Understanding the Needs of Enterprise Users in Collaborative Python Notebooks

This paper examines enterprise user needs in collaborative Python notebooks through a dyadic interview study

Catherine Li
Brown University

Talie Massachi
Brown University

Jordan Eschler
Independent Researcher

Jeff Huang
Brown University
hci@jeffhuang.com

ABSTRACT

Python notebooks are an important productivity tool for technical employees in software companies. The Python notebook format originates from open-source coding projects and scientific research; notebooks were intended to spread knowledge about solving problems and modeling analytic approaches through code. In this case study writeup, we describe a qualitative study of Python notebooks as sites of user collaboration among varied roles (engineers, data scientists, and technical investigators) in a Fortune 500 software enterprise. Findings of the case study build on previous research on collaboration via notebooks, and articulate specific collaborative tasks undertaken by participants, the benefits of these collaborative tasks to the user and the broader enterprise, and design implications of findings around user needs for collaborative workflows. Finally, we reflect on the findings of this study in terms of applying a method specific to the use context of interest, as well as the study's impact on enterprise software strategy.

CCS CONCEPTS

• **Human-centered computing** → Collaborative and social computing; Collaborative and social computing design and evaluation methods; Human computer interaction (HCI); Interaction paradigms; Collaborative interaction.

KEYWORDS

collaborative workflows, enterprise users, literate programming, python notebooks, dyadic study, mentorship

ACM Reference Format:
Catherine Li, Talie Massachi, Jordan Eschler, and Jeff Huang. 2023. Understanding the Needs of Enterprise Users in Collaborative Python Notebooks: This paper examines enterprise user needs in collaborative Python notebooks through a dyadic interview study. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544549.3573843

1 INTRODUCTION

Python notebooks have become an integral part of code sharing and open-source collaboration, science research and education, and key artifacts in code prototyping and data analysis in industry [13–15]. From previous research, we know a great deal about the sharing and reusability of Python notebooks on code repositories, such as GitHub; the experiences of users collaborating via notebooks; and the decay of notebook artifacts as usable documents [5, 6, 12].

In this paper, we examine collaborative use of notebooks in a large software company among a range of roles (engineers, data scientists, and technical investigators). We build on previous work about collaborative notebook labor (e.g., [19]) to describe the role of Python notebooks as valuable artifacts for learning and mentoring, exercising quality control of analyses, and socializing process automation techniques.

Specifically, the third author—an internal researcher at the software company—interviewed 20 users within a software company about their use of Python notebooks (on an internally-built and -maintained Python notebook product, a clone of Jupyter notebooks we will call "Bingo"). Where possible, the internal researcher recruited users in "dyads"—pairs of frequent individuals who work together via shared notebooks—to understand participants' collaborative workflows and the sociotechnical patterns around notebook-based collaboration.

A dyadic interview data collection method was employed due to previous research with Python notebook users in the enterprise (specifically, analysis of product user telemetry data that showed notebook sharing and editing patterns, as well as self-reported collaboration workflows in previous internal user surveys). In light of this prior, internal data, the enterprise Python notebook team requested more insight as to the paired work occurring naturally via notebooks.

In discussing collaborative notebook-based work, our participants shared their nuanced practices around authoring, sharing, and maintaining notebooks to accomplish their learning, productivity, and career goals. We pursued the following research questions in conducting interviews and analyzing the resulting data:

CHI EA '23, April 23–28, 2023, Hamburg, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3573843

- RQ1. How do users currently collaborate using Python notebooks in the enterprise?
- RQ2. What are the benefits to users and the enterprise when collaborating via Python notebooks?
- RQ3. How might we better support collaborative workflows through our notebook product design?

This case study highlights the values of tailoring data collection methods to user contexts in the enterprise (e.g., examining the unit of a user dyad, rather than single users), and demonstrates the impact a rigorous, qualitative study can have on enterprise technology product strategy and ongoing engineering investment.

2 PYTHON NOTEBOOKS, INSIDE AND OUTSIDE INDUSTRY

Python notebooks are a widely used tool in research and industry settings. In this section, we review the origins of notebooks as computational narrative artifacts, and describe prior research on notebooks as sites of collaboration.

