Organizing Oceanographic Infrastructure: The Work of Making a Software Pipeline Repurposable

ANDREW B. NEANG, University of Washington, United States
WILL SUTHERLAND, University of Washington, United States
DAVID RIBES, University of Washington, United States
CHARLOTTE P. LEE, University of Washington, United States

Drawing from a longitudinal case study, we inspect the activities of an expanding team of scientists and their collaborators as they sought to develop a novel software pipeline that worked both for themselves and for their wider community. We argue that these two tasks — making the software work for themselves and also for their wider scientific community — could not be differentiated from each other at the beginning of the software development process. Rather, this division of labor and software capacities emerged, articulated by the actors themselves as they went about their tasks. The activities of making the novel software "work" at all, and the "extra work" of making that software repurposable or reusable could not be distinguished until near the end of the development process — rather than defined or structured in advance. We discuss implications for the trajectory of software development, and the practical work of making software repurposable.

CCS Concepts: • **Human-centered computing → Computer supported cooperative work**.

Additional Key Words and Phrases: software pipeline; development; coordination; articulation work

ACM Reference Format:
Andrew B. Neang, Will Sutherland, David Ribes, and Charlotte P. Lee. 2023. Organizing Oceanographic Infrastructure: The Work of Making a Software Pipeline Repurposable. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW1, Article 79 (April 2023), 18 pages. https://doi.org/10.1145/3579512

1 INTRODUCTION

Scientific software tools and analysis pipelines are increasingly coming to be conceived as objects of coordination, as things that are mobile, reusable, generic, sustainable, or considered a community resource. As increasingly central resources in scientific practice, such tools have been swept up in broader movements, such as open science [13, 46] and reproducibility [17]. There are growing expectations that the software that researchers produce in the course of their work will end up finding new uses within their larger collaborative milieu, or even in other "domains" [56]. Prior studies have framed this problem as a tension between "specialist applications and generic solutions" [21], or between "professional end-user development" and coding for others [60], or as a transition from a "personal tool" to a "community resource" [74]. These characterizations frame slightly different problems, but they all bear close relevance to long-running discussions in CSCW, such as that on the creation of common information spaces (e.g., [2]) and on infrastructure [66].

Efforts to address these problems emerging around scientific software have encountered difficulties that are both familiar and unfamiliar to these discussions. It has become clear that developing

Authors' addresses: Andrew B. Neang, University of Washington, Seattle, Washington, United States, neanga@uw.edu; Will Sutherland, University of Washington, Seattle, Washington, United States, willtsk@uw.edu; David Ribes, University of Washington, Seattle, Washington, United States, dribes@uw.edu; Charlotte P. Lee, University of Washington, Seattle, Washington, United States, cplee@uw.edu.

2573-0142/2023/4-ART79
https://doi.org/10.1145/3579512

research software tools for others is a significantly different venture than developing that software for a single user or small scientific team. Making software reusable also demands quite a bit of "extra work" and increased coordinative complexity [74]. What new tasks are required, who will do it, and how that work will be organized within the context of academic research labs and career trajectories is an extremely pressing issue [59, 61].

In this paper we examine not just how people in the sciences manage the increased amount of work associated with the project of making repurposable software, but how they reorganize themselves around it. Through a longitudinal investigation; drawing on participant observation [14], documentary analysis, and historical investigation [63]; we develop a longitudinal account of the development of a research pipeline in the ocean sciences as it transitions from a tool developed for a single primary use case to one that supports many different kinds of research in neighboring subfields. We use the term pipeline here to frame a collection of scripts and software packages assembled by researchers into a set of sequential data processing and analysis steps. The pipeline in question is the software component of a larger instrument [i.e. a flow cytometer, 34] that aims to provide high-resolution maps of phytoplankton communities. This specific instrument and its attendant software pipeline, which here we will simply call SeaFlow, is intended for use by a number of closely related subfields in oceanography — in sum the instrument will generate data and findings for multiple disciplinary fields. SeaFlow and the data that it produces are also new resources for certain oceanographic research groups, and we track efforts to stabilize its novel use within these sciences — that is, we focus not on just how the pipeline is made to "work" for many disciplinary users, but also to make it "work" at all as a scientific instrument detecting phytoplankton.

Our initial analysis led us to ask: *How do members of an academic research lab organize their work in the process of making their research software pipeline repurposable?* We use the notion of infrastructure as a sensitizing concept to characterize what it means for the SeaFlow pipeline to become a community resource, or in our terms "repurposable". In particular we draw on one characteristic of Star and Ruhleder's [66] original formulation of infrastructure: the reach or scope of infrastructure across sites of work, and the attendant efforts to reconcile tensions between local and global requirements. Drawing on the concept of articulation work, we examine three points in the development of the pipeline where new relations between people and their respective tasks and responsibilities are worked out through interaction. As we will demonstrate, these inflection points were also articulation processes [69] at which significant distinctions were made in roles or kinds of work on the pipeline.

We argue that this coordinative work of negotiating different relations to the work of the pipeline is what allows it to become a repurposable resource. We also explore how the tasks that emerged in the process of making the software pipeline repurposable did not fall cleanly into a pre-existing matrix of discrete tasks and responsibilities. It is only over time that these actors came to define specific tasks that could be called the "technical work" of making the software reusable rather than the "scientific work" of making the instrument fit for the tasks of oceanography. In concluding sections of this paper, we examine how work roles — understood as negotiated in-situ, rather than clearly demarcatable in advance — have consequences for how CSCW research can and should inform software development itself. We provide further background on scientific software and CSCW research on infrastructure in the literature review and provide more detail on the pipeline under study in the site description.

2 LITERATURE REVIEW

2.1 The Reuse of Scientific Software

Scientific software has been a site for the study of collaboration for a long time, with studies focusing on the sharing of computational workflows [22], the coordination of software work [42, 73], and scientists as end-user developers [59]. More recently this stream of research has fed into a broader movement, involving funding bodies and researchers themselves, aimed at facilitating and improving scientific software development in a more proactive way. There have been calls for actively building communities around software development and maintenance in the sciences [24], and a number of institutes and organizations have emerged to facilitate this [44, 65, 75]. There are also programs such as the carpentries [8], which aim to train researchers in programming practice. The motivation for these efforts comes from a variety of distinct but connected issues related to scientific software, including notions of robustness, reuse [26, 40], reproducibility and replication [17, 67], sustainability [9, 53], and openness [28]. These varied concerns have produced broad and diverse efforts to improve the practice of software development in the sciences, including the creation of institutes and organizations for facilitating scientific software development [44, 65, 75].

While much of the literature has focused on software development practice (that is, the practices that researchers use in developing software and the development of best practices), there has also been increasing attention paid to this issue as one of organization and coordination. Attention has turned towards providing institutional organizational support for scientific software work [10], as well as towards the careers of the diverse array of people who make and maintain research software. This includes those that Berente et al. [4] collect under the term "cyberinfrastructure personnel", research software engineers (RSEs) [3, 11, 62] , and the various capacities emerging around data work [43, 51]. In these roles the problem of making software infrastructural becomes one of coordinating different kinds of work and professional identities. This is most evident in the different difficulties experienced by scientists and RSEs working on scientific software: software work does not easily fit into scientists' reward structures [27], while conversely RSEs have a hard time constructing a career trajectory within the academic and laboratory reward structures in which they find themselves [9].

