originally coined and applied in the fields of management and design that denotes how groups of people collaboratively create something of meaning through an orchestration of people, activities, and tools—to study how fully remote software teams co-create digital artifacts that can be considered as a form of documentation. We report on the results of a qualitative, interview-based study with 25 software professionals working in remote teams. Our primary findings are the definition of four models of co-creation, examples of sequencing these models into work chains to produce artifacts, factors that influence how developers match tasks to models and chains, and insights into tool support for co-creation. Together, our findings illustrate how co-creation is an intentional activity that has a significant role in how remote software teams' choose to structure their collaborative activities.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in collaborative and social computing**; **Computer supported cooperative work**.

KEYWORDS

Collaboration, Remote software development, Developer tools, Virtual software teams, Software team practices

ACM Reference Format:

Victoria Jackson, André van der Hoek, and Rafael Prikladnicki. 2024. Co-Creation in Fully Remote Software Teams. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3597503.3623297>

1 INTRODUCTION

While some organizations have always worked remotely, the recent COVID-19 pandemic necessitated many more organizations to shift to fully remote work in which each individual contributes from their own location [58]. Today, while several organizations are adopting a hybrid approach, many organizations have committed to retaining fully remote work [51].

A practice that has emerged in the context of fully remote work is to take an artifact-centered approach: some artifact, or some set of closely related artifacts, is the focal point through which the collaboration takes place [45]. A team, for instance, may engage in a video call to brainstorm potential new product features in a Google Doc, after which the team lead organizes the ideas in a Google Form and issues a vote to prioritize the features. As another example, a developer may craft a revised UI for some interface to then share the mock-up and invite comments from others through a Slack conversation or by requesting them to leave comments in the online document. After integrating the comments, the developer organizes a video call for a final, synchronous review.

Collaboration practices in distributed software development have been the subject of many studies (see Section 3). However, to date none has examined how teams of software developers orchestrate themselves through these kinds of sequenced activities to develop shared artifacts when working entirely remotely. Developing such an understanding is important, however, to uncover potential best practices, determine if common models exist that are especially applicable to fully remote work, understand the tooling landscape, identify current challenges, and more.

This paper contributes an interview-based qualitative study that examines collaboration practices in fully remote software development work through the lens of co-creation, a concept developed and used in the management (e.g., [35]) and design (e.g., [41]) literature. The term co-creation refers to the orchestration of activities, people, and tools that enables a team to create an artifact collaboratively and have this artifact contribute to the team's collective knowledge and shared understanding [29]. We purposefully choose the lens of co-creation, because it considers the fact that, to produce an artifact representing its shared views, team members often engage not in a singular activity, but in an intentional sequence of closely-related activities that lead to the eventual version of the artifact (as in the two examples above). To date, exactly how teams approach doing so has not been studied.

The specific research question we answer in this paper is: **How do fully remote software teams co-create artifacts?** In answering this question, our main contribution is the identification of four different models of co-creation that vary along the synchronicity of the communication activities (synchronous vs. asynchronous) and the number of authors (one vs. many). Furthermore, we provide examples of how these models are linked together into chains of action and factors that developers consider when matching a task to a model or chain. We also find that a single, integrated toolset seems adequate in supporting all four models.



This work is licensed under a Creative Commons Attribution International 4.0 License. *ICSE '24, April 14–20, 2024, Lisbon, Portugal*

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0217-4/24/04.

<https://doi.org/10.1145/3597503.3623297>

2 SCOPE

Our study focuses on *fully remote teams* (also known as fully virtual teams) and considers such teams as groups of people who work together on a project or product while being dispersed across different geographical locations; no team members are co-located [19]. We have purposefully narrowed our focus to remote teams as we expect many companies to persist with remote working (e.g., GitLab [46]) or adopt hybrid work arrangements involving remote work some of the time [47].

We further scope our study to focus on co-creation in the context of *digital artifacts that can be considered a form of documentation* that guide some next aspect of development. Examples of such documentation includes requirements, designs of new features, and system architectures. We purposely exclude source code, test cases, build scripts, and the like because the practices and tools by which teams of developers work on these are well-established and strongly supported by existing tools (e.g., [48, 18]).

3 RELATED WORK

In this section, we review previous research on artifact-centered collaboration practices, remote work, and co-creation.

Collaboration practices. Many different collaborative practices revolve around artifacts. For example, a recent review identified 83 different design thinking techniques used by software teams, noting thought is required to determine which artifacts to elaborate in framing the problem and determining the solution [33]. As another example, user stories for describing requirements can be created through a series of meetings of a team with its product manager to refine an initial concept into detailed stories [22] [44] or by running user story workshops involving the team and the product owner or customer [13]. As a final example, user interfaces may be agreed upon through a “design studio” session where each participant sketches their user interface (UI) design before sharing and building upon the ideas to agree on a final UI [55].

More generally, software teams have always used diagrams and sketches to discuss aspects of the software under development, such as an architectural change or a new feature [4]. Sketching creates a common understanding between colleagues [1], while the act of collaboratively sketching itself supports reasoning activities such as consideration of alternatives [23].

Software teams also make use of collaborative writing for lengthier documents [56]. Five collaborative writing strategies have been identified such as a single writer, a scribe, and separate writers [34]. More recently, with the widespread adoption of tools such as Google Docs with real-time editing capabilities, six collaborative writing work patterns were identified that involved multiple people adding content through a combination of synchronous and asynchronous methods [31]. Synchronous use of Google Docs also encourages teams to use the tool for online ideation activities [17]. In a study of a real-time editing collaborative writing tool [6], it was noted that half of the documents have a single author and that whilst collaboration on common parts of the document occurs, it generally happens asynchronously with authors taking turns to edit. Not all tools support multiple people concurrently editing a document so alternatives are to use the open-source approach

of fork-and-pull via a tool such as GitHub [52] or to let anyone contribute content to a wiki when they hold an exclusive lock [54].

Remote work. Distances (geographic, temporal) between distributed developers cause a number of barriers, including collaboration challenges [12] and difficulties establishing common ground [32]. Producing digital artifacts with multiple, distributed editors can help distributed teams coordinate knowledge [10]. Fully remote organizations can adopt one of two distinct organizational designs to aid collaboration: either an asynchronous orientation or a real-time collaboration orientation [37]. Asynchronous organizations are characterized by document-mediated interactions with rich decision trails whereas real-time organizations rely on human-to-human interactions with meeting-centric discussions. Within remote software teams, both asynchronous and synchronous communication modes are used when collaborating ([28, 11]) as the alternative modes aid different aspects of the collaboration. For example, the use of asynchronous chat can resolve issues while a synchronous video call can focus on ambiguities [7]. Teams use a variety of collaborative tools [20] to support coordination, artifact management and task management [42].

Co-Creation. The concept of co-creation has been adopted in domains such as design [41] and business management [35] to explain how groups of different people with varying perspectives, experiences and skills collaboratively create something of meaning. At its most basic, co-creation refers to any act of collective creativity shared by at least two people [41]. More recent definitions consider co-creation through engagements with “interactive platforms”. Such “platforms” consist of relationships of artifacts, processes, interfaces, and persons, and are supported by digital technology [35]. In considering the meanings of co-creation across varied domains [14], scholars have argued open-source development with its bazaar model [36] is also a form of co-creation. Co-creation can enhance the effectiveness of collaboration and positive group outcomes [16]. Moreover, bringing together individuals to co-create helps meet their needs for socialization and meaning making [14].

Aligned with [24], we consider co-creation as a “form of collaboration.” Based on other domains [41, 35], we define co-creation as the *orchestration of activities, people, and tools to create an artifact*. By using orchestration as the focus of our examination of how remote software teams engage, our research differs from previous research that has considered singular aspects of this orchestration in detail (e.g., collaborative writing [31]) or focused on a specific collaboration practice (e.g., “design studio” [55]). As a result, our study is able to identify several canonical co-creation models that document higher-level collaborative practices as carefully ordered sequences of activities, and thereby provide active guidance to organizations and remote software teams as to how to best structure their artifact-centered collaborative work.

4 STUDY DESIGN

We decided on semi-structured interviews as our research method, since their open-ended nature provides flexibility to explore topics of interest [43]. As organizational design may impact the collaboration approach [37], we decided to interview people from different companies in varying industries to provide a breadth of experiences.