2.1 Python Notebooks and Computational Narratives

Wolfram's Mathematica notebooks were one of the original notebook products that allowed users to combine code, human-readable markdown text, and data visualization in one artifact; these powerful notebooks were the foundation of the subsequent open-source Jupyter Notebook tool. The original objective of Jupyter Notebooks was to support *"the collaborative creation of reproducible computational narratives that can be used across a wide range of audiences and contexts"* [14]. In the years since Jupyter's launch, Python notebooks have become the tool of choice for Data Scientists and computational science endeavors, and are becoming a standard in data science education [15].

Despite the potential value of a computational narrative in understanding, analyzing, and drawing conclusions from data, Python notebooks have notable shortcomings in terms of workflow and coordination. For example, Python notebooks fall short in terms of source control [5], and notebooks facilitate a certain "messiness" due to emphasis on data exploration in the notebook [17], previously described by Head et al. [6] and Kery et al. [8]. Each of these problems, or any of them in combination, can impact a Data Scientist's ability to tell a clear computational narrative.

In addition, data science practitioners—whether by official job title or practical function—approach data science and its principles from various backgrounds. In both educational and professional settings, the disparity of skills and tool familiarity among data science practices can be quite broad [11, 15]. As a result, some data scientists come into enterprise data work with an extensive coding background, while other data scientists may know certain proprietary tools (e.g., Excel or SPSS), but not Python or R code.

Specific pain points in terms of collaboration via notebooks from previous research are described below.

2.2 Python Notebooks as Collaborative Work Spaces

Previous literature has identified pain points related to sharing and collaborating in Python notebooks [2, 19]. Perhaps the most challenging aspect for collaborating via Python notebook products involves the coordination of work among collaborators, whose strengths can vary among domain expertise, programming, and communication [18]. Specifically, previous research has found that versioning of code for data exploration often occurs informally, through commenting out code and preserving previous iterations of code to retain some history—and therefore, accumulated learning about—data exploration and understanding work [7, 8]. These informal versioning practices can lead to complications in maintaining a source of truth, or a clear data exploration process history, among multiple collaborators [19].

In support of the above point, Pimentel and co-authors [16] identify a number of barriers to reproducibility that are exacerbated by notebook conventions, such as executing cells in non-linear order, or lack of accounting for relative filepaths to allow re-executing cells with data dependencies. In sum, all of these barriers can reduce the value and coherency of notebook collaborations.

Such collaboration woes are being addressed in the notebook product space, including new collaboration-first notebook products such as Colab[1] and DeepNote[2]. These new collaborative notebook products attempt to solve issues with coordinating and communicating during shared work (e.g., as described in [1]) through supporting "socially translucent systems," wherein users gain visibility, awareness, and accountability [4] via the notebook interface.

2.3 Python Notebooks as 'Literate Programming'

An important note we make here: notebooks have been previously described in literature (and in this paper, above) as "computational narratives"—that is, arguments consisting of code, text, and outputs to converge on answers to a problem, or an approach to data understanding. More recently, however, notebooks are described as an instance of literate programming (e.g., [9]). Literate programming is defined as *programming artifacts that are meant not only to instruct a computer, but to inform fellow humans about instructions for a computer* [10].

Early assessments of literate programming viewed the paradigm of literacy as one that would benefit, primarily, systems implementations in large-scale computer programming efforts (e.g., [3]). In this case study, however, we will discuss the practice of literate programming in terms of the collaborations participants described at the notebook level; that is, coders work in tandem to author code in the notebook, but they also communicate in human language with each other via the notebook artifact itself.

3 METHODS

3.1 Research Setting

The data collection described below was conducted with employees at a Fortune 500 software company. Each of the employees

[1] https://colab.research.google.com/
[2] https://deepnote.com/

Table 1: Overview of dyads, participants, and nature of collaborative relationship*

Dyad#	Individual 1 role	Individual 2 role	Dyad collaboration(s)*
D01	Technical Investigator	Technical Investigator	Individual 1 mentored 2 (e.g. helped them learn Python)
D02	Data Scientist	Technical Investigator	Individual 1 mentored 2 (e.g. helped them learn Python)
D03	Technical Investigator	Data Scientist	Individual 2 mentored Individual 1 (semi-technical)
D04	Technical Investigator	Technical Investigator	Collaborative, equally technical (e.g. worked together)
D06	Data Engineer	Data Scientist	Individual 1 mentored 2 (e.g. provided feedback)
D09	ML Engineer	Software Engineer	Non-mentorship, equally technical (e.g. worked together)
D10	Production Engineer	Production Engineer	Non-mentorship, equally technical (e.g. worked together)
D12	ML Engineer	Software Engineer	Non-mentorship, equally technical (e.g. worked together)

