Reimagining Participatory Agile Development in Community-Industry Partnerships

Calvin A. Liang*
Northwestern University
USA

Emily Tseng*
Microsoft Research & University of Washington
USA

Elizabeth Fetterolf
Stanford University
USA

Mary L. Gray
Microsoft Research New England
USA

Abstract

Computing's ubiquity and accumulation of capital have positioned modern tech giants to be key players in society's responses to crises. Fulfilling this potential, however, requires methods and incentives for the industry to meaningfully support communities ("community-industry partnerships"). This paper examines one such partnership: an effort to co-develop software with and for community health workers that began in 2020 with the COVID-19 pandemic. Our multi-year, ethnographic work explores how Agile development, the industry's standard development practice, delimits the possibilities of community participation. Analyzing the project's breakdowns, we find stakeholders aligned on goals but misaligned on three key tensions: whose expertise takes primacy, how disagreements are surfaced, and how accountability is enacted. We propose reimagining these tensions as guiding principles for stronger partnerships: yielding to community expertise, embracing disagreement as productive friction, and ensuring accountability through realignment. Our framework offers guidance for community-industry partnerships to enhance societal resilience, in crises and beyond.

CCS Concepts

• **Human-centered computing → HCI theory, concepts and models**.

Keywords

Participatory design, Agile development, community-industry partnerships

ACM Reference Format:
Calvin A. Liang, Emily Tseng, Elizabeth Fetterolf, and Mary L. Gray. 2026. Reimagining Participatory Agile Development in Community-Industry Partnerships. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3772318.3791312

*These authors contributed equally to this research.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3791312

1 Introduction

Our modern global economy is increasingly centered on the technology industry, particularly a handful of computing giants, as concentrators of expertise, financial capital, and political power. As our world faces multiple crises—of climate, democracy, and social fabrics—, scholars of sociotechnical systems have begun to consider how the modern computing industry might meet the demands of the moment, by leveraging both the ubiquity of commercial computing systems and the industry's relative wealth of resources. Meaningful collaborations between community groups and tech companies ("community-industry partnerships") have the potential to support society's resilience to the crises of today and tomorrow—if we can develop the methods, incentives, and processes required for industry partners to enrich and sustain community needs. While there are reasons to be skeptical of such a future, the kind of reformation we call for is not out of reach either, given modern tech's culture of iteration and improvement. Already, for example, we see increasing interest across the industry in more participatory and community-responsive ways of building artificial intelligence (AI) systems [53, 96, 98]. For an industry that has prided itself on the act of disruption, what would happen if we turned that motivation inward?

In this paper, we take up this imperative through a critical deconstruction of one U.S.-based tech company's effort at community collaboration. We detail the study context (Section 1) and related literature (Section 2), before elaborating on our case study (Section 3). In Sections 4-7, we then show how our stakeholders diverged in their ideas of success, via three areas of misalignment—expertise, disagreement, and accountability. Throughout, we reimagine each area towards three principles for more successful community-industry partnerships: towards yielding, friction, and realignment. Our framework, summarized in Table 1, provides guidance for technologists and communities collaborating to address crises today, and in the uncertain future.

1.1 Study Context

As the COVID-19 pandemic took hold in 2019 and 2020, community health workers faced a formidable challenge. Those on the ground were tasked with pushing their already maxed-out capacities: to not only care for those ill with COVID, but also administer vaccines as they became available to community members underserved by today's healthcare systems. Doing so successfully and effectively could not have been possible without their decades of experience as public health experts and their relational ties to their communities that, though varied in demographics, needs, and motivations, united in advancing health equity.

These vital efforts took place across the world. In the United States, the lack of national health services rallied public and private sectors alike around the banner of societal resilience. Many pledged to combine resources to assist critical community health services (e.g., [81]). During this time, the fourth author partnered with a group of public health-focused Community Based Organizations (CBOs) in [locality redacted for anonymous review], with the aim of marshaling their expertise in community health alongside expertise in social sciences and computer science to understand COVID's impact on historically marginalized communities and identify how technology might facilitate CBOs' frontline work.

As the worldwide emergency continued through 2020 and 2021, one key CBO received state funding to accelerate equitable vaccination uptake in historically marginalized communities. It then partnered with CBOs across the state to coordinate vaccine drives. Spurred by that backbone grant funding, this CBO convened a coalition of fourteen CBOs that collectively provided everything from food boxes and cooking classes to lead remediation programs and visiting nurse services alongside vaccine distribution. In tandem, the fourth author convened a team of industry-affiliated engineers, researchers, and project and product managers—both full-time employees of the tech company and external contractors—to work with this network of CBOs to design and develop new web-based tools for information management of the coalition's efforts.