What we want to draw out of these different observations is that the project of making scientific software reusable, replicable, sustainable, and so on, has a number of coordinative dimensions. In many sciences people are reorganizing around this demand, both in terms of how researchers collaborate with each other, and also in terms of the emergence of new roles and kinds of work.

2.2 Organizing Infrastructure

CSCW and related fields can provide a wealth of literature on how information systems "scale up" and come to be shared across communities of people. The concept of common information spaces captures the effort to put information "in common", requiring the effort of different groups to understand each other's contexts (of production and use) across potentially large physical and temporal distances [2]. Bertelsen and Bødker [5] demonstrate that for such large information spaces, the goal is not a summative, uniform view across all people, but rather views respective to a person's locale and their immediate responsibilities and information needs. This points our attention towards a more varied topology of visibility and accounting between different coordinating people and groups.

Star and Ruhleder [66] note that infrastructure must reach across sites of work and their attendant problems, and reconciles the different needs and interests of each. In this understanding a distinction is produced between "global" and "local" aspects of the pipeline; parts of the pipeline which are specific to the activities of different groups and work which is deemed general across them. In this

way it is made ready-to-hand for different use cases. Studies of research infrastructure, as well as "cyberinfrastructure" projects, have examined a variety of problems that come along with the proposition of making research tools and data formats into community resources. These include problems of coordinating diverse groups of stakeholders [37], sustainability and change over time [12, 30, 54, 54], and the construction of communities [32, 55]. While many of these are relevant to the problem of making robust software, we focus specifically on the evolving interactions between different stakeholders.

In grappling with the oftentimes non-linear development of infrastructural systems, notions of emergence, defined as processes whereby action and interactions result in global behavior of the system [58], occasionally appear in the CSCW literature [38, 45]. This aspect of ongoing change is explicitly modeled in the concept of "infrastructuring" [31, 33], which focuses on the way that infrastructures are shaped and reshaped in an ongoing way by communities. In this characterization change is continual, and continuity is produced in the juxtaposition of new uses and designs with older forms. While we are not adopting a participatory design approach, this work guided our analysis by pointing us to the ways that infrastructures are continually reshaped through the layered and successive interventions of human and technical actors [49, 50].

The project under study in this case is not an infrastructure already, but rather is an endeavor to make a tool infrastructural. Our focus therefore is not only on change over time, but specifically on the transition from a personal tool, designed in a top-down way, to the creation of a resource that is responsive to different stakeholder groups, and embedded in their diverse practices. In order to render this process in detail, we draw on the concept of articulation work and the segmentation and differentiation of subworlds [68]. The concepts of articulation work and articulation process provide a robust framework for examining how divisions of labor come into being on projects and teams [1, 15, 16, 23]. The notion of articulation work [69] points our examination towards the kinds of work involved in shaping and relating other tasks in an ordered work process. In this way our longitudinal investigation provides a handle on tasks and the interrelating of tasks as an emergent phenomenon in the development of software that works for one task and can also be repurposed to another.

In more recent years, the concept has also served as an analytical lens to help deepen the understanding of managing globally distributed software development work [20, 41]. Boden et al. [6], for instance, use it to highlight the importance of informal, often invisible processes in the organizing of remote software development projects. We use it, firstly, because it aims at capturing, as Strauss puts it: "how organizational process contributes to project order" [70]. Secondly, it includes a focus on temporality in terms of actors' management of contingencies. While this is often used to account for breakdowns or disruptions, we consider contingency and emergence as characteristic of the research software we are studying here. The segmentation of subworlds was an earlier foray of Strauss' into examining the somewhat larger junctures between subworlds and how different subworlds are differentiated or come to be connected [68].

3 RESEARCH SITE AND METHODS

Our research site is a scientific instrument development project which has been ongoing since 2008. The goal of this project is the development of an underway flow cytometer and its attendant software pipeline, together called SeaFlow, which has been developed for creating high-resolution maps of the distribution and abundance of marine phytoplankton in surface waters. These plant-like organisms are the foundation of the marine food web and responsible for nearly half the earth's annual production of oxygen and organic carbon [18]. As a result, capturing information on the temporal and spatial distributions of phytoplankton communities is critical for understanding how these organisms shape and are shaped by their environmental conditions (e.g. nutrient availability,

temperature, and light). Flow cytometry has served as a powerful technique for studying the dynamics of phytoplankton for the past three decades by allowing researchers to measure the optical properties of thousands of individual cells per second [64]. However, the instrument described in this paper was designed specifically to capture the smallest kinds of phytoplankton in ocean environments, with the goal of better understanding where these tiny organisms are found in the open ocean, how they are responding to environmental perturbations (e.g., warming oceans) and why.

The larger project encompasses the development of the hardware instrument, the deployment and data collection process, the development of software for processing and analyzing the data, and finally the process of publishing or distributing the data products. The term pipeline could productively be applied to this whole process, but here we focus on the software for processing and analyzing the data. In more formal software engineering and computer science definitions a pipeline is constituted by a set of sequential, software-defined processing steps, each of which accepts the output of another as input. In research settings these might consist of a mix of custom scripts and software packages assembled by researchers for the purpose of transforming and analyzing data products. Pipelines are closely related to "workflows" [22], and a pipeline could itself be collected together and shared as a singular software "package." The notion of a pipeline also forefronts the relationship between software development work and a sequential process of data transformation and analysis [7, 47].

In the case of the SeaFlow project, the software pipeline turns the output of the flow cytometer into data usable in further analysis or publication. In operation, the SeaFlow flow cytometer sits aboard an oceanographic research vessel and receives a continuous stream of water from the ocean via an intake line provided by the vessel. The cytometer operates by passing this seawater across a laser and measuring light scatter from pigments within their individual cells — these data require significant further processing for human interpretation and use. The "SeaFlow pipeline" in our usage is the set of sequential software-defined processing steps developed to take these early stage data products and transform and analyze them. The entanglement of software and data was therefore a connection that was enforced by our field site. When we asked questions about the structure, accessibility, publication, and format of SeaFlow flow cytometry data, the responses often led back to the constraints and contingencies of software development work.

The project is situated within a biological oceanography research lab at the University of Washington, but it is only one of a number of ongoing research projects in the lab. By the time of writing the SeaFlow project had become a small sub-team within the lab, including a number of research scientists, technicians, an instrument engineer, as well as students who come and go from the project. The core of the group developing the pipeline are therefore located within the same lab and meet once a week in that context. However, as the project grew, it came to involve the interests of individuals and communities in other labs and institutions. In this sense we have followed the nascent stages of the pipeline as an infrastructural resource, with the pipeline only taking on some of the qualities of infrastructure towards the end of the first author's observation.