* *Participants in planned dyads 5, 7, 8 (Data Scientists) and 11 (Software Engineer) did not have two participants; the second potential participant declined to engage with the study.*

who participated in this study used the internal Python notebooks product, which we will refer to as Bingo in this paper. Bingo is a notebook product that was adapted from the open-source Jupyter platform—and, at the time of data collection, cloned both the function and form of the open-source Jupyter tool. Bingo is built and maintained internally, as a secure, proprietary notebook platform for employees working on software and analytic projects in the enterprise setting.

The Bingo product is included in the standard secure data analysis product suite used by technical employees—e.g., employees who are capable of coding in Python, R, and/or query languages—in all areas of the enterprise and in its sub-organizations, such as data science, data engineering, software and machine learning model development, hardware and production engineering, and product-related metrics tracking. Bingo averages 11,000 to 13,000 unique users per month, and its similarity to the open-source Jupyter user interface facilitates rapid uptake and training for previous notebook users.

Case study data was provided to the academic authors per a data agreement with the enterprise. The third author, an internal product researcher who had ongoing access to the notebook product users, conducted semi-structured interviews and consented participants into data collection and audio recording. Finally, the research protocol was approved in accordance with the company's internal privacy and ethics review process for research involving employee participants.

3.2 Data Gathering and Analysis

The internal product researcher conducted individual, one-hour interviews with each of 20 users of the Python notebooks product within the social media company. These 20 users comprised eight complete frequent collaborator dyads and four individual users. In the case of individual participants without a complete dyad, the third author was unable to recruit the work peer to complete the dyad; in all four of these cases, the work peer demurred due to "newness to their role" and/or "lack of technical knowledge." We nevertheless retained the 4 single users' transcripts for this study due to their insights regarding knowledge production and learning from their own points of view (see Table 1).

Interviews were conducted at the beginning of office closures for the COVID-19 pandemic, during March and April 2020. Due to office closures, all interviews took place via secure video calls on an internal meeting platform, and were audio recorded by the researcher. Recordings were transcribed, de-identified, and the audio files were subsequently deleted, in compliance with internal company research policies. Participants were compensated $25 in e-gift rewards.

Once interview transcripts had been finalized, two authors and the internal product researcher employed iterative open coding cycles to create an axial codebook [9]. First, three overlapping transcripts were coded by researchers, who then discussed the open codes to consolidate an initial codebook based on a set of grouped themes. The codebook was further refined by broadening coding to 6 transcripts, after which the researchers again iterated on the codebook. In all, 4 revisions of the codebook were drafted, with the final codebook yielding themes to answer RQ1 and RQ2. We present the findings in the following section.

3.3 Participant Roles

- Technical Investigators: Technical Investigators research patterns of user engagement with the enterprise company's suite of products. Technical investigator participants used notebooks as part of their toolkit to analyze related data.
- Data Scientists: Data Scientists work in all pillars of the company, running data analytics and metrics assessments to support product goals. Notebooks are an important platform for producing new data analyses and visualization artifacts in their workflow.
- Data Engineers: Data Engineers build and maintain data pipelines to support data science work, and are experts in data provenance and quality. Notebooks are used in Data Engineering to prototype and troubleshoot data pipelines.
- Engineers: Production, Software, and Machine Learning Engineers use notebooks to prototype production code updates, plan for hardware capacity, and to investigate machine learning model iterations.

When we refer to "notebook users" in this study, we describe individuals who directly interact with notebooks, including those who authored, adapted, and re-used notebooks. In contrast, "stakeholders" describe people whose role and workflow depends upon notebook-based production, (e.g., findings from analyses or performance of machine learning models) but do not author or adapt notebooks and may not ever view notebook code directly.

4 RESULTS

Participants described three primary use cases for collaborating via notebooks, in response to RQ1 ("*How do users currently collaborate using Python notebooks in the enterprise*?"): mentoring and learning; quality control; and process automation. For each of these use cases, participants also articulated clear benefits to their workflows and, ultimately, to their stakeholders (in response to RQ2, "*What are the benefits to users, and the enterprise, when collaborating via Python notebooks*?"). We elaborate on each of the use cases below.