4.1 Interviews

Two rounds of interviews were conducted. After the first round, we analyzed the data, noticed the emergence of co-creation models, and identified areas we wished to explore in greater detail. We decided to discuss our findings and explore these ways of co-creation further through follow-up interviews with five of our previous participants.

For the first round, semi-structured interviews were used to gather data on each participant's perspectives on and experiences in collaborating remotely with colleagues. An interview protocol was used to guide the interviews. This protocol contained questions to understand better the working practices (including tools) used to collaborate remotely. Topics such as ideation and undertaking design/architecture work were included. In all interviews, the researchers introduced themselves, the purpose of the research, and gained the participant's informed consent, including permission to record the interview. The same two researchers undertook the interviews with the lead researcher asking the majority of the questions whilst the second researcher listened and asked clarification questions. All interviews were conducted remotely over Zoom between November 2021 and February 2022. Each interview was recorded and auto-transcribed by Zoom. Each transcript was manually corrected by the lead researcher as the Zoom auto-transcribed transcript contained errors. The duration of the interviews ranged from 24 to 53 minutes with an average of 42 minutes.

The second round of semi-structured interviews with five of the original participants (P04, P6, P16, P18, P22) took place remotely over Zoom between February 2023 and March 2023. These participants were selected from different organizations, as well as based on a combination of the diversity of their prior answers, availability, and willingness to attend a second interview. A short interview protocol was prepared to guide these interviews. The interviews were conducted by the same two researchers as earlier. These interviews had an average duration of 30 minutes.

4.2 Ethics and Data Collection

The interview study was approved by the researchers' institution's Human Research Protections program and included gaining verbal informed consent from each study participant. The recorded interviews and transcripts were stored on a secure university drive with access restricted to the researchers. The researchers removed personal identifiable information from the transcripts.

4.3 Participants

We selected participants who worked: (i) within a software team, (ii) in a fully remote setting, (iii) spoke English, and (iv) worked on a commercial software product. We also ensured that the set of participants was diverse in terms of: (i) industry, (ii) the type of company, (iii) size of company, (iv) job role, and (v) number of years experience. Convenience sampling was initially used to identify participants from the researchers' extensive industry networks. Additional participants were subsequently identified through snowballing. Of the 25 participants (Table 1), 19 of the participants are men and six are women. Due to the snowballing, six people were interviewed from one company and five from another. In each case, the participants worked in different teams, had different roles, and

Table 1: Participant Demographics

ID	Role	Exp.(Yrs)	Location	Industry
P01	Staff Eng.	12	USA	Technology
P02	Snr. Soft. Eng.	12	USA	Leisure
P03	Strat. Part. Dev. Mgr.	21	USA	Technology
P04	Product Mgr.	14	Arg.	Technology
P05	Architect	15	USA	Technology
P06	Mgr. of Tech.	20	USA	Games
P07	CEO	12	USA	EduTech
P08	Software Eng.	8	USA	Technology
P09	Eng. Mgr.	8	USA	Aeronautical
P10	Pgm. Mgr.	15	USA	Food
P11	Head of Agile	22	USA	Financial
P12	Scrum Master	7	UK	Financial
P13	Head of Eng.	28	UK	Financial
P14	Engineer	3	Can.	Technology
P15	Architect	11	USA	Technology
P16	Team Lead	6	USA	Healthcare
P17	CTO	12	USA	Manufacturing
P18	Soft. Eng.	4	USA	Technology
P19	Snr. VP of Plat. Prod.	20	USA	Sports
P20	Tech. Prj. Mgr.	10	USA	Technology
P21	Agile Lead	30	USA	Technology
P22	Hd. of Sys. Arch.	13	UK	Financial
P23	Eng. Pgm.Mgr.	15	USA	Technology
P24	Hd. Agile Dev.	20	UK	Financial
P25	CTO	18	USA	Games

were interviewed separately. All participants voluntarily joined the interviews; there was no financial or other incentive.

4.4 Data Analysis

The interview transcripts were used as the data source for the analysis. Inductive thematic analysis [39] was used to identify the themes emerging from the interviews. This approach meant the data analysis commenced with an empty codebook that was refined as the coding progressed. The same two researchers who conducted the interviews coded the transcripts using MaxQDA [25].

When coding the first round of interviews, the two researchers coded the first two transcripts together to initiate the codebook. Subsequently, the lead researcher coded the remaining transcripts in batches, coding two additional transcripts, then proceeding in groups of four until all transcripts were coded. After each batch, and prior to coding the next batch, the second researcher reviewed the coding privately before meeting with the lead researcher to discuss any disagreements on the coding until consensus between the two was reached. Close attention was paid to any codes emerging in the most recent batch of coded transcripts to ensure both researchers agreed on the necessity of the new codes. As the codebook evolved, prior transcripts were re-coded to use the new codes with a further round of review with the second researcher to discuss and resolve disagreements. At the regular discussions between the two researchers, several themes were identified by grouping related codes and sub-codes and analyzing the usage of codes in the transcripts. After ten transcripts had been coded, the theme

of “four co-creation models” started to emerge as the researchers noted there was commonality in the way teams used people, activities, temporality, and tools to co-create artifacts. An example of another theme to emerge was “one integrated toolset”. No new codes/sub-codes or themes were identified in the coding of the final five transcripts. The researchers, including one who had not been involved in the coding, discussed this in detail and felt that, whilst there were no new themes emerging, it would be beneficial to validate and gather additional data on two subthemes present in the data relevant to the research question. Specifically, the subthemes to explore further were: (i) factors that determine the model to be used, and (ii) chaining of the models. These two subthemes had not been explored in depth in the initial interviews.

For the second round of interviews, we followed a similar data analysis approach with the lead researcher coding each transcript, the second researcher privately reviewing the coding, and discussions to resolve any disagreements. New codes related to factors (e.g., “culture”, “experience”) and chaining emerged in this second round of data analysis and were added to the initial codebook.

4.5 Supplementary Material

The interview protocols and the codebook are available [15]. The transcripts are unavailable as these contain sensitive participant data, which the participants agreed to share only with the paper’s researchers.

5 FINDINGS

In this section, we answer our research question: How do fully remote software teams co-create artifacts?

When asked about their practices for collaboratively creating artifacts, our participants readily articulated their perspectives about combining authorship, temporality, and tools when working remotely on various artifact-centered tasks associated with software development. We find that their intuitive sense of best practices falls into four main categories of models that are matched with particular tasks and linked together into chains of action. Below, we discuss the four models, provide examples of work chains, introduce the factors that influence how developers match tasks to models, and describe the role of tools.

5.1 Four Models

The combination of asynchronous and synchronous modes of communication and the number of authors led us to identify **four elementary models used to co-create artifacts**. These models varied by the primary communication mode involved in co-creating the artifact (asynchronous vs. synchronous) and the number of authors (single vs. many) and are shown in Fig. 1. By authors, we mean the people who are the main writers of the final contents. This can either be a sole author who actively incorporates the ideas and views from other participants or multiple authors who all contribute content to the document.

The activities and participants of each of these models are illustrated in Fig. 2 and detailed below. Each model shows the main activities involved in co-creating the artifact, the sequencing of the activities, the participants (including the author(s)), and the

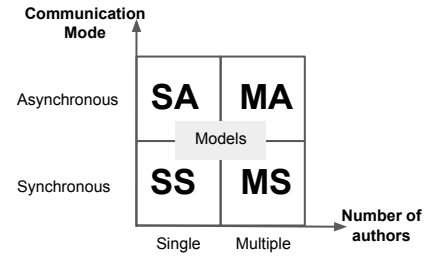


Figure 1: The Dimensions of Co-creation

synchronicity of each activity.