3.1 Methods

The fourth author negotiated entree with the PI of a lab involved in several projects, including the SeaFlow project, beginning in 2008. She has, with collaborators, subsequently maintained relations with members of the field site. At the time of writing, the first author has been studying the SeaFlow project for six years as a participant observer, attending weekly project meetings, ad hoc meetings, and conducting interviews concurrently with observation. Our initial interview sample focused on members of the SeaFlow project team, including those who attended weekly project meetings, anyone identified as being a contributor to the project by the principal investigators, and those

noted as being interested in using SeaFlow data during meetings. This resulted in interviews with 12 participants, as well as follow up interviews with core members of the SeaFlow project. In total, 22 interviews were conducted.

The first author's period of observation began in 2015, but the period of the development of the pipeline that we trace in this paper spans from 2008 onward. Recollections from interviews, scientific publications, and other documents were used to fill out the early years of the project, as well as discussions about the past which occurred in lab meetings, during the initial enrollment of the field site, and observations by the first author and a research assistant. Documents included software code on GitHub, data collection and instrument operation protocols, and email exchanges between members. Interviews were also conducted with previous members of the project. For instance, core members of the project who left in 2014 were interviewed in 2016, at their new positions at other universities, to gather their recollections of working on the project.

The first author continuously inquired about how they collected, processed, analyzed, and shared data from different oceanographic research cruises. In addition, the first author regularly chatted more informally with participants about how the software pipeline was being used and iterated upon. We have attempted to collect a wide range of perspectives on the project. However, the primary research scientist leading the project, Roger, was a central source of activity in initiating and maintaining connections with these other stakeholders, and for that reason he plays a prominent role in our narrative. Accounts provided by other members of the team and external collaborators are used for constant comparison [14] throughout. We use pseudonyms to refer to all project members in this paper.

3.2 Data Analysis

Throughout data collection we employed a qualitative data analysis approach in which concepts, categories, and theories were developed in a process of frequently returning to the empirical data [14]. Our initial open coding helped characterize the overall landscape of the SeaFlow team's research work: how they produced data, components of their software pipeline to further probe, and various external collaborators involved in the project. This focused our attention in particular on the progressive size and variety of the group of people associated with the project. Through directed coding we developed more specific concerns focused on how the SeaFlow project grew and relationships established with new stakeholders, such as downstream data users. Codes were discussed and refined collectively by the authors, focusing especially on making sense of what was meant by categories deployed by those involved in the SeaFlow project, such as "user" and "developer." These codes were refined through discussion amongst the authors and lead to theoretical sampling of follow-up interviews with specific stakeholders. Some codes were separated off as being out of scope (the work of data collection, for instance) as the focus on software development work grew. Themes that were covered in our final set of codes involved on the one hand the different kinds of work necessary for making a data type generalizable, and on the other the coordination of specialized skill sets required to make this work possible.

4 FINDINGS

The longitudinal character of our engagement with the SeaFlow project allowed us to examine the pipeline in terms of its long-running and continual organizational change. Beginning in 2009, the SeaFlow pipeline expanded from a set of scripts developed primarily by one researcher to the work of a small team engaged with multiple external communities of researchers. We focus on particular points of articulation in order to organize the findings below through four vignettes. These are not points in time, but rather significant interactions between important stakeholders through which work processes were negotiated and reordered. Some of these interactions extend

over years. The first vignette describes the earliest stage of development and serves as a basis for comparison with the following three. Vignette 2 follows the effort of a student to make use of the pipeline in a novel geographic location, which constituted the project's first "user"; vignette 3 follows the work of a research consultant who joined the project and took on "technical" tasks of "hardening" and optimizing the software of the pipeline; and vignette 4 follows a more distant reuse as a unrelated research team adopted the pipeline's data products for their own use cases. This could be described as a process of the tool becoming a community resource [74] or becoming "generic" [52]. Here we will use the term repurposable because it highlights the effort involved in making a tool work for a new purpose. Across this trajectory, the pipeline gains the property of repurposability, becoming easier (but not easy) to adapt to novel goals. Simultaneously, the labor of working on the pipeline becomes parsed into its "scientific" and software "technical" components. As we will show, achieving repurposabability and distinguishing the scientific from the technical happen together, emerge over time, and were simply not there at all in the earliest stages of development – which is where we begin.

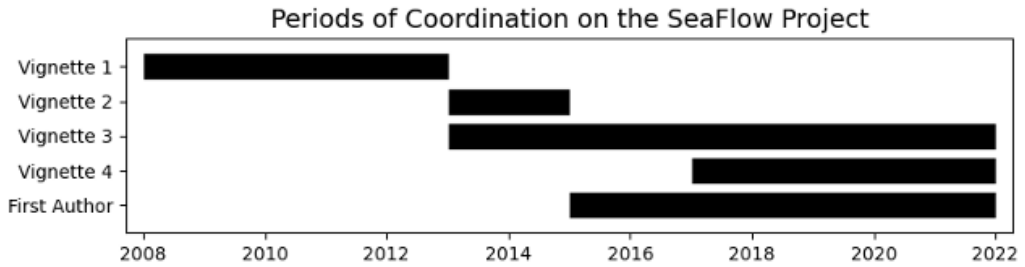

Fig. 1. Bars indicate the time over which the coordination described in each vignette occurred. The bottom bar indicates the period of the first author's involvement in the field site.

4.1 Vignette 1: Making SeaFlow Work (for Roger)

During the early period of the development of the pipeline, it was meant primarily for a particular scientific use case of interest to Roger and members of his lab. The pipeline itself was a collection of scripts, written in R, and developed primarily by Roger himself. Over a number of years, Roger accompanied the SeaFlow instrument on regular oceanographic research cruises between Hawaii and Seattle. On these cruises he developed the software in "real time", processing data generated by the instrument and altering his scripts to address problems and anomalies that emerged. The criteria for the development of the software at this point were the contingencies of a specific research problem: representing specific populations of phytoplankton in the Pacific Ocean in terms of abundance counts (the number of organisms) while the research vessel was underway. The data produced by the pipeline could potentially be quite useful to members of his lab and to the larger field, but when asked about whether he intended the early version of the software to be used by others, Roger emphasized that at this point he was primarily focused on the immediate scientific problem of making sense of the data produced by the SeaFlow instrument:

> "And it was not really meant originally...SeaFlow was really at the start...I didn't know if other people would be interested in this data anyway. So really, we didn't build it in the sense that, 'oh this is going to be really...It has to be flexible, it has to be...We're going to be having different people using it.' So it was really not built in this direction.

> Because that was not the first concern. We had first, can we make sense of this data, and so flowPhyto was more like an exploratory tool. And then when we were like this data seemed to make sense. And then we realized that flowPhyto was not really built to share with other people" (Roger, Research Scientist).

Roger refers to the pipeline with an earlier name of the software, "flowPhyto". He contrasts the exploratory nature of the software at this point with "flexible" software to be used by different people. Initially he was surprised that it was even possible for his pipeline to measure the small celled phytoplankton called prochlorococcus (early on, in portions we do not recount, he was not even sure if what he was seeing was a real organism, dismissing it as 'noise'). Working at sea, Roger ran through the whole pipeline process repeatedly, running seawater through the instrument, processing the data immediately with his scripts, and examining the output as visualizations. Anomalies in these visualizations came to turn Roger's attention, for instance, to the alignment of the laser in the flow cytometer, or to a particular part of the code where misaligned readings are filtered out, and so on. Roger moves from earlier steps to final output and then back, reworking parts of the pipeline in relation to what he sees in the data outputs. Through this iterative redevelopment of the pipeline Roger attempts to remove noise and compare what he is seeing to predictions about population changes in prior studies, and it is only at the end of this 'tuning' process that he begins to become confident in the results that the pipeline is revealing. This work was done entirely in the Pacific Ocean, and with a focus on what another member of the SeaFlow project would call the "blue blue waters around Hawaii" (Yvonne, Research Scientist).