4.1 Mentoring and Learning

We found that mentoring and learning via notebooks occurred informally; often, participants described collaborating via the Python notebook product specifically for its ability to mix code and markdown (or human-readable language, such as instructions for using the notebook).

4.1.1 Mentoring. In an engineering-forward setting like this enterprise, less-technical notebook users felt a great deal of pressure to ramp up their technical expertise quickly:

> *There's so much more that I could be doing every day with this data. And it frustrates me that I don't have those skills and I don't feel like I have the time to invest in building those skills. . . but it's just not been a priority yet or it's been pushed to the back burner. So I feel more shame about the fact that we're not using it more often.* – D02.2, Technical Investigator

In response to this pressure, we found that participants had formed informal mentor/mentee relationships via notebook collaboration tasks. With these activities, less-technical peers felt they were more likely to build their analytical skills and embrace the efficiency increase notebooks offered in completing data-driven tasks. In most cases, Python was not a required skill for work for these less technical peers. A large part of the mentoring experience was open communication and feedback between the more- and less-technical collaborators, facilitated by the easy back and forth provided by notebooks:

> *If you're just getting started, it's really scary to just open [a notebook], get on Bingo, and then where do I start? But if I can like, "Hey, let's sit together, work on [project] together and you can shadow me as I write the code, import libraries."...It's so much easier instead of just screen-sharing. I think it's a great opportunity. Less isolating, I think.* – D01.2, Technical Investigator

4.1.2 Learning. Although Technical Investigators utilized active mentoring, we found that participants in other, more technical, roles largely relied on individual learning tactics. In these cases, notebooks are used as a central demonstration space where users can see code applied to various data problems and learn new code libraries or problem-solving techniques.

Participants also described learning via notebooks published to the enterprise code repository, which were searchable and discoverable. This "encountering" technique was especially useful for engineers and Data Scientists who already had confidence using Jupyter (outside of the enterprise setting). Participants were able to find useful code in published notebooks, notebook content gave them information to practice new analysis techniques, borrow code snippets, and reference already-published solutions. D02.1 described looking up a fellow user's published notebooks after admiring one of their analysis write-ups:

> *I'm like, "Huh, this person's work seems really interesting. I wonder what else they're into." And can be like, "Notebooks belonging to Bob." And like, "wow, that's super interesting. He's using X and X and I would've never thought about that."* – D02.1, Data Scientist

Unlike the relatively one-sided form of collaboration that we see in mentoring, where one participant teaches and the other learns, published notebooks act as a way to facilitate education for both the publisher and the reader. When participants published (or shared) notebooks, they were also looking to learn. Often, users looked forward to feedback about the quality of their code or input about new ways to solve data problems.

> *So, I usually share [notebooks] for two things. One is to tell people what [a notebook] can do, and what they can use it for. But, at the same time, I would definitely look forward to their feedback as well. If there are things that they feel that should be added to it, which probably I might have missed out on earlier, right?* – D06.2, Data Scientist

This two-way learning occurred when users published their work to the enterprise code repository, or wrote up their notebook findings in an internal blog post.

4.1.3 Barriers to mentoring and learning. The most persistent barrier to mentoring proved to be outside the scope of the internal product researcher's purview; specifically, mentoring was an unrecognized activity that was not officially supported by management or leadership. However, in terms of encountering valuable notebooks for learning on the job, multiple participants pointed to the difficulty in discovering useful notebooks:

> *Whenever I've searched [published notebooks in the repository], I don't think I've ever found anything useful using the notebook search tool. . . relevance, I don't think, is calculated in any way.* – D06.1, Data Engineer
>
> *That's the biggest problem, when I go [to look at published notebooks], I don't know how to differentiate between valuable stuff that other people published intentionally for me to use, valuable stuff that people published intentionally for me to use that's still up to date, and things that were for debugging that people shared with somebody else to debug that never were intended for consumption beyond the individual.* – D01.1, Technical Investigator

Notebook creators also lamented not knowing if their work was showing up in other users' searches, or how many other users had viewed or cloned their notebooks. Without knowing the relevance and popularity of their notebooks, users were unable to share their notebooks to the right audiences within the enterprise who might benefit from their published work.