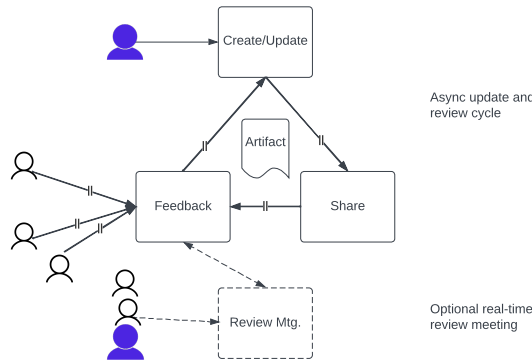
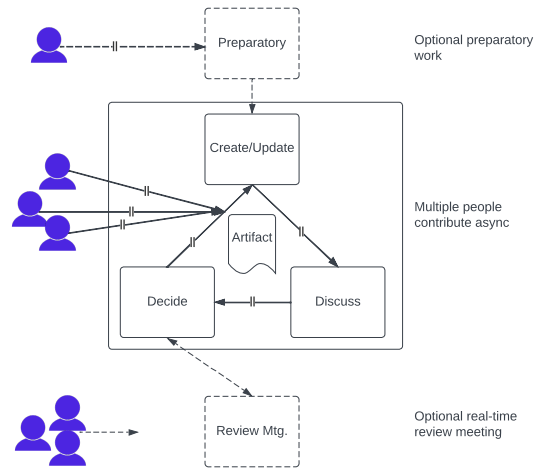
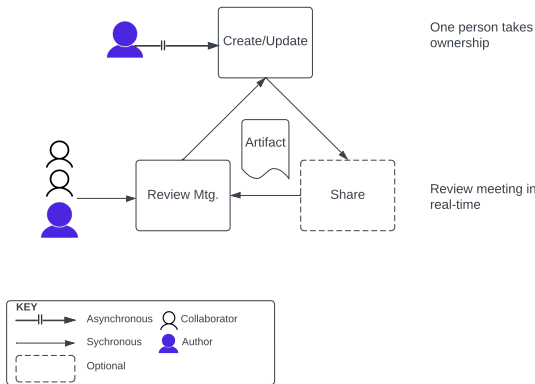
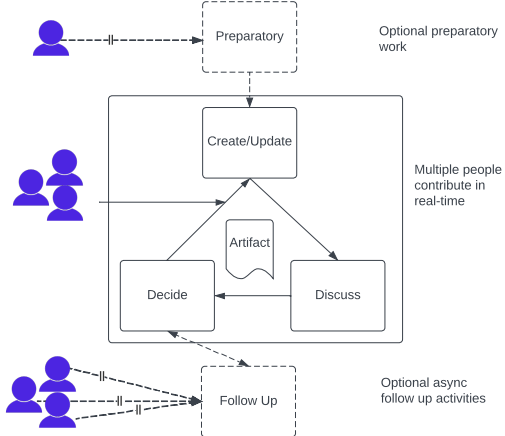
SA - Single Author & Asynchronous Feedback

A single person takes responsibility for creating and editing the artifact and seeks feedback from team members asynchronously. This cycle continues until the team is ready to move on with the artifact. Our participants gave examples of using this model for creating design documents or detailing a feature. Feedback from others is received asynchronously, “*But with remote designing, I found it to be better because you need to write down every single aspect of the design and people would comment on it, and we would have discussions on that Google Doc.*” (P18). While these discussions can take place in e-mail or a tool such as Slack, many participants preferred receiving it directly within the artifact, “*We are trying to get teams to rather than do like email threads...if you’re thinking about this feature create your boxes and arrows diagram in LucidChart, share that out.*” (P19)

Our participants noted that it can take longer to complete collaborative tasks when working asynchronously in a remote team. One approach to overcome this is to ensure all participants are aware of deadlines “*I think that the key aspect there is setting the expectations. Hey? We need this design, approved by X date, right? And then we’ll write it for I don’t know. 3, 5 days, and then we have another 3, 5 days for approval.*” (P6)

In this model, the aim is to complete the task asynchronously. However, in both asynchronous models, we observed that sometimes disagreements or issues arose that could not be resolved efficiently leading to a final synchronous meeting to conclude the overall collaboration, “*Sometimes you can keep arguing on a Google Doc, and not really get what someone is saying because it’s a small chat box that you need to put all your thoughts, and sometimes talking makes things easier so whenever there’s some contentious point, we prefer doing it in a presentation.*” (P18)

The approach of specifying the content in an artifact and receiving asynchronous feedback has a few benefits: (i) it enables authors to receive feedback without recourse to a meeting, (ii) people who would struggle to attend a meeting can respond at their own time, and (iii) feedback persists so it is discoverable in the future and furthermore enables team members to take other feedback into account in providing their own. As one participant highlighted, “*We really try to push async feedback and async discussion as much as we can. For that textual availability for revisiting the subject, but then also peoples’ calendars are just flooded nowadays.*” (P20)

Single Author & Asynchronous Feedback (SA)**Multiple Authors & Asynchronous Co-creation (MA)****Single Author & Synchronous Feedback (SS)****Multiple Authors & Synchronous Co-creation (MS)****Figure 2: The Four Models of Co-creation****MA - Multiple Authors & Asynchronous Co-creation**

Compared to a single author in Model SA, in Model MA multiple people contribute content to the artifact in their own time and asynchronously. Discussions about the artifact take place via chat or by embedding comments directly in the artifact, “So, my manager or my colleague starts an outline sketch, and then I will go in the same document and then start adding stuff. And there may be items where she has written something that I don’t think, I have a comment about or something, that I add a comment there. And then, usually you know they respond to it, they resolve it. They take action or they say why they don’t want to do it right, basically it’s like a communication but in an offline mode on a shared document that’s very common.” (P03). Examples of such artifacts produced this way included product vision and specification documents.

Optionally, teams may hold a synchronous pre-meeting beforehand to agree how to divide the work “So they do a synchronous meeting to coordinate. Who is going to write what part of the tech

spec? And then they fall back into asynchronous work.” (P18) or to agree on the content “It starts with a kickoff conversation with that working group. And then that follows through to a kind of a specification, kind of like a write up that is put together.” (P07)

Similar to model SA, if asynchronous discussions were not concluding satisfactorily, a **synchronous review meeting was used to reach consensus** before finalizing the artifact, “And whatever was not answered, or sorted out with that we have a final review meeting to look at it.” (P06)

Even though multiple authors were contributing to the artifact, in some cases, a single person from the group of authors was held accountable for ensuring the artifact was completed in a timely manner, “So you gotta have somebody driving it. Check closure.” (P6)

Amongst the benefits mentioned by our participants in working asynchronously was the sense of teamness that arises when editing a document by seeing that colleagues have been contributing too, “There are times when my manager and I are doing editing, at the

same time. I can see that she's there right there I can tell who's active and I can see that she is editing a different paragraph, I'm editing a different paragraph, how much more collaborative can it get in an offline world right?" (P3). This is an important consideration given the sense of isolation that some remote workers experience [49].

MS - Multiple Authors & Synchronous Co-creation

Multiple people come together in an online meeting to co-create the artifact. They discuss, add content, and collaboratively decide on the content of the artifact. Despite awareness of "Zoom fatigue" [38], online synchronous meetings were a popular way amongst our participants of undertaking ideation activities at the early stages of defining a feature, "Especially in the beginning of new projects where you need to be divergent and converge then I think Miro is the main tool we use to, to help, ideate, you know, to create new ideas or to collaborate more." (P17) We noted this model was also used for solution design, UI design, and system architectures.

Similar to the findings of [34] concerning collaborative writing, different approaches were taken to capture the content discussed in the session. One approach is for everyone to draw on a whiteboard and use turn-taking for discussion, "Anyone throws in, and then everyone can move and then, you know, we went through in a circle and everyone also had a chance to explain and give voice to their thoughts." (P07) An alternate approach was to assign a scribe whilst others discussed, "Well basically we've used Teams, and to get everyone like together in the meeting, and yeah I was sharing the screen and creating like a diagram with the team." (P05), or for everyone to contribute to the document simultaneously, "Confluence nowadays has a collaborative editing mode, where multiple people can be editing a document simultaneously, you see everybody's cursors. And so for the sort of wordier collaborations that's worked really well." (P13)

Participants were conscious of the **need to make effective use of everyone's time** in the synchronous session. Approaches to aid efficiency within the session included the use of an external facilitator to guide the group and ensuring key people are present. Additionally, preparatory activities were sometimes undertaken prior to the session, including the use of pre-prepared whiteboard templates so the gathering of input from participants was more targeted, "The architect, he created a kind of a template to talk about features. So he has some, it's kind of a workflow. So he has some questions, and some boxes, that's connected, and then you go there and fill it up." (P01), a draft artifact prepared beforehand, "So the minute we're talking about something where there's data flow, or like a lot of components involved, I think a diagram is very necessary. You can try and talk your way through it with words, and it just doesn't convey the same information. So sometimes when we have these discussions, I make sure to create a small PowerPoint with an image which shows the flow of data or the different components involved, and how they're interacting to make this feature work." (P16), and seeking input from participants beforehand "So I will send like one week before, like a Google Survey, to say, Okay, what are all the ideas that that you have...that also help us to have the different viewpoints from different people." (P23)

Follow-up concluding steps were sometimes used, such as a final asynchronous decision making step based on the meeting's discussions, "We would create a sort of like a Google Form and it would have, you know, three questions that's like of these which is your top choice, which is your second choice, which is your third choice.