This period of development could be characterized as a kind of end-user development insofar as the goal of developing the software was a specific scientific use case, rather than for a large number of users with varying needs [35, 59]. It is also similar to what Howison and Herbsleb [27] call "incidental software", which is motivated entirely by its usefulness for an immediate scientific problem. While Roger and others in his lab were aware that the pipeline and the data that it produces could be more broadly useful, at this stage Roger was not yet engaging the divergent needs of different stakeholders nor the concerns of sustainability and robustness of the software. Rather, the primary concern was whether the software could be made to work at all as an instrument for sensing particular populations of phytoplankton. In the following anecdotes, Roger along with new contributors, will attempt to make the pipeline useful to other research problems, which means making it useful in new environments (coastal waters), using it to detect different species of phytoplankton, and representing those populations with new kinds of measures. As Roger states above, it was not until the pipeline was turned to new use cases that developing the generic or repurposable characteristics of the pipeline became salient.

4.2 Vignette 2: The First New Purposes for the Pipeline

Our second vignette follows an inflection point in the development of the pipeline as a new doctoral student, Harriet, sought to use the scripts that Roger had developed for her own research projects. This constituted the first of a number of efforts to repurpose the SeaFlow pipeline to a new use case and prompted a great deal of change to the software of the pipeline as well as to the processes by which it was developed.

Harriet was a doctoral student within the same laboratory as Roger but was attempting to use the pipeline for her own research, which differed from the situation in which Roger had initially developed it. Her research involved capturing the presence of populations of phytoplankton in a new site of data collection, the Amazon Plume region at the mouth of the Amazon River. This effort surfaced a variety of issues. For instance, in parts of the cruise where the vessel entered an Exclusive Economic Zone data could no longer be collected (by treaty) thus producing gaps in the

data collection; In another situation all of the phytoplankton had been killed because the water line had previously been bleached. Taken together these introduced a large number of gaps into the datasets that would cause Roger's scripts — developed in conditions of continuous unhindered data production — to produce errors.

Addressing these problems became a coordinated effort between Roger and Harriett, but it was initially difficult to figure out how the work could be accomplished between them. In an interview, Harriet describes some frustration with the process:

> "I think one difficulty that I perceived was that there wasn't good communication between the users and the developers. So like I was pulling extra hours, putting band-aids on code, and I'm not sure anybody really knew what I was running into. Sometimes I would- I think I submitted a couple of bug reports but mostly I just wanted to fix it myself, because I wanted to have it work well enough so that I could have a chapter, And I didn't want to have to wait" (Harriett, Ph.D. Student).

Here Harriett uses the terms "users" and "developers" as a way of capturing a nascent distinction in tasks in which the students would surface issues that emerged in their particular case, and Roger would address them.

However this distinction is rendered more visible (to us and them) as it broke down [66]: Harriett expressed that she was under the time pressure of finishing her dissertation, and Roger did not have the time or energy to solve all the issues that were emerging. The result was that she ended up making a lot of "workarounds" to get her version of the code working, rather than "universal fixes" which would become permanent parts of the SeaFlow pipeline to be used by others. For instance, she developed a more robust system for "flagging" data to deal with the problem of missing values [29], but she never contributed it back to Roger's version of the code, and it remained her own custom code. Part of the difficulty also involved the tools for sharing and integrating code: Roger was initially sharing versions of the scripts by email, and even when the project migrated to GitHub neither Roger nor Harriett were particularly familiar with how to use it.

Repurposing the pipeline to this new project required addressing a variety of problems in getting the software to run under new data collecting conditions, but it also meant working out a coordination routine between "users" and "developers" on the team. Transitioning from a single person's end user development project — Roger with his synoptic view of all aspects of the software pipeline – to adopting and adapting the code of a new user, with new purposes for the pipeline, demanded the development of a new coordination routine and toolset across these contributors. Distinctions between use and development, as well as the distinction between workarounds and universal fixes, are relevant at this later time whereas they would have been meaningless in vignette 1 when Roger himself was developing the system for a single, immediate research problem. A new use for the pipeline was indeed accomplished — Harriet used it for her dissertation research — but it was somewhat awkward and frustrating to those involved due to the fact that they were in the process of working out the relationships and routines that could support repurposing.

4.3 Vignette 3: An Emergent Distinction Between the Scientific and the Technical

Around the time of the events described in vignette 2, Brandon, a research consultant, began working part time on the SeaFlow project. Brandon had long worked in the laboratory doing "script jobs" for various researchers. Because of his software development experience, he came to play a significant role in separating off tasks such as optimization and data storage and manipulation, and finding ways of improving them.

In an interview, Brandon described how he and Roger would use distinctions of "scientific" or "technical" issues in order to prioritize and delegate work on the pipeline. These were important ways of figuring out who would work on what:

> "So in terms of how we go about determining what's most essential, we try to separate issues or concerns based on whether they are technical or perhaps more scientific in nature as a first pass" (Brandon, Research Consultant).

He parsed things like optimizing certain steps of the data processing or moving the whole system to a new file format as technical issues, while scientific concerns were things like assessing whether changes in the data reflected instrumental effects or changes in phytoplankton populations.

Over time, some aspects of the work became firmly designated as technical matters. Some tasks, such as optimization and database design, were so exclusively handled by Brandon using specialized techniques, that Brandon worried that Roger would no longer be able to work on them. In discussing how the development work had become more complex, Brandon referenced this divergence in delegations as a primary concern:

> "So one of the things I think about is, ok, I understand how it works, but is Roger going to be able to maintain this code himself? And I don't know if the answer is yes...These [changes] are more complicated than just writing your normal R code, so, one of the considerations is, are the people who are going to be responsible for maintaining this software in the future...am I adding too much complexity and maintenance overhead in the future? Is that justified by the value we're adding?" (Brandon, Research Consultant).

This is an example of tasks that, over time, were strongly distinguished between contributors to the pipeline. Roger and others continue to develop other aspects of the pipeline, without much ongoing insight into the problem of optimization. It is precisely this separation of concerns that worries Brandon in relation to the prospect of maintaining the software over time. Our focus here is not so much the issue of maintenance specifically, but the increased complexity in task division that becomes evident in Brandon's concern.

Other kinds of work were not so easily separable. Brandon initially suggested that he was not as closely involved in the scientific concerns raised by Roger or Yvonne (a research scientist on the SeaFlow project):

> "I should clarify that I typically don't work on the more scientific concerns you hear Roger or Yvonne discussing but it's still important for me to have a handle as to what those issues are since it can definitely have implications on what changes or refinements need to be made to [SeaFlow]. So, like, the work around determining cell abundances of different populations and assessing whether that count is accurate. For instance, did the instrument's laser affect the count somehow or is it because the cells themselves have changed?" (Brandon, Research Consultant).