> *People definitely read my notebook. But to be honest, I don't know how many of our notebook users we have in a company as Data Scientists, right? So I don't think engineers would clone my work because we do different work, but most likely would be a data person who clones my work.* – D08.1, Data Scientist

4.1.4 Benefits to users and the enterprise. Finally, we found that participants engaging in mentoring and learning use cases benefited both users and enterprise productivity. Specifically, mentoring and learning via notebooks effectively improved coding and analytic skills for less-technical team members, and facilitated discovery of shared problem-solving techniques to improve productivity and enhance insights through data.

Sometimes, in contrast to mentoring and learning, users paired up via notebooks to ensure quality control throughout a project or coding process, as we detail in the next section.

4.2 Quality Control

Participants described processes of "collaborative coding" or authoring a notebook artifact in tandem with technical peer collaborators to address a shared problem or task. In these cases, the work in notebooks was often used to either present different viewpoints and information to collaborators with different interests or levels of technical knowledge, or to iterate on a shared problem in order to arrive at a shared understanding for a solution.

Specifically, notebooks acted as a bridging space where peers with different areas of expertise were able to collaborate and contribute based on their strengths. For example, in Dyad 02, we see the combination of a more- and less-technical peer; however, the less technical peer (D02.2) brought valuable subject matter expertise to their joint investigations:

> *My background is unconventional for this field. . . I came into this line of work because I know [foreign languages], but I had none of the technical skills required to do this work, so it's all been a learning curve. And having someone like [D02.1] was absolutely critical to me being able to do this work in a new sort of way because her insight helps us take this subject matter expertise and turn it into something technical that other people can use.* – D02.2, Technical Investigator

Notebooks were also a communication medium with stakeholders, although the level of preparation to publish notebooks to the enterprise repository varied by the type of stakeholders a notebook might serve. For example, D10.2, a Production engineer, did not do much to polish up their notebooks prior to publishing; stakeholders for their analyses were often very technical. In contrast, when notebook stakeholders were less technical, more preparation was required to convey quality to the notebook audience:

> *It depends...If I'm sending something to a PM who's a director and who has less [time] that he can spend, looking at something and trying to understand, I'll make sure I spell everything out in the plots and all that. If it's my partner who is also working pretty closely with whatever I'm doing. . . they've already been hearing about what I'm working on. They know what to look for.* – D06.1, Data Engineer

4.2.1 Barriers to quality control. Limitations of the Bingo notebook product led to source control and versioning issues that prevented easy work coordination; for example, D11.1 (Software engineer) noted that one notebook-based collaboration devolved into confusion when, after several iterations, "*it turns out we used different versions of datasets*" during the course of a complex analysis. Because of these issues, participants were enthusiastic about improving features in the notebook product for coordinating work, including version and source control, to more effectively execute on quality control collaborations.

4.2.2 Benefits to users and the enterprise. Benefits of quality control use cases among notebook users included increased confidence in the quality of analytic and investigative notebook write-ups, as well as facilitated a healthy feedback culture to add to the collective knowledge of notebook users.

In the next section, we discuss the third and final use case for collaboration via notebooks: process automation.

4.3 Process Automation

Within the enterprise, and among the Python notebooks product team members, a particularly surprising result of these interviews was discovering a growing practice of process automation taking place in internal Python notebooks.

In general, process automation efforts took place when a cluster of users (in the same team, or role, or both) routinely faced similar data problems. One example of such process automation occurred in Dyad 04, both Technical Investigators. Investigators' day to day work involved a range of standard procedures to inform investigations, resulting in a variety of routine work tasks that were ripe for automation in notebooks. Dyad 04 described decisions to create a comprehensive notebook to operationalize data analysis related to technical investigations:

> *We decided that we were going to automate [a formatting and data-cleaning] process. . . We rebuilt the workflows, most it was offloaded because sometimes it produced a lot of data. And could be cleaned and then by the time it got to Bingo, it could primarily just be pulled and have minimal processing so that it was fast to run.* – D04.1, Technical Investigator