You'd ask each of the participants to do that and for each question then they would choose one of the options that had been created in the earlier brainstorming session." (P15)

An advantage of co-creating remotely via a shared document or digital whiteboard is that remote individuals can easily multitask and use their computer to look into data, code, or other artifacts, thus leading to more productive meetings, "You can have that benefit of, I look it up, and I come back and say actually what we're discussing is not going to work because its not understood how it's already working, and so straight away you can discount something." (P22)

SS - Single Author & Synchronous Feedback

A single person authors an artifact in their own time and presents it to their colleagues in an online, real-time meeting for feedback. This appears a common approach for artifacts requiring a formal approval from a governing body such as an Architecture Review Board. One participant described this approach when preparing architecture decision records (ADRs), "We also do something within the development team where we document certain decisions that we've made as ADRs... and we review them every week." (P22)

Optionally, the document may be shared in advance of the review meeting so the participants have a chance to review ahead of time. This helps the meeting be more effective, "So after the tech spec is done. Again, it's shared in, in the channels, in Slack channel, prior to the tech spec review meeting." (P16)

The chief benefit of Model SS is that it provides quiet, focused time for an individual to craft an artifact suitable for wider presentation and discussion with others. It also allows experts to be utilized efficiently in the review meetings, "If there's different options and alternatives. Usually that's also when principals come to help guide the decision." (P16)

5.2 Example Work Chains

We found that the models are often **chained together**, resulting in periods of asynchronous and synchronous work to co-create the artifact.¹ For example, Model MS was used to generate ideas with colleagues which were then taken forward by a single person for analysis before presenting a refined artifact for review with collaborators (Model SS). Such an approach leads to a better understanding of the context of the artifact within the team as they have been involved in its production from ideation through to review, "So I certainly prefer that way of come up with some ideas. Take it away. You know, dive deeper into that. Come up with the pros and cons and alternatives, and then feed that back. So the meetings then become more reviewing something that you've already done, rather than just trying to come up with a design ad-hoc within an hour's session or something which we used to do." (P22)

Another participant described how it was commonplace in their team, for one person to author a design and share it for asynchronous feedback (SA) before scheduling a synchronous meeting to discuss and incorporate further changes (SS), "A lot of people now kind of do more prep in the proposal and of the diagrams they build. And then they share it out, everyone reads them kind of asynchronously over like a day or two. And then the person who's proposing this new

¹ Such chaining is different from the optional preparatory or concluding steps in each of the models; chaining creates distinct phases of work, whereas the preparatory and concluding steps are an integral part of a single activity.

design will schedule a meeting. And then will attend the meeting, will walk through the design and kind of edit the document and leave comments. So people can leave comments prior to that kind of sync-up meeting but we often just use that meeting to walk through what they built, add all the feedback.” (P14)

Another approach was for a single author to work asynchronously and solicit feedback synchronously (SS) before making changes and sending for an asynchronous final review (SA), “*Maybe put something in Powerpoint or just write it up.. And then we’ll come together as a group, usually with a few of the other development managers who have got good knowledge of the systems that I’m talking about and we’ll discuss the various options. I’ll take feedback from them, and then I’ll go back and update the diagrams and the solutions and then send them back to review.*” (P22)

Sometimes, the same model is **repeated** multiple times in a row to evolve an artifact until it’s finished. One participant described the repeated use of Model MS, “*And I started like just writing a few lines, describing, very high level use cases, ...at the beginning, those meetings are one hour long, because we need to brainstorm and do actual work, and then when it’s more an execution mode and we need just to get feedback, they used to last 30 minutes...We would have a couple of sessions of writing questions and iterating the sketch.*” (P04)

5.3 Matching Task to Model

We noticed that many of our participants used both asynchronous and synchronous approaches. This led us to discuss what, if any, factors were considered in determining the most effective approach when initiating a specific work task involving an artifact (e.g., detailing new features, solution design). The identified factors are shown in Table 2. We have categorized these factors into three different categories: nature of the task itself, cultural, and personal. The nature of the task itself consist of factors inherent to the actual task, whereas personal and cultural factors are more contextual based on the person undertaking the task and the workplace cultures.

Example reasons for taking a predominantly asynchronous approach (SA, MA) include:

- *Low complexity task*: If the task is not complex and likely to have few issues, “*They’re quite happy working asynchronously, ...people kind of know what they’re doing.*” (P04)
- *Team prefers few meetings*: Many developers dislike meetings [26] as it “*disrupts their flow.*” (P18)
- *National culture*: As noted by Hofstede [53], nations have different cultural values that effect communication. One participant noted that some colleagues, due to their national culture, felt uncomfortable interrupting men in conversation so “*feel more comfortable giving written feedback*” (P04).
- *Experienced author*: If the author is “*well versed with our areas of the technology*” (P16), then they are more comfortable working “*asynchronously.*” (P16)
- *Language barriers*: For developers whose employer’s business language differs from their primary language, they may be more comfortable communicating in written form rather than verbally in meetings as it gives them “*more time to think, how to express, how to communicate.*” (P04)

Table 2: Factors for Matching Task to Model

Factor	Description
Nature of the task	
Complexity	The level of complexity or uncertainty inherent to the task
Deadline	How quickly the task needs to be completed
Stakeholder availability	Key stakeholders are overloaded, so scheduling a meeting guarantees their full attention to offer their views
Cultural	
Team	The team cultural values include preferences for collective ownership or for avoiding meetings
Organization	Organizational culture such as preferred ways of working (e.g., meetings over asynchronous work)
National	National cultures impact an individual’s way of working
Personal	
Experience	Level of experience a person has within the team
Non-native speaker	Comfort level with using a non-native language such as English
Personality	Preferences for a way of working

Conversely, there were tasks where it was decided in advance that some periods of synchronous work (MS, SS) would be required to complete the artifact. Example reasons were:

- *High complexity*: Complex tasks (e.g., due to the scale of the task or the technicality) may be best progressed by holding an initial meeting to efficiently get team members perspectives before making a decision and moving to an asynchronous mode, “*Once we’ve narrowed down on some larger aspect, or like one vague idea, then we go into asynchronous.*” (P16)
- *Key stakeholders availability*: In some companies, key stakeholders (e.g., architects) can be involved in many projects and the only way to obtain their input is through a meeting, “*And there are some where you have to just sort of read through the doc again with them*” (P16)
- *Preference for meetings*: Some contributors prefer a meeting to discuss the task “*There will be people that say, can we just have a meeting to discuss, which can be frustrating.*” (P22)
- *Inexperienced author*: If the main contributor is new to the application or the role, then meetings are required more frequently to assist them in the task, “*it’s like they’re more unsure of, or need more guidance on, whether this is the right approach.*” (P16)

It appears that **organizational or team culture can have a role** in selecting a model or chain. One participant noted that artifacts were always authored by a single person to ensure it was completed “*You need someone to lead it...Otherwise, it’s never going to be done.*” (P22). Contrastingly, another participant noted their

team’s culture of different owners for different architecture components resulting in multiple authors contributing to the artifact, “*And this is why, you know, you have multiple people contributing to a given design.*” (P18). Moreover, the organizational culture needs to embrace asynchronous work for it to be effective “*I mean, you will get an email notification saying so and so started a new document, right. Do you have it in now as a practice to go look it up? Read it, respond to it in a timely manner.*” (P03)

When working on more complex tasks, our participants highlighted that **working asynchronously can feel unproductive** [11]. There is a sense that it takes **longer to evaluate new ideas** compared to co-located teams due to difficulties in gaining the attention of others, “*It’s a lot slower, when it comes to trying out new ideas, evaluating new ideas and all that. It probably takes more time to try and get the attention of every other person.*” (P08)

5.4 Tool Support for Co-creation

We observed what appears to be a core set of collaborative tools that aids teams in co-creation. However, challenges remain in finding important information related to the artifact and manually transforming the artifact’s content into other developer tools.