Note here the articulation of kinds of work, as Brandon distinguishes his own work from that of Roger and Yvonne. He also describes scientific work as a separate sphere of activity which he does not work on, but which he needs to track closely because these "scientific" concerns have close contingency with how he needs to resolve "technical" concerns. As a scientific concern, he gives the example of evaluating results from the pipeline as an effect of the instrument or of changes in the populations of phytoplankton themselves. However, he immediately corrected himself and suggested that the example he had picked was actually not entirely a scientific one:

> "But now that I think about it...this issue is not really just a scientific concern one per se. I guess it can also be considered a technical one in that I've been exploring ways to

> automatically find the bead coordinates and incorporate that into Popcycle's filtering step" (Brandon, Research Consultant).

Brandon explains that technical work on automatically finding beads used for calibrating the instrument (see also [34]) would help resolve the issue of assigning data output to instrumental effects or changes in the phenomenon. In this sense the scientific problems engaged by the team are highly contingent on his technical work to the point that it is not possible to cleanly distinguish it as a separate concern or as kinds of labor.

These examples demonstrate a gradient of problems, from situations where technical matters and scientific matters were not cleanly separable (or not yet separable), to situations, such as with optimization, in which a relatively firm distinction was accomplished between Brandon's work and the work of other contributors. As Brandon indicates in the first quotation above, this was a process. Through ongoing discussion in meetings and the repeated delegation of work, tasks such as optimization could be distinguished from work now considered only scientific. These distinctions had substance, in that they serve to articulate labor and facilitate coordination across a complex interdependent task, and, as Brandon points out, they have consequences for the ongoing maintenance of the project. However, these distinctions had to be worked out through the repeated negotiation and delegation of work; they were not tidy, innate distinctions available to begin with, and in some cases they never became so.

4.4 Vignette 4: "Downstream" Data Users

This final vignette describes an effort to repurpose the pipeline for a new community of downstream data users which occurred some years after the first effort described in vignette 2. By this time the SeaFlow pipeline had become a more robust tool, with data products from the pipeline had begun to circulate amongst oceanographers beyond the original development team. Here we focus on Roger's efforts to effectively account for the work of pipeline development (and data production) to more "distant" (geographically, disciplinarily, and in terms of integration with the original development team) goings-on within the SeaFlow project with the work of data analysis occurring in these different "downstream" sites of work.

Beginning in 2018, the SeaFlow team developed a new branch of the pipeline which would produce a new data type, carbon biomass measurements. This new data type would represent populations of phytoplankton in terms of the amount of carbon biomass, rather than number of organisms, and it could potentially connect the pipeline project with a new group of downstream stakeholders in the fields of chemical oceanography and oceanographic modeling who needed that measure. A chemical oceanographer at the University of Washington explained how this relationship emerged:

> "Roger helped us partly because we requested it and he also saw the value in doing so. I mean we're not the only ones that need the data to be in carbon units, there are other chemists and modelers in the community who share this need as well" (Iris, Chemical Oceanography Faculty Member).

Iris's reference to "the community" here gathers together a handful of research groups who would need carbon biomass measurements in order to use SeaFlow data, including the modeling groups, who also used carbon biomass rather than abundance counts. Through meetings with Iris and other researchers in chemical and modeling groups, Roger established carbon biomass measurements as a shared need across many groups, and in producing these measurements this new "community" would be brought into the orbit of the pipeline as an important group of stakeholders.

Configuring a relationship with this new group of stakeholders required further work, however. In particular, when Roger met with a research scientist from a modeling group, she indicated that

they would need error estimates, which provide a statistical account of the uncertainty in both the accuracy and precision of the SeaFlow instrument's measurements. Ideally, these estimates would be propagated through the pipeline's various steps to account for cumulative error which might emerge in the final product. The provision of error estimates was also a non-trivial endeavor in that it subjected the processes of the pipeline to a new set of evaluative criteria. Roger described it as a benefit of working with modeling groups that they would question the quality of the data in ways that he felt contributed to the rigor of the pipeline's processing steps:

> "Because they are people who — they don't want just a number. They want to have a whole — how confident you are that this number is this? What tools, what statistics did you use to infer this data?" (Roger, Principal Research Scientist).

These requirements are in part an aspect of the distance between the work of the modelers and the work of developing the SeaFlow pipeline. We use "distance" here with manifold meanings, such as geographic dispersion, but also in terms of new user's familiarity with the instrument and processing steps. Roger described a kind of "transparency" between these two sites of work, and compared it to processes in place in other highly established data production pipelines:

> "They forced us biologists to make the whole process transparent. For example, right now, with Griffin we are trying to estimate uncertainties in every step, which is if you look at NASA projects, they do this. Of course, we have to do this. How do you know that the data that you give is any good?" (Roger, Principal Research Scientist).

Roger and other members of the project had to actively produce an account of their work which could make it comprehensible to the modelers. "Transparency" here was not some full or complete visibility into the details of the code or the development process, but rather a minimum necessary account that would facilitate work between the pipeline and a number of "downstream" sites of work. As long as the correct form of the data and appropriate metadata were provided (carbon biomass measurements and uncertainty estimates) the chemical and modeling groups could work with the data products despite having little interaction with or access to the development of the pipeline or the specifics of the software.

The relationship configured with this new community of data users was distinctly different from our previous examples, as it involved an arm's length relationship with a larger group of researchers. Here we can see the construction of a community and their requirements, not unlike the construction of a "generic need" [52], as well as the configuration of that relationship through the design of data and metadata. Although particular practices, such as providing error estimates, were not needed in interactions with other stakeholders within the lab, they eventually became commonsensical aspects of the relationship with this new community of users, as is evident in Roger's suggestion above that "...of course we have to do this". As we have shown throughout, novel technical capacities for the pipeline were accompanied by organizational change and pathways for coordination across heterogenous and "distant" users. While we have discussed one instance of coordination across heterogenous disciplinary groups, it was understood that carbon biomass data would thereafter also be of use to additional user groups. At this later time multiple stakeholders were involved in developing metadata, for and of the pipeline, for the purposes of serving additional scientific communities and in anticipation of currently unknown research questions. This process resulted in the extension of the software at many points in the pipeline in order to generate the appropriate metadata, as well as prompting the SeaFlow project team to reevaluate the pipeline in terms of statistical error.

5 DISCUSSION

The longitudinal view of the SeaFlow project allows us to examine how diverse and interdependent tasks were negotiated and redelegated over time. We first discuss how this highlights ongoing organizing work as part of the project and the difficulty of making research software repurposable. We then connect this organizing work, the production of interdependent roles and responsibilities, to core tensions in the larger problematic of research software.

5.1 Articulating Pipeline Development Work

Our characterization of the SeaFlow project's transition is a *growth and differentiation of relations*, such that by the end of our account there are multiple interested stakeholders with distinct but mutually interdependent lines of work. To become infrastructural the pipeline must serve many different interests and become ready-to-hand for a variety of different groups engaged in different problems. Moreover, it must facilitate repurposing to as of yet unknown use cases. Through interactions with new stakeholders, distinctions emerge between "scientific" and "technical" work, between "users" and "developers" (both of whom are biologists), and between the work of the pipeline project and a "community" of downstream data users. The articulation of these delegations and their respective work was central to the process of making the SeaFlow pipeline and its data products repurposable. They were productive in that they aided the SeaFlow project members in managing an increasing amount and variety of work that needed to be accomplished, but they are also ways of reconciling tensions that emerge as the pipeline attempts to traverse increasingly varied sets of practices.