> *Generally speaking, when a teammate would get a particular type of indicator, they might go through steps A, B, and C in order to answer the question of whether it's valuable or pertinent to our team's mission. But one user might go through steps A, B, and F. Another person might just do C—There was no real consistency in terms of what was being done. . . the team rallied around the idea that consistency was valuable.* – D04.2, Technical Investigator

Table 2: Aligning use cases and barriers with design implications and long-term impact to the product roadmap

Use case and barrier	Design implication(s) to address barrier	Impact to product roadmap
Mentoring and learning: Notebook discovery	Improve search algorithm to find notebooks	[Long-term] Added enhanced search feature to engineering roadmap [Medium-term] Investigate machine learning model to cluster notebooks by code similarity (proxy for problem similarity)
Mentoring and learning: Socializing Notebooks	Give notebook creators a dashboard to understand engagement with their notebooks	[Short-term] Launch notebook user dashboard within 6 months
Quality control: Version and source control	Improve user interface for version and source control	[Short-term] Utilize co-design to revamp versioning user controls and to iterate on version and source control system feedback
Process automation: Version and source control		[Medium-term] Launch new user interface to clarify version and source control feedback

The work of process automation took place in notebooks, participants noted, because they could use markdown to include instructions, describe code inputs and outputs, or parameterize variables for analysis. Participants described how they cleaned up code, commented to explain logic and methods, reorganized cells, and provided necessary project context:

> *when you’re trying to share a notebook, it would have to be cleaned up, right?...[the notebook] needs to have certain comments within it which could explain what you’re doing in a certain code frame. So, I would add comments to it, and then I would separate certain code frames depending on those comments as well, so that I would make it easier to understand, and cleaner to read for people.* – D06.2, Data Scientist

4.3.1 Barriers to process automation. Although participants were enthusiastic about automating processes via notebooks, they also noted that the current versioning and source control features in Bingo notebooks complicated notebook maintenance.

> *For our team’s main notebook, we do have different sections, but not... there is no way to maintain them. And it sometimes becomes outdated and [the notebook] stops working...If I have time, I will just update that notebook, change the APIs to new ones or something like that.* – D11.1, Software Engineer

4.3.2 Benefits to users and the enterprise. This process automation allowed for future, similar tasks to be completed easily and without the need to rewrite processes, saving time for users and subsequently money for the enterprise.

5 DISCUSSION

In this section, we discuss the practical implications arising from opportunities identified in the course of user interviews described here. Then, we reflect on the value of considering the user context when designing research in the enterprise.

5.1 Practical Implications

Once the enterprise Python notebooks engineering team understood that collaboration was both occurring routinely and viewed as beneficial to both users and enterprise, the team worked with their internal product researcher (who also collected the data internally) to invest in a collaboration-focused engineering workstream on the product roadmap. This process answered RQ3, “*How might we better support collaborative workflows through our notebook product design*?”

Informed by a deep understanding of the entire picture of notebook-based collaboration–thanks to the dyadic interview structure–the product team was able to understand how a range of notebook users (more- and less-technical, inhabiting a variety of roles) experienced collaboration workflows in the existing notebook product.

Specifically, for each of the barriers identified with regard to collaboration use cases, the product team identified features to prioritize to enhance the user experience, and adjusted future design and engineering roadmap items for the product to continue improving the suite of collaboration features in the product long-term (see Table 2).

5.2 Reflection on Case Study

There are two primary takeaways of this case study. First, HCI research in the enterprise can often be limited (by researcher or product manager imagination or skillset, or by organizational constraints or policy) to individual user interviews or usability tests. However, relying on such methods that are limited to understanding a lone user’s experience do not always reflect use context, and therefore may not produce the right level of understanding around a behavioral phenomenon among technology users. Here, using dyadic interviews gave additional detail and helped the product team understand both viewpoints of a notebook-based collaboration; in particular, capturing the collaborations of more- and less-technical user dyads illuminated a hidden, but important, facet of learning technical skills on the job, in the enterprise.

Second, implications of industry-based research do not solely affect product or process, but should also impact future approaches to design, as well as investment in relevant engineering work. For example, in Table 2, we indicate that further design research was needed to investigate user options related to version and source control in the notebook product. These user options were complex; as a result, the internal product researcher and the product designer used co-design to investigate opportunities for revamping the user experience around version and source control. With these co-design sessions, the researcher and designer again engaged with user dyads, to simulate actual practices around notebook collaborations.