One integrated toolset supports all four models. Fig. 3 illustrates the collaborative toolset that we found to be commonly used by our participants to co-create artifacts. Whilst the same categories of tools were present, there was variety in the specific tools used within each tool category, as shown in Table 3.

Central to the toolset are artifact management tools such as UI design tools, digital whiteboarding tools, diagramming tools, and documentation tools (e.g., document editors, wikis). These tools are the primary ones to actually generate content. The resulting artifacts are stored in a secure, typically version controlled repository from where they can be shared with others by providing a link to the artifact in either a chat or a ticket in a task-tracking tool such as Jira. Chat and audio/video conferencing tools complete the integrated toolset and are used to communicate asynchronously and synchronously with colleagues depending on the needs of the situation. As observed elsewhere [2], chat in particular serves as a communication hub throughout the entire effort as it is the de-facto

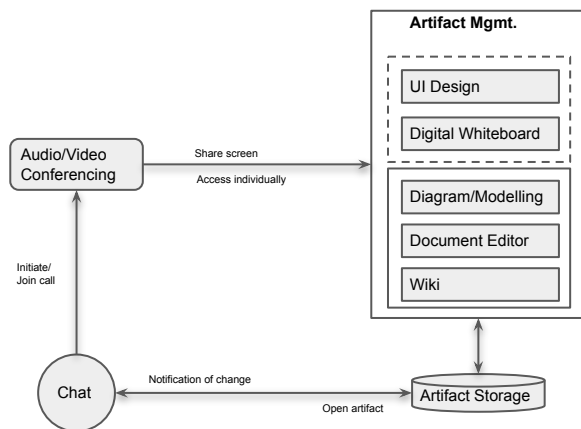


Figure 3: Typical Collaborative Toolset

Table 3: Examples of Tools Used to Co-create

Tool Category	Tools
Communication	
Chat	Slack, MS Teams, Google Chat
Audio/Video Conferencing	Slack Huddle, MS Teams, Zoom
Artifact Management	
User Interface (UI) Design	Figma
Digital Whiteboard	Miro, Mural, LucidSpark
Diagram/Modelling	LucidChart, Diagrams.Net, MS Visio
Document Editor	Google Docs/Slides, MS Word
Wiki	Confluence, Internal Developed Wiki
Artifact Storage	Google Drive, MS OneDrive

way of informing and approaching colleagues. Partly this is because chat tools can be readily integrated with other collaborative tools, which helps with team awareness [50] as many of the integrations use bots to notify users when changes have been made to artifacts [21]. As one participant stated: “*So you can have a JIRA and Confluence integration into Slack and a G Suite integration as well, and Zoom. Cool.*” (P10)

This integrated toolset supported our participants in both asynchronous and synchronous modes of co-creation. We observed that for synchronous co-creation, participants used multiple tools such as chat, audio conferencing, document editors, and diagramming tools, as one participant described their approach for discussing a feature “*You get on a call with Slack, and they talk through whatever feature it is that they want to solve. Somebody is a notetaker. So they take notes, and if there’s something that needs to be drawn or illustrated, or like, run through visually, then someone will just pull up a Microsoft Paint or something, and they will draw boxes.*” (P25)

Asynchronous modes of co-creation were also supported by document editors with their support for diagramming, together with collaboration features such as commenting and support for threads of conversation. As one participant noted about Google Docs, “*Google Docs is our go to tool...[It] allows us to iterate over our design and just work on ideas on Google Docs*” (P8), and digital whiteboards for sharing ideas “*We first created the LucidSpark board and sent out the link...And then through our group thread, we said hey don’t forget to add these.*” (P11)

Participants noted that some tools were particularly well-suited to specific tasks. For example, digital whiteboards were preferred for ideation activities “*We use Miro*” (P17), whereas document editors are better for lengthier documents, “*Requirements or for things that are longer, go put that in Google Doc, share that and you know have that collaborative kind of editing.*” (P19)

We noted that tools that did not support collaborative editing forced a single author approach, “*Diagrams that we do in Visio, you can’t have multiple people collaborate at the same time.*” (P22)

Not all tools are essential. Our participants expressed varying opinions on the value of UI tools and digital whiteboards. Even when a tool was available for use, it sometimes was rejected because of its steep learning curve, “*I tried to use Figma to build some UI but it’s too time intensive. I realized that it’s too much for me. So, I*

think I prefer to draw some new design on a piece of paper, I think I'm faster this way." (P01) Another reason is the inherent complexity of a digital whiteboard, as one participant noted "It becomes distracting fighting the tech [when doing something simple like] drawing a box." (P25) Some participants preferred the simplicity of more basic tools, "If we need to do something where we have to draw pictures or relationships, somebody literally just shares Microsoft Paint." (P12) This negative attitude toward UI tools and digital whiteboards was not universal, as some participants expressed strong preferences for certain tools, "If today you, you tell me. Oh, let's have a meeting. Let's collaborate about some topic, I would still use Miro because I like Miro." (P01) It appears personal preference has a large role in determining which tool is used rather than purposefully considering the best tool for a particular co-creation activity.

Difficult to find important information related to the artifact. This is due to **the use of multiple tools** when co-creating an artifact and also **searchability issues with document repositories**. For example, consider developers discussing the implementation of a feature inside a chat tool such as Slack. They subsequently create a Google Doc to describe the solution further. Finally, they hold an online meeting via Zoom to share and discuss with colleagues, with an accompanying sidebar chat in which additional deliberation takes place. Key information such as decisions, rationale, and risks are subsequently contained in multiple tools, "Information becomes kind of fractionalized between so many different tools that it can be hard to find what you're looking for." (P07) Participants also highlighted difficulties in finding documents within shared repositories such as Google Drive, "Sometimes you, you just spend hours sifting through Google Docs folders, just to understand maybe this folder in GitHub that you see." (P08) Reasons given included colleagues storing documents in a personal rather than a team folder, confusing folder hierarchies, and unnecessarily restrictive access rights, "I think that even the owners of those different folders and organizations, they have a harder time of keeping up with requests to get access." (P20)

Manual work to transform unstructured data. We found that manually transforming unstructured data from a co-created artifact in one artifact management tool into another tool was a common usage pattern. This led to additional effort to communicate the results of the co-creation activity to all team members. One source of additional effort was the **manual effort required to create follow-up development tasks** in a task tracker after the co-creation of an artifact. One participant noted that after they discuss and write up a design it gets "translated into a GitHub or an Asana project board." (P07) Whilst some tools provide mechanisms to automate such hand-offs from one tool to another, not all support this usage pattern. A second source of additional effort occurred when teams preferred to use one tool for its collaboration capabilities in co-creating an artifact, yet corporate standards required them to store the artifact in a different format and repository. As one participant described "Google Docs is used as a means to iterate and collaborate, but when something is set in stone, we move that over to GitHub and maintain as ReadMe markdown." (P8) This resulted in the **data being manually copied from one artifact to another**, thus compounding the information retrieval challenge highlighted earlier.

6 DISCUSSION

This paper contributes a first study that uses the lens of co-creation to examine how software teams produce artifacts. Rather than focusing on the use of a specific tool such as Google Docs, or a specific practice such as brainstorming, using the lens of co-creation provides a new way of examining and documenting collaborative team practices as an orchestration of people, tools, and activities that teams use to produce an artifact together. Doing so brings to light that teams structure their work in different ways, with different tradeoffs, depending on the situation, as fully remote teams are only too aware of the temporal and distance challenges [8] that remote work forces upon them. A novel finding is that one way in which teams have adapted to remote working is by adopting four common co-creation models through which they orchestrate small sequences of activities to produce artifacts to guide development. In considering the four co-creation models, teams **trade off various factors in selecting the most appropriate model** to use for a task. Working asynchronously offers time for deep thinking and may result in more potential solutions being identified, and it reduces the need for people to get together in real-time, freeing people from constant meetings. Using documents to host conversations also enables information to persist over time, something not easily possible when the conversation takes place in real-time over audio/video. However, this flexibility comes at a cost as it may take longer to reach a decision when people contribute and review in their own time. In contrast, having synchronous sessions involving a digital whiteboard or shared document may result in quicker idea generation and decision making, but introduces another meeting in busy schedules. It also may not be inclusive as some people may be unable to attend and some people are uncomfortable brainstorming online [60]. That said, participants often **chained models together in structuring an overall set of sequenced activities** through which they produce an artifact, enabling different participants to engage on their own terms.