We point to the use of terms here (e.g., "users" and "developers") because they help illustrate the distinction, but these articulations were differences in the kinds of work members of the project took on, what aspects of the project were visible to them, and the relationship their work had with the work of others. Similar to Strauss'[68] discussion of the segmentation of subworlds, difference and connection is produced through the creation of junctures, interfaces, and intersections. These articulations were materially instantiated, involving changes to, with, and around the code of the pipeline. Different lines of work were mediated through GitHub, through the creation of datasets with specific formats, or between different technologies or programming languages. Many of these dynamics are familiar to the field of CSCW. Hoeppe [25] studied the way that datasets become instructing artifacts — although in our case it is through both changes to data fields as well as the construction of metadata elements. Similar to the astronomical catalogue Hoeppe studied, SeaFlow data becomes "hard" when it comes to enable diverse uses coherently and was also made accountable to (different) research objects. As in Pollock, Williams, and D'Adderio's [52] account of generification of software systems: Roger and others on project engage in the structuring of stakeholder groups, such as modelers or biologists focused on particular research areas, and the construction of communities and their generic needs.

One observation from these findings is simply that becoming infrastructural was not just a characteristic of the codebase, but of the organization of work within the SeaFlow project and between the project members and a variety of new stakeholders. Moreover, the research pipeline is a resource that is continually in the making. Pipelines and workflows are often taken in terms of their "runtime" chronology, as a sequence of processing steps through which a particular dataset is transformed. This is certainly an important dynamic in the development of a pipeline [57], but as a number of scholars have pointed out, thinking of pipelines and workflows in these rigidly linear terms can be limiting [7, 71]. Understanding the continual change of these sequences — for instance, how the addition of later steps lead to the alteration of earlier ones — requires examination of the negotiations between stakeholders, and the continual reorganization of the work of developing the

pipeline. This follows the chronology of the pipeline as it develops as a project, through continual redesign and iteration. In the work of pipeline development coordination, articulation work and articulation process are not puzzles to be "solved" and put away. There is continuity of challenges and opportunities as new people join, funding channels become available, the pipeline is leveraged against new research questions, and new kinds of data and metadata are made available. These opportunities and challenges not only have consequences for the code itself, but also for the organization of work on the project.

5.2 Scientific and Technical Entanglements in Research Software

Our findings also force us to consider the pipeline project, and in this case the laboratory, as a site where subworlds [68] come together or are created rather than just as a place for the hyper-specialization of a single line of work: that of a specific subfield or subdiscipline. By the end of our data collection for this paper, the SeaFlow project had brought together researchers focused on oceanographic research questions, contributors focused on the better development of research software, research scientists overseeing data collection and quality assessment, and those building the hardware of a cytometer. This particular project and lab are somewhat atypical in this regard: the SeaFlow Project is an instrument-building project in a large and comparatively well-funded research lab, which has the means to bring diverse skill sets to bear on research problems. However, the instrument-building, interdisciplinary, and outreach-oriented nature of this site make it a good case for examining dynamics that may be unfolding in other sites in similar ways [78].

Following the organizational shifts in a site such as this can help us characterize a particular difficulty of the problem of scientific software. Research software emerges in one consideration as a matter of software engineering, beholden to a set of software development best practices, while in other considerations its only criteria for quality are the scientific problems of a grad student or researcher. The tension between these different priorities is a persistent theme in discussions around scientific software, including in scientists' mixed reactions to the injunction to learn software engineering practices [77], as well as in the tangled and sometimes disjoint motivations for software work and scientific work [27]. Paine and Lee [47] observed a similar dynamic, in which work might be regarded as extra "janitorial" work, but is nevertheless deeply integrated with and constructive of scientific knowledge.

Given this entanglement, it is increasingly clear that the project of building repurposable tools in the sciences will not be limited to the adoption of a best practices by researchers, but will have implications for the kinds of people present in academic research labs, the interrelations between their different skill sets, and the kinds of tasks that constitute scientific work versus other kinds of work. This is apparent in the growing body of work on research software engineering and cyberinfrastructure personnel [4]. In our case it is most visible in Roger and Brandon's efforts to distinguish technical and scientific problems, an endeavor that is sometimes productive and sometimes impossible. Part of CSCW's contribution to understanding research software development is to examine these distinctions and interrelations between tasks and kinds of work as the object of coordinative work. The domains of "scientific" and "technical" are not ready-made, as in Latour's [36] characterization of ready-made science, but rather are worked out through coordinative practice. Such distinctions are both outcomes and resources for the organization of scientific work. Moreover, developing methods for bringing together the skills, tasks, priorities, workflows, values, and practices–the collaborative milieu–around software pipelines will be important for the progress in collaborative science.

6 CONCLUSION

Software pipelines, as embedded in work practices, are core information infrastructure. In knowledge building projects, such as in our field site where new methods and knowledge are being continually created, the challenge is great because the targets are always moving. In this paper we have explored various distinctions made about the work of software development that come to bear on how work is divided across academic research labs. More targeted investigations are needed into how software pipelines and the organization of work in research teams co-evolve. Designing, developing, maintaining, and sustaining infrastructural systems will require closer attention to these kinds of sociotechnical concerns, and it is likely that more long-term studies, such as this one, will be useful for finding answers. The ocean scientists in this study span a wide gamut of disciplines and consequently grapple with diverse instruments, datasets, metadata, methods, research interests, development interests, levels of job security, and reward structures. If we understand infrastructure development as necessitating engagement from diverse stakeholders and communities that come and go and participate in very different ways [19, 48], how can we best support those engagements over time? With the proliferation of information infrastructure and software platforms, studying pipelines as emergent, sociotechnical phenomena will be of increasing importance for users, researchers, developers and also for funders and national infrastructure policy makers. In practice, these pipelines are sociotechnical systems and CSCW technologies that are meant to support communities with diverse interests and ways of doing things.

6.1 Limitations and Future Work

Due to limitations of scope, we were not able to pay close attention to the design and collection of the SeaFlow data products. There have for a long time been calls to situate data products within the context of particular working arrangements[71, 72, 76], and we feel that the pipeline as a research object has the potential to bridge research on scientific software and research on the exchange and design of data. Following the longitudinal development of a pipeline allows for the examination of the emergence and evolution of data types in relation to the development of the instrument that produces them — how data types emerge and take shape as well as how new uses of data prompt reworking of the instruments and software that produce them. Moreover, it allows us to examine how problems of provenance and "data shadows" [39] emerge as the set of work processes surrounding the pipeline become more complex and actors run into problems with a lack of visibility into each other's work. This potential exists as long as we approach the pipeline not as a linear set of software components (as the pipeline exists at one particular "runtime"). Instead, we must collectively take into account the process of developing those components as well as the practices for using them over time.