6 LIMITATIONS

The limitations of this study are as with any qualitative, empirical work; we are bound by the context of the data collected and make no claims regarding generalization of our findings. Instead, we have endeavored to include as much detail about the subject enterprise and the research context as possible, to provide findings and implications that are abductively useful in comparable settings (specifically, notebook-based knowledge work in software enterprises).

7 CONCLUSION

Python notebooks are an important literate programming tool for users in software enterprise settings to share domain expertise and solve problems in a shared medium. The findings of this case study are informed by and build on previous research on collaboration via notebooks, specifically lending a glimpse to notebook-based collaborative practices in software enterprise settings. We found that users collaborated to benefit themselves and the enterprise by engaging in activities such as mentoring and learning, quality control, and process automation. We argue that the data collection approach of this case study (dyadic interview units to mirror collaborative practices via notebooks), as well as its practical impact on product engineer roadmaps, illustrates the value of HCI research for understanding complex user behavior in the product context.

8 HISTORY DATES

Received October 2022

ACKNOWLEDGMENTS

This work was supported in part by a NASA Rhode Island EPSCoR Seed Grant and a gift from Meta. The authors also wish to thank the infrastructure engineering team at Meta for their input and expertise in conducting the study.

REFERENCES

[1] Bardram, Jakob E. "Activity-based computing for medical work in hospitals." ACM Transactions on Computer-Human Interaction (TOCHI) 16, no. 2 (2009): 1-36.

[2] Chattopadhyay, Souti, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. "What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities." In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2020.

[3] Cordes, David, and Marcus Brown. "The literate-programming paradigm." Computer 24, no. 6 (1991): 52-61.

[4] Erickson, Thomas, and Wendy A. Kellogg. "Social translucence: an approach to designing systems that support social processes." ACM transactions on computer-human interaction (TOCHI) 7, no. 1 (2000): 59-83.

[5] Grus, Joel. I Don't Like Notebooks, JupyterCon 2018, online, accessed 11.11.19, https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/, https://youtu.be/7jiPeIFXb6U

[6] Head, Andrew, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. "Managing messes in computational notebooks." In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2019.

[7] Kery, Mary Beth, Amber Horvath, and Brad A. Myers. "Variolite: Supporting Exploratory Programming by Data Scientists." In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 1265–1276. 2017.

[8] Kery, Mary Beth, and Brad A. Myers. "Interactions for untangling messy history in a computational notebook." In 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 147-155. IEEE, 2018a.

[9] Kery, Mary Beth, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. "The story in the notebook: Exploratory data science using a literate programming tool." In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1-11. 2018.

[10] Knuth, Donald Ervin. "Literate programming." The computer journal 27, no. 2 (1984): 97-111.

[11] Kross, Sean, and Philip J. Guo. "Practitioners teaching data science in industry and academia: Expectations, workflows, and challenges." In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-14. 2019.

[12] Lau, Sam, Ian Drosos, Julia M. Markel, and Philip J. Guo. "The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry." In 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1-11. IEEE, 2020.

[13] Netflix Technology Blog. 2018. Beyond Interactive: Notebook Innovation at Netflix. https://medium.com/netflix-techblog/notebook-innovation-591ee3221233

[14] Perez, Fernando, and Brian E. Granger. "Project Jupyter: Computational narratives as the engine of collaborative data science." Retrieved September 11, no. 207 (2015): 108.

[15] Perkel, Jeffrey M. "Why Jupyter is data scientists' computational notebook of choice." Nature 563, no. 7732 (2018): 145-147.

[16] Pimentel, João Felipe, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. "A large-scale study about quality and reproducibility of jupyter notebooks." In 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), pp. 507-517. IEEE, 2019.

[17] Rule, Adam, Aurélien Tabard, and James D. Hollan. "Exploration and explanation in computational notebooks." In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2018.

[18] Wang, April Yi, Anant Mittal, Christopher Brooks, and Steve Oney. "How data scientists use computational notebooks for real-time collaboration." Proceedings of the ACM on Human-Computer Interaction 3, no. CSCW (2019): 1-30.

[19] Wang, April Yi, Zihan Wu, Christopher Brooks, and Steve Oney. "Callisto: Capturing the" Why" by Connecting Conversations with Computational Narratives." In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1-13. 2020.