A secondary novel finding is the identification of a set of factors that influence the decision making process of which model or chain to use, and that **go deeper than just matching task to model**. Factors such as deadline, stakeholder availability, personal experience, company and team culture all appear to effect which model (chain) is selected at the outset of an artifact generating task.

We also observe that the collaborative toolset shown in Fig. 3 is an enabler to all four models as it provides flexibility for both asynchronous and synchronous forms of co-creation, as well as supporting teams of developers in interchanging synchronous and asynchronous contributions. It appears that having an integrated toolset consisting of flexible tools that support the four co-creation models is more important than the specific tools themselves.

Implications for practitioners. We encourage remote teams to reflect on the four co-creation models and to **consider which model or chain of models is appropriate for a given situation**. In particular, the factors identified to match task to model should be explicitly considered before commencing a task, as this may lead to a more effective, and inclusive, approach. From our interviews, it is clear that some organizations are aware of the multiple models, but others tend to favor one model always to the detriment of

the overall collaborative effort. Opening up and considering that alternative approaches are available is crucial in such cases.

In terms of tooling, organizations should **ensure that their teams have an integrated collaboration toolset** similar to Fig. 3 that supports all four co-creation models and thus **empowers teams to determine the way they wish to collaborate**. This toolset should be accessible to all internal and external collaborators to minimize collaboration issues. Personal preference has a part to play in the acceptance and use of a tool by individuals, so organizations will need to balance the tension between an organization's need for standardization of tools against individual preferences.

Implications for tool providers. On the whole, we find that the collaboration toolset is considered sufficient to get the work done in the way participants want to work, "*Fine, they're good enough.*" (P21) However, there are still enhancements to improve the overall developer experience, which in turn can increase developer productivity and job satisfaction [9]. One issue is that crucial information such as decisions made or requirements considered cannot be easily found as people switch co-creation activities between tools. Existing mechanisms such as integration of tools to support information flow from one tool to another or automated identification of important facts within individual tools (e.g., key meeting points such as questions and consensus events [40] or essential snippets of information in Slack chats [61]) do not appear to address the challenge. Search across tools, too, seems to be inadequate at this time [30].

Another issue is to better support the flow of unstructured data captured in artifact management tools such as Google Docs or a sketching tool into the more structured format required by a task tracker such as Jira. Such improved support should go beyond simply copying the data and consider factors such as traceability, the need for developers to add detail, and being able to transition in the reverse direction to, for instance, revisit an issue discussion.

Implications for researchers. Our research has identified a number of factors used by teams to match a task to a model. These factors are related to the nature of the task, cultural aspects, and personal aspects, but we suspect this list is not exhaustive. It is likely additional factors (e.g., psychological safety) previously shown to influence the effectiveness of software teams [59, 3] also impact the use and effectiveness of the different co-creation models. Future research should determine if there are additional factors, as well as how the many factors impact collaboration effectiveness.

Given the expected growth of hybrid working [27], another important direction of future research is to consider how these models, chaining, factors, and supporting toolset are used or may change in hybrid environments. As one example, more complex tasks with higher levels of uncertainty may be best suited to chaining an initial synchronous model (MS) where the team can get together in-person for discussion at a physical whiteboard before one person undertakes detailed analysis and refinement with asynchronous feedback (SA). We speculate that our models provide a base foundation that, enhanced with a third dimension of colocation versus dispersed, can help articulate best practices in hybrid settings.

7 THREATS TO VALIDITY

Following advice by Creswell [5] for validating findings from qualitative research, we employed three validity strategies (reflexivity, peer debriefing, member checking) to consider the credibility, qualitative reliability, and transferability of the findings.

Credibility: The two researchers who performed the data analysis reflected on and shared their personal biases before coding. Both researchers have industry experience, with one researcher having significant experience working in distributed teams in the tech industry. There is a risk such experience could lead to biases being introduced into the coding steps. The researchers took care to analyze the transcripts based on what was said by the participants and not their own experiences. Additionally, the two researchers used peer debriefing to share the codebook, analysis, and findings for review and discussion with a third researcher. Member checking was also used by presenting our initial analysis and findings to the five participants in the second round of interviews. Our findings resonated with their experiences, which lends credibility to the validity of the models and observations about tool use.

Our study consisted of 25 participants with a risk they are not truly representative of the technology industry. We have tried to minimize this risk by selecting a set of diverse participants from a variety of companies and backgrounds. Two companies provided eleven participants so there is a risk their company culture could impact the findings. We believe this risk is minimized as the participants worked in different teams, roles, and had varying experience levels. Additionally, there is a risk that 25 participants are not a large enough group to draw conclusions applicable to the broader population. However, we felt saturation was reached in coding the first round interviews, as we did not determine any new codes or themes when analyzing the last five interviews. The second interviews were fewer and focused on discussing the models and identifying factors used to match a task to model. As discussed earlier, we recommend further research is required to expand on this set of factors as we do not believe this set is exhaustive.

Reliability: We have documented the study design in Section 4. The same two researchers were involved in the data analysis and through regular meetings used a consensus mechanism to resolve any disagreements in the coding. As discussed, a third researcher was involved across the entire analysis to monitor the process being followed and verify the analysis and its results.

Transferability: Our participants worked in several different industries, with participants predominantly based in the USA, so it is likely our results are transferable to software teams working in similar industries in the USA. However, American cultural work practices and norms likely have some impact on the experiences participants described to us. Further research in other countries is required to determine if our results are transferable to countries outside of the USA.

8 CONCLUSIONS

In this paper, we presented our findings on how fully remote software teams co-create artifacts. The results are based on an interview study with 25 software professionals working in industry. Our primary contribution lies in the identification of four models of co-creation, which document common ways in which teams combine

authorship, temporality, and tools to collaboratively produce an artifact necessary to progress work. Moreover, we observed these models were linked together into chains of action and various factors were considered when matching a task to a model. We noted a common, integrated toolset is used to support all four models.

For future work, we plan to dig deeper into the issue of remote work and creativity. Although not further pursued in the above discussion, a number of participants mentioned that they felt that remote co-creation reduces overall team creativity. This is a serious concern and appears to be driving the current trend toward hybrid work [57]. Our next goal is to build an understanding whether this is merely a perception or, if team creativity is indeed reduced in remote co-creation compared to colocated co-creation, what the underlying causes are and how they can perhaps be overcome – be it with new models, different chains, and/or new tools.

ACKNOWLEDGMENTS

We would like to thank our participants for their involvement in our study. This material is based upon work supported by the National Science Foundation under grant CCF-2210812. Rafael Prikladnicki is partially funded by Fapergs and CNPq, Brazil.