7 ACKNOWLEDGEMENTS

This article is based on work supported by National Science Foundation grants (#1954620 and #1302272). We thank our anonymous reviewers as well as Drew Paine at Google and Michael Beach and Negin Alimohammadi at the University of Washington for their feedback. Finally, we are grateful to the members of the SeaFlow team and Simons Foundation oceanography research initiatives (CBIOMES, SCOPE, and SCOPE-Gradients) for their participation in our study and their feedback in our findings.

REFERENCES

[1] Joanna Abraham and Madhu C Reddy. 2013. Re-coordinating activities: an investigation of articulation work in patient transfers. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 67–78.

[2] Liam Bannon and Susanne Bødker. 1997. Constructing common information spaces. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*. Springer, 81–96.

[3] Rob Baxter, N Chue Hong, Dirk Gorissen, James Hetherington, and Ilian Todorov. 2012. The research software engineer. In *Digital Research Conference, Oxford*. 1–3.

[4] Nicholas Berente, James Howison, Joel Cutcher-Gershenfeld, John L King, Stephen R Barley, and John Towns. 2017. Professionalization in cyberinfrastructure. *Available at SSRN 3138592* (2017).

[5] Olav W Bertelsen and Susanne Bødker. 2001. Cooperation in massively distributed information spaces. In *ECSCW 2001*. Springer, 1–17.

[6] Alexander Boden, Bernhard Nett, and Volker Wulf. 2008. Articulation work in small-scale offshore software development projects. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*. 21–24.

[7] Christine L Borgman and Morgan F Wofford. 2021. From Data Processes to Data Products: Knowledge Infrastructures in Astronomy. *arXiv preprint arXiv:2109.01707* (2021).

[8] The Carpentries. 2021. The Carpentries. https://carpentries.org/

[9] Jeffrey C Carver, Ian A Cosden, Chris Hill, Sandra Gesing, and Daniel S Katz. 2021. Sustaining Research Software via Research Software Engineers and Professional Associations. *arXiv preprint arXiv:2103.01880* (2021).

[10] Jeffrey C Carver, Sandra Gesing, Daniel S Katz, Karthik Ram, and Nicholas Weber. 2018. Conceptualization of a us research software sustainability institute (URSSI). *Computing in Science & Engineering* 20, 3 (2018), 4–9.

[11] Jeremy Cohen, Daniel S Katz, Michelle Barker, Neil Chue Hong, Robert Haines, and Caroline Jay. 2020. The four pillars of research software engineering. *IEEE Software* 38, 1 (2020), 97–105.

[12] Marisa L Cohn. 2016. Convivial decay: Entangled lifetimes in a geriatric infrastructure. *In Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing* 4, CSCW3 (2016), 1511–1523.

[13] Johanna Cohoon and James Howison. 2021. Norms and Open Systems in Open Science. *Information & Culture* 56, 2 (2021), 115–137.

[14] Juliet Corbin and Anselm Strauss. 2008. Strategies for qualitative data analysis. *Basics of Qualitative Research. Techniques and procedures for developing grounded theory* 3 (2008).

[15] Andy Crabtree, Jacki O'Neill, Peter Tolmie, Stefania Castellani, Tommaso Colombino, and Antonietta Grasso. 2006. The practical indispensability of articulation work to immediate and remote help-giving. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 219–228.

[16] Louise Færgemann, Teresa Schilder-Knudsen, and Peter H Carstensen. 2005. The duality of articulation work in large heterogeneous settings—a study in health care. In *ECSCW 2005*. Springer, 163–183.

[17] Melanie Feinberg, Will Sutherland, Sarah Beth Nelson, Mohammad Hossein Jarrahi, and Arcot Rajasekar. 2020. The New Reality of Reproducibility: The Role of Data Work in Scientific Research. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–22.

[18] Christopher B Field, Michael J Behrenfeld, James T Randerson, and Paul Falkowski. 1998. Primary production of the biosphere: integrating terrestrial and oceanic components. *science* 281, 5374 (1998), 237–240.

[19] Thomas F Gieryn. 1995. Boundaries of science. In *Science and the Quest for Reality*. Springer, 293–332.

[20] Rosalba Giuffrida and Yvonne Dittrich. 2015. A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams. *Information and Software Technology* 63 (2015), 11–30.

[21] Carole Goble. 2014. Better software, better research. *IEEE Internet Computing* 18, 5 (2014), 4–8.

[22] Carole A Goble, Jiten Bhagat, Sergejs Aleksejevs, Don Cruickshank, Danius Michaelides, David Newman, Mark Borkum, Sean Bechhofer, Marco Roos, Peter Li, et al. 2010. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic acids research* 38, suppl_2 (2010), W677–W682.

[23] Suzanne Grant, Jessica Mesman, and Bruce Guthrie. 2016. Spatio-temporal elements of articulation work in the achievement of repeat prescribing safety in UK general practice. *Sociology of health & Illness* 38, 2 (2016), 306–324.

[24] Michael A Heroux, Lois McInnes, David Bernholdt, Anshu Dubey, Elsa Gonsiorowski, Osni Marques, J David Moulton, Boyana Norris, Elaine Raybourn, Satish Balay, et al. 2020. *Advancing Scientific Productivity through Better Scientific Software: Developer Productivity and Software Sustainability Report.* Technical Report. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

[25] Götz Hoeppe. 2021. Encoding Collective Knowledge, Instructing Data Reusers: The Collaborative Fixation of a Digital Scientific Data Set. *Computer Supported Cooperative Work (CSCW)* 30, 4 (2021), 463–505.

[26] Neil Chue Hong. 2014. Minimal information for reusable scientific software. In *Proceedings of the 2nd Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2. 1)*.

[27] James Howison and James D Herbsleb. 2011. Scientific software production: incentives and collaboration. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. 513–522.

[28] James Howison and James D Herbsleb. 2013. Incentives and integration in scientific software production. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 459–470.

[29] Sarah Inman and David Ribes. 2018. Data Streams, Data Seams: Toward a seamful representationof data interoperability. (2018).

[30] Helena Karasti. 2014. Infrastructuring in participatory design. In *Proceedings of the 13th Participatory Design Conference: Research Papers-Volume 1*. 141–150.

[31] Helena Karasti and Karen S Baker. 2008. Community Design: growing one's own information infrastructure. In *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*. 217–220.

[32] Helena Karasti, Karen S Baker, and Florence Millerand. 2010. Infrastructure time: Long-term matters in collaborative development. *Computer Supported Cooperative Work (CSCW)* 19, CSCW2 (2010), 377–415.

[33] Helena Karasti and Anna-Liisa Syrjänen. 2004. Artful infrastructuring in two cases of community PD. In *Proceedings of the eighth conference on Participatory design: Artful integration: interweaving media, materials and practices-Volume 1*. 20–30.

[34] Peter Keating and Alberto Cambrosio. 1998. *Interlaboratory life: Regulating flow cytometry*. Palgrave Macmillan.

[35] Amy J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. 2011. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)* 43, 3 (2011), 1–44.

[36] Bruno Latour. 1987. *Science in action: How to follow scientists and engineers through society*. Harvard university press.

[37] Charlotte P Lee, Paul Dourish, and Gloria Mark. 2006. The human infrastructure of cyberinfrastructure. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 483–492.