REFERENCES

- [1] Sebastian Baltes and Stephan Diehl. 2014. Sketches and diagrams in practice. en. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, Hong Kong China, (Nov. 2014), 530–541. ISBN: 978-1-4503-3056-5. doi: 10.1145/2635868.2635891.
- [2] Fabio Calefato and Filippo Lanubile. 2016. A Hub-and-Spoke Model for Tool Integration in Distributed Development. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*. ISSN: 2329-6313. IEEE Computer Society, Irvine, CA, USA., (Aug. 2016), 129–133. doi: 10.1109/ICGSE.2016.12.
- [3] Peter Cardon, Carolin Fleischmann, Jolanta Aritz, Haibing Ma, Ann Springer, and Scott Springer. 2022. The Influence of Psychological Safety and Personality on Technology Acceptance of Team-Based Technology in Global Virtual Teams. en. In *Proceedings of the 55th Hawaii International Conference on System Sciences*. (Jan. 2022), 634–643. doi: 10.24251/HICSS.2022.076.
- [4] Mauro Cherubini, Gina Venolia, Rob DeLine, and Amy J. Ko. 2007. Let's go to the whiteboard: how and why software developers use drawings. en. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, (Apr. 2007), 557–566. ISBN: 978-1-59593-593-9. doi: 10.1145/1240624.1240714.
- [5] John W. Creswell and Cheryl N. Poth. 2018. *Qualitative Inquiry & Research Design*. (4th ed.). SAGE Publications, Los Angeles. ISBN: 978-1-5063-3020-4.
- [6] Gabriele D'Angelo, Angelo Di Iorio, and Stefano Zacchiroli. 2018. Spacetime Characterization of Real-Time Collaborative Editing. *Proceedings of the ACM on Human-Computer Interaction*, 2, CSCW, (Nov. 2018), 41:1–41:19. doi: 10.1145/3274310.
- [7] Daniela Damian, Filippo Lanubile, and Teresa Mallardo. 2008. On the Need for Mixed Media in Distributed Requirements Negotiations. *IEEE Transactions on Software Engineering*, 34, 1, (Jan. 2008), 116–132. Conference Name: IEEE Transactions on Software Engineering. doi: 10.1109/TSE.2007.70758.
- [8] Julia Eisenberg and Aparna Krishnan. 2018. Addressing Virtual Work Challenges: Learning From the Field. *Organization Management Journal*, 15, 2, (Apr. 2018), 78–94. doi: 10.1080/15416518.2018.1471976.
- [9] Fabian Fagerholm and Jürgen Münch. 2012. Developer experience: Concept and definition. In *2012 International Conference on Software and System Process (ICSSP)*. (June 2012), 73–77. doi: 10.1109/ICSSP.2012.6225984.
- [10] Yulin Fang, Derrick Neufeld, and Xiaojie Zhang. 2022. Knowledge coordination via digital artefacts in highly dispersed teams. en. *Information Systems Journal*, 32, 3, 520–543. doi: 10.1111/isj.12358.
- [11] Denae Ford, Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Sonia Jaffe, Chandra Maddila, Jenna L. Butler, Brian Houck, and Nachiappan Nagappan. 2021. A Tale of Two Cities: Software Developers Working from Home during the COVID-19 Pandemic. *ACM Trans. Softw. Eng. Methodol.*, 31, 2, (Dec. 2021), 27:1–27:37. doi: 10.1145/3487567.
- [12] J.D. Herbsleb and A. Mockus. 2003. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29, 6, (June 2003), 481–494. doi: 10.1109/TSE.2003.1205177.
- [13] Irum Inayat, Lauriane Moraes, Maya Daneva, and Siti Salwah Salim. 2015. A reflection on agile requirements engineering: solutions brought and challenges posed. In *Scientific Workshop Proceedings of the XP2015 (XP '15 workshops)*. Association for Computing Machinery, New York, NY, USA, (May 2015), 1–7. ISBN: 978-1-4503-3409-9. doi: 10.1145/2764979.2764985.
- [14] Nicholas Ind and Nick Coates. 2013. The meanings of co-creation. *European Business Review*, 25, 1, (Jan. 2013), 86–95. Publisher: Emerald Group Publishing Limited. doi: 10.1108/09555341311287754.
- [15] Victoria Jackson, Rafael Prikladnicki, and Andre van der Hoek. 2023. Supplementary material for co-creation in fully remote software teams. (2023). <https://doi.org/10.5061/dryad.612jm6hw>.
- [16] Peter Jones. 2018. Contexts of Co-creation: Designing with System Stakeholders. en. In *Systemic Design*. Springer, Tokyo, 3–52. doi: 10.1007/978-4-431-55639-8_1.
- [17] Young-Wook Jung, Youn-kyung Lim, and Myung-suk Kim. 2017. Possibilities and Limitations of Online Document Tools for Design Collaboration: The Case of Google Docs. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. Association for Computing Machinery, New York, NY, USA, (Feb. 2017), 1096–1108. ISBN: 978-1-4503-4335-0. doi: 10.1145/2998181.2998297.
- [18] Eirini Kalliamvakou, Daniela Damian, Kelly Blincoe, Leif Singer, and Daniel M. German. 2015. Open Source-Style Collaborative Development Practices in Commercial Projects Using GitHub. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 1. ISSN: 1558-1225. (May 2015), 574–585. doi: 10.1109/ICSE.2015.74.
- [19] Jonas Kniel and Alice Comi. 2021. Riding the Same Wavelength: Designers' Perceptions of Shared Understanding in Remote Teams. en. *SAGE Open*, 11, 3, (July 2021), 21582440211040129. doi: 10.1177/21582440211040129.
- [20] Filippo Lanubile, Christof Ebert, Rafael Prikladnicki, and Aurora Vizcaino. 2010. Collaboration Tools for Global Software Engineering. *IEEE Software*, 27, 2, (Mar. 2010), 52–55. doi: 10.1109/MS.2010.39.
- [21] Carlene Lebeuf, Margaret-Anne Storey, and Alexey Zagalsky. 2018. Software Bots. *IEEE Software*, 35, 1, (Jan. 2018), 18–23. doi: 10.1109/MS.2017.4541027.
- [22] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E. M. van der Werf, and Sjaak Brinkkemper. 2016. The Use and Effectiveness of User Stories in Practice. en. In *Requirements Engineering: Foundation for Software Quality (Lecture Notes in Computer Science)*. Maya Daneva and Oscar Pastor, (Eds.) Springer International Publishing, Cham, 205–222. ISBN: 978-3-319-30282-9. doi: 10.1007/978-3-319-30282-9_14.
- [23] Nicolas Mangano, Thomas D. LaToza, Marian Petre, and André van der Hoek. 2015. How Software Designers Interact with Sketches at the Whiteboard. *IEEE Transactions on Software Engineering*, 41, 2, (Feb. 2015), 135–156. doi: 10.1109/TSE.2014.2362924.
- [24] Dusica Marijan and Sagar Sen. 2022. Industry–Academia Research Collaboration and Knowledge Co-creation: Patterns and Anti-patterns. *ACM Transactions on Software Engineering and Methodology*, 31, 3, (Mar. 2022), 45:1–45:52. doi: 10.1145/3494519.
- [25] [n. d.] MAXQDA | All-In-One Qualitative & Mixed Methods Data Analysis Tool. en-US. (). Retrieved Dec. 8, 2021 from <https://www.maxqda.com/>.
- [26] André N. Meyer, Earl T. Barr, Christian Bird, and Thomas Zimmermann. 2021. Today Was a Good Day: The Daily Life of Software Developers. *IEEE Transactions on Software Engineering*, 47, 5, (May 2021), 863–880. Conference Name: IEEE Transactions on Software Engineering. doi: 10.1109/TSE.2019.2904957.
- [27] Microsoft. 2021. The Next Great Disruption Is Hybrid Work—Are We Ready? en. (Mar. 2021). Retrieved Dec. 5, 2021 from <https://www.microsoft.com/en-us/worklab/work-trend-index/hybrid-work>.
- [28] 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. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. ISSN: 1558-1225. IEEE, Virtual, (May 2021), 624–636. doi: 10.1109/ICSE43902.2021.00064.
- [29] Sunila Modi, Pamela Abbott, and Steve Counsell. 2013. Negotiating Common Ground in Distributed Agile Development: A Case Study Perspective. In *2013 IEEE 8th International Conference on Global Software Engineering*. ISSN: 2329-6313. (Aug. 2013), 80–89. doi: 10.1109/ICGSE.2013.18.
- [30] Meredith Ringel Morris. 2013. Collaborative search revisited. In *Proceedings of the 2013 conference on Computer supported cooperative work (CSCW '13)*. Association for Computing Machinery, New York, NY, USA, (Feb. 2013), 1181–1192. ISBN: 978-1-4503-1331-5. doi: 10.1145/2441776.2441910.
- [31] Ricardo Olenewa, Gary M. Olson, Judith S. Olson, and Daniel M. Russell. 2017. Now that we can write simultaneously, how do we use that to our advantage? *Commun. ACM*, 60, 8, (July 2017), 36–43. doi: 10.1145/2983527.
- [32] Gary M. Olson and Judith S. Olson. 2000. Distance Matters. en. *Human-Computer Interaction*, 15, 2-3, (Sept. 2000), 139–178. doi: 10.1207/S15327051HCI1523_4.
- [33] Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2022. How Has Design Thinking Being Used And Integrated Into Software Development Activities? A Systematic Mapping. en. *Journal of Systems and Software*, 187, (May 2022), 111217. doi: 10.1016/j.jss.2022.111217.

- [34] I.R. Posner and R.M. Baecker. 1992. How people write together (groupware). In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. Vol. iv. (Jan. 1992), 127–138 vol.4. doi: 10.1109/HICSS.1992.183420.
- [35] Venkat Ramaswamy and Kerimcan Ozcan. 2018. What is co-creation? An interactional creation framework and its implications for value creation. en. *Journal of Business Research*, 84, (Mar. 2018), 196–205. doi: 10.1016/j.jbusres.2017.11.027.
- [36] Eric Raymond. 1999. The cathedral and the bazaar. en. *Knowledge, Technology & Policy*, 12, 3, (Sept. 1999), 23–49. doi: 10.1007/s12130-999-1026-0.
- [37] Jen Rhymmer. 2023. Location-Independent Organizations: Designing Collaboration Across Space and Time. en. *Administrative Science Quarterly*, 68, 1, (Mar. 2023), 1–43. Publisher: SAGE Publications Inc. doi: 10.1177/00018392221129175.
- [38] René Riedl. 2022. On the stress potential of videoconferencing: definition and root causes of Zoom fatigue. en. *Electron Markets*, 32, 1, (Mar. 2022), 153–177. doi: 10.1007/s12525-021-00501-3.
- [39] Johnny Saldana. 2021. *The Coding Manual For Qualitative Researchers*. en. (4th ed.). SAGE Publications, London, (Mar. 2021). ISBN: 978-1-5297-3175-0.
- [40] Samiha Samrose, Daniel McDuff, Robert Sim, Jina Suh, Kael Rowan, Javier Hernandez, Sean Rintel, Kevin Moynihan, and Mary Czerwinski. 2021. Meeting-Coach: An Intelligent Dashboard for Supporting Effective & Inclusive Meetings. en. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, (May 2021), 1–13. ISBN: 978-1-4503-8096-6. doi: 10.1145/3411764.3445615.
- [41] Elizabeth B.-N. Sanders and Pieter Jan Stappers. 2008. Co-creation and the new landscapes of design. *CoDesign*, 4, 1, (Mar. 2008), 5–18. doi: 10.1080/15710880701875068.
- [42] Anita Sarma and André van der Hoek. 2010. Categorizing the Spectrum of Coordination Technology. *Computer*, 43, 6, (June 2010), 61–67. doi: 10.1109/MC.2010.163.
- [43] Carolyn B. Seaman. 2008. Qualitative Methods. en. In *Guide to Advanced Empirical Software Engineering*. Forrest Shull, Janice Singer, and Dag I. K. Sjøberg, (Eds.) Springer, London, 35–62. ISBN: 978-1-84800-044-5. doi: 10.1007/978-1-84800-044-5_2.
- [44] Todd Sedano, Paul Ralph, and Cécile Péraire. 2019. The Product Backlog. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. ISSN: 1558-1225, (May 2019), 200–211. doi: 10.1109/ICSE.2019.00036.
- [45] Helen Sharp, Rosalba Giuffrida, and Grigori Melnik. 2012. Information Flow within a Dispersed Agile Team: A Distributed Cognition Perspective. en. In *Agile Processes in Software Engineering and Extreme Programming* (Lecture Notes in Business Information Processing). Claes Wohlin, (Ed.) Springer, Berlin, Heidelberg, 62–76. ISBN: 978-3-642-30350-0. doi: 10.1007/978-3-642-30350-0_5.
- [46] Sid Sijbrandij. 2023. GitLab's CEO on Building One of the World's Largest All-Remote Companies. *Harvard Business Review*, (Mar. 2023). Section: Business management. Retrieved Mar. 28, 2023 from <https://hbr.org/2023/03/gitlabs-ceo-on-building-one-of-the-worlds-largest-all-remote-companies>.
- [47] Darja Smite, Emily Laue Christensen, Paolo Tell, and Daniel Russo. 2023. The Future Workplace: Characterizing the Spectrum of Hybrid Work Arrangements for Software Teams. *IEEE Software*, 40, 2, (Mar. 2023), 34–41. Conference Name: IEEE Software. doi: 10.1109/MS.2022.3230289.
- [48] Darja Smite, Marius Mikalsen, Nils Brede Moe, Viktoria Stray, and Eriks Klotins. 2021. From Collaboration to Solitude and Back: Remote Pair Programming During COVID-19. en. In *Agile Processes in Software Engineering and Extreme Programming* (Lecture Notes in Business Information Processing). Peggy Gregory, Casper Lassenius, Xiaofeng Wang, and Philippe Kruchten, (Eds.) Springer International Publishing, Cham, 3–18. ISBN: 978-3-030-78098-2. doi: 10.1007/978-3-030-78098-2_1.
- [49] Darja Smite, Anastasiia Tkalic, Nils Brede Moe, Efi Papatheocharous, Eriks Klotins, and Marte Pettersen Buvik. 2022. Changes in perceived productivity of software engineers during COVID-19 pandemic: The voice of evidence. en. *Journal of Systems and Software*, 186, (Apr. 2022), 111197. doi: 10.1016/j.jss.2021.111197.
- [50] Viktoria Stray, Nils Brede Moe, and Mehdi Noroozi. 2019. Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. Montreal, QC, Canada, (May 2019), 111–121. doi: 10.1109/ICGSE.2019.00031.
- [51] J Teevan, N Baym, J Butler, B Hecht, S Jaffe, K Nowak, A Sellen, and L Yang. 2022. Microsoft New Future Of Work 2022. (2022). Retrieved Aug. 1, 2022 from <https://aka.ms/nfw2022>.
- [52] El Pa Pa Pe-Than, Laura Dabbish, and James Herbsleb. 2021. Open Collaborative Writing. EN. *Proceedings of the ACM on Human-Computer Interaction*, (Apr. 2021). doi: 10.1145/3449211.
- [53] [n. d.] The 6 dimensions model of national culture by Geert Hofstede. en-GB. (). Retrieved June 16, 2023 from <https://geerthofstede.com/culture-geert-hofstede-gert-jan-hofstede/6d-model-of-national-culture/>.
- [54] Thea Turner, Pernilla Qvarfordt, Jacob T. Biehl, Gene Golovchinsky, and Mari-beth Back. 2010. Exploring the workplace communication ecology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10). Association for Computing Machinery, New York, NY, USA, (Apr. 2010), 841–850. ISBN: 978-1-60558-929-9. doi: 10.1145/1753326.1753449.
- [55] Jim Ungar and Jeff White. 2008. Agile user centered design: enter the design studio - a case study. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '08). Association for Computing Machinery, New York, NY, USA, (Apr. 2008), 2167–2178. ISBN: 978-1-60558-012-8. doi: 10.1145/1358628.1358650.
- [56] Dakuo Wang, Haodan Tan, and Tun Lu. 2017. Why Users Do Not Want to Write Together When They Are Writing Together: Users' Rationales for Today's Collaborative Writing Practices. *Proceedings of the ACM on Human-Computer Interaction*, 1, CSCW, (Dec. 2017), 107:1–107:18. doi: 10.1145/3134742.
- [57] Tianna Xu, Advait Sarkar, and Sean Rintel. 2023. Is a Return To Office a Return To Creativity? Requiring Fixed Time In Office To Enable Brainstorms and Watercooler Talk May Not Foster Research Creativity. In *CHIWORK 2023*. ACM, (June 2023). <https://www.microsoft.com/en-us/research/publication/is-a-return-to-office-a-return-to-creativity-requiring-fixed-time-in-office-to-enable-brainstorms-and-watercooler-talk-may-not-foster-research-creativity/>.
- [58] Longqi Yang et al. 2021. The effects of remote work on collaboration among information workers. en. *Nat Hum Behav*, (Sept. 2021), 1–12. doi: 10.1038/s41562-021-01196-4.
- [59] Murat Yilmaz, Rory V. O'Connor, Ricardo Colomo-Palacios, and Paul Clarke. 2017. An examination of personality traits and how they impact on software development teams. *Inf. Softw. Technol.*, 86, C, (June 2017), 101–122. doi: 10.1016/j.infsof.2017.01.005.
- [60] Chien Wen (Tina) Yuan, Yu-Hsuan Liu, Hao-Chuan Wang, and Yuan-Chi Tseng. 2019. Gender Effects on Collaborative Online Brainstorming Teamwork | Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. EN. In *ACM Conferences*. Association for Computing Machinery, Glasgow, Scotland, UK, (May 2019). ISBN: 978-1-4503-5971-9. Retrieved June 16, 2022 from.
- [61] Amy X. Zhang and Justin Cranshaw. 2018. Making Sense of Group Chat through Collaborative Tagging and Summarization. *Proc. ACM Hum.-Comput. Interact.*, 2, CSCW, (Nov. 2018), 196:1–196:27. doi: 10.1145/3274465.