[38] Charlotte P Lee and Drew Paine. 2015. From The matrix to a model of coordinated action (MoCA) A conceptual framework of and for CSCW. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. 179–194.

[39] Sabina Leonelli, Brian Rappert, and Gail Davies. 2017. Data shadows: Knowledge, openness, and absence. , 191–202 pages.

[40] James J Marshall, Robert R Downs, and Shahin Samadi. 2010. Relevance of software reuse in building advanced scientific data processing systems. *Earth Science Informatics* 3, 1 (2010), 95–100.

[41] Stina Matthiesen, Pernille Bjørn, and Lise Møller Petersen. 2014. " Figure out how to code with the hands of others" recognizing cultural blind spots in global software development. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 1107–1119.

[42] Reed Milewicz and Elaine Raybourn. 2018. Talk to me: A case study on coordinating expertise in large-scale scientific software projects. In *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, 9–18.

[43] Naja Holten Møller, Claus Bossen, Kathleen H Pine, Trine Rask Nielsen, and Gina Neff. 2020. Who does the work of data? *Interactions* 27, 3 (2020), 52–55.

[44] MOLSSI. 2021. The molecular science software institute (MolSSi). https://molssi.org/

[45] Bonnie A Nardi, Steve Whittaker, and Heinrich Schwarz. 2002. NetWORKers and their activity in intensional networks. *Computer Supported Cooperative Work (CSCW)* 11, 1 (2002), 205–242.

[46] NASA. 2021. NASA Open Source Development. https://www.nasa.gov/open/open-source-development.html

[47] Drew Paine and Charlotte P Lee. 2014. Producing data, producing software: Developing a radio astronomy research infrastructure. In *2014 IEEE 10th International Conference on e-Science*, Vol. 1. IEEE, 231–238.

[48] Carole L Palmer. 2013. *Work at the boundaries of science: Information and the interdisciplinary research process*. Springer Science & Business Media.

[49] Elena Parmiggiani and Helena Karasti. 2018. Surfacing the arctic: politics of participation in infrastructuring. In *Proceedings of the 15th Participatory Design Conference: Short Papers, Situated Actions, Workshops and Tutorial-Volume 2*. 1–5.

[50] Elena Parmiggiani, Eric Monteiro, and Vidar Hepsø. 2015. The digital coral: Infrastructuring environmental monitoring. *Computer Supported Cooperative Work (CSCW)* 24, 5 (2015), 423–460.

[51] Kathleen Pine, Claus Bossen, Naja Holten Møller, Milagros Miceli, Alex Jiahong Lu, Yunan Chen, Leah Horgan, Zhaoyuan Su, Gina Neff, and Melissa Mazmanian. 2022. Investigating Data Work Across Domains: New Perspectives on the Work of Creating Data. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–6.

[52] Neil Pollock, Robin Williams, and Luciana D'Adderio. 2007. Global software and its provenance: generification work in the production of organizational software packages. *Social studies of science* 37, 2 (2007), 254–280.

[53] Karthik Ram, Carl Boettiger, Scott Chamberlain, Noam Ross, Maëlle Salmon, and Stefanie Butland. 2018. A community of practice around peer review for long-term research software sustainability. *Computing in Science & Engineering* 21, 2 (2018), 59–65.

[54] David Ribes. 2014. The kernel of a research infrastructure. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 574–587.

[55] David Ribes, Karen S Baker, Florence Millerand, and Geoffrey C Bowker. 2005. Comparative interoperability project: Configurations of community, technology, organization. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*. 65–66.
[56] David Ribes, Andrew S Hoffman, Steven C Slota, and Geoffrey C Bowker. 2019. The logic of domains. *Social studies of science* 49, 3 (2019), 281–309.
[57] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
[58] Robert K Sawyer. 2005. *Social emergence: Societies as complex systems*. Cambridge University Press.
[59] Judith Segal. 2007. Some problems of professional end user developers. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*. IEEE, 111–118.
[60] Judith Segal. 2008. Scientists and software engineers: A tale of two cultures. (2008).
[61] Judith Segal. 2009. Software development cultures and cooperation problems: A field study of the early stages of development of software for a scientific community. *Computer Supported Cooperative Work (CSCW)* 18, 5 (2009), 581–606.
[62] Benjamin Hayden Sims. 2022. *Research software engineering: Professionalization, roles, and identity*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
[63] Robert Soden, David Ribes, Seyram Avie, and Will Sutherland. 2021. Time for historicism in CSCW: An invitation. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–18.
[64] Heidi M Sosik, Robert J Olson, and E Armbrust. 2010. Flow cytometry in phytoplankton research. In *Chlorophyll a Fluorescence in aquatic sciences: methods and applications*. Springer, 171–185.
[65] SSI. 2021. Software sustainability institute (SSI). https://www.software.ac.uk/
[66] Susan Leigh Star and Karen Ruhleder. 1996. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information systems research* 7, 1 (1996), 111–134.
[67] Victoria Stodden, Matthew S Krafczyk, and Adhithya Bhaskar. 2018. Enabling the verification of computational results: An empirical evaluation of computational reproducibility. In *Proceedings of the First International Workshop on Practical Reproducible Evaluation of Computer Systems*. 1–5.
[68] Anselm Strauss. 1984. Social worlds and their segmentation processes. *Studies in symbolic interaction* 5, 1 (1984), 123–139.
[69] Anselm Strauss. 1985. Work and the division of labor. *Sociological quarterly* 26, 1 (1985), 1–19.
[70] Anselm Strauss. 1988. The articulation of project work: An organizational process. *Sociological Quarterly* 29 (1988), 163–178. Issue 2.
[71] Andrea K Thomer, Dharma Akmon, Jeremy York, Allison RB Tyler, Faye Polasek, Sara Lafia, Libby Hemphill, and Elizabeth Yakel. 2022. The craft and coordination of data curation: complicating" workflow" views of data science. *arXiv preprint arXiv:2202.04560* (2022).
[72] Andrea K Thomer, Michael Bernard Twidale, and Matthew J Yoder. 2018. Transforming Taxonomic Interfaces: " Arm? s Length" Cooperative Work and the Maintenance of a Long-lived Classification System. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–23.
[73] Erik Trainer, Chalalai Chaihirunkarn, and James Herbsleb. 2013. The Big Effects of Short-term Efforts: Mentorship and Code Integration in Open Source Scientific Software. *Journal of Open Research Software* (2013).
[74] Erik H Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D Herbsleb. 2015. From personal tool to community resource: What's the extra work and who will do it?. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*. 417–430.
[75] URSSI. 2021. US Research Software Sustainability Institute (URSSI). (2021). https://urssi.us/
[76] Janet Vertesi and Paul Dourish. 2011. The value of data: considering the context of production in data economies. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. 533–542.
[77] Gregory V Wilson. 2006. Where's the real bottleneck in scientific computing? *American Scientist* 94, 1 (2006), 5.
[78] Susan J Winter and Nicholas Berente. 2012. A commentary on the pluralistic goals, logics of action, and institutional contexts of translational team science. *Translational Behavioral Medicine* 2, 4 (2012), 441–445.

Received January 2022; revised July 2022; accepted November 2022