In the fall and summer of 2021, this coalition of researchers, engineers, and CBOs identified a specific technology-based need: creating a secure, shareable, and open-source intake form that independent CBOs could use for their vaccine drives. By then, a leadership group had organized within the CBO network, including stakeholders from four organizations who played a key role in directing the development effort. Drawing on their expertise, the intake form was envisioned as the first component of a series of modules to house and handle data for tracking client case management and larger programmatic needs, from staffing vaccine clinics to food drives, and eventually to enable data sharing across organizations.

1.2 Study Motivation

This paper focuses on the process of co-developing software under the unique circumstances of a community-driven software development project that sought to combine the complementary strengths of a community health CBO network and a large tech company. As a research team, we were both observers and active contributors to a software development process that brought together the community-based organizations as subject matter experts, with engineers from both contracted software firms and the parent tech company.

Our development process pursued a Participatory Agile approach: one that focused on (a) incorporating different forms of expertise, including community experts, by borrowing from tenets of Participatory Design (PD); and (b) working in an iterative Agile style involving a regular cadence of check-ins, not only on project progress but also ways of working together. Our decision to enact these principles came from the research team's investment in community-based research and its growing prevalence in Human-Computer Interaction (e.g., [26] and technology development (e.g., [74]). Given Agile's current primacy in software development processes, we drew from its structure by identifying a most viable product (MVP), pursuing two-week sprints, and facilitating regular team meetings to review progress and establish future plans.

We had expected that combining Agile and PD in practice would support the project's vision of success—one that not only developed the intake tool in a time- and cost-efficient way but also prioritized and acted on the CBO network's needs. Agile is designed to be an iterative, adaptive, and customer-centered way of developing software [37], while Participatory Design is built upon democratic principles to bring all stakeholders into the design process to meet a broad set of needs [86]. With both methods' investment in human-centered ways of building computing systems, they would seem complementary.

However, over the course of the project we witnessed breakdowns in our collaborative efforts. The project ended in an outcome familiar to so many endeavors that position computing as key drivers for social good—it failed to deliver upon its hopeful intentions (e.g., [6]). At the time of writing, the CBO network and the tech company had moved on to pursuing similar goals with different partners. Understanding the breakdowns in the 2020-2022 collaboration thus provides an illustrative case study for how community-industry partnerships might proceed more successfully in the future. We focus in this paper on the following research questions:

(1) RQ1: How do competing values and definitions of "success" complicate partnerships between industry and community organizations?
(2) RQ2: What changes to current Agile development are necessary to sustain meaningful, participatory community-industry partnerships that stay committed to the social justice aims and priorities of PD?

Our ethnographic and reflective analysis interrogates the challenges of bringing PD practices and the tacit logics of Agile together to build software that meets all stakeholders' needs. Building on previous efforts to bridge participatory design and community engagement with engineering practices (e.g., [74, 89]), our specific contributions are as follows. First, we outline a critique of how introducing subject matter experts into an Agile loop can fail, stemming from divergent interpretations of what constitutes success. Next, we demonstrate how Agile and PD can share a vision of success, but ultimately misalign on three core areas: expertise, disagreement, and accountability. Third, we reimagine these three areas as principles of yielding, friction, and realignment, providing a framework for expanding the Agile engineering process to meaningfully and structurally embrace core principles of Participatory Design.

In presenting these visions, we remain attentive to the structural incentives that lead the tech sector to prioritize productivity and profitability over social accountability. In line with visioning traditions in computing, however, there is value in refusing the inevitability of these forces and examining alternative futures [34, 40]. As science journalist Ed Yong describes the purpose of his work, to reimagine is "to use the work to expand our moral imagination [...] to show the full scope of what is possible" [107]. Therefore, we align with the overall project of similar efforts to reimagine community-industry collaborations [35, 94] by offering conceptual and pragmatic points of intervention, of what is possible. Our lenses are starting points for technologists working within these structures to change things from the inside. We also hope they provoke points of connection for community members, organizers, and advocates to identify how tech company partnerships can better suit their needs.

2 Related Work

This work builds upon existing literature that outlines the strengths and shortcomings of Agile approaches (2.1), technology development for social good (2.2), and participation (2.3).

2.1 The Values and Practices of Agile Development

Agile is the latest of a long history of organizational approaches to software engineering, created and popularized to embrace modern computing's breakneck pace of innovation. Software development requires complex yet seamless coordination among groups of people and tasks and has been a field of study since the first NATO conference on software engineering was held in 1968, where coders, engineers, and scholars established a culture of continuous self-improvement [78]. For decades, the dominant paradigm in software was the waterfall method: a structured approach in which development teams followed predetermined, linear plans in sequential phases that led to the next version [7, 9]. However, when the software industry boomed in the 1990s, development teams needed approaches to bring products to market faster, cut down on the costs of development, and adapt to unpredictable and ever-changing real-world conditions. Agile was developed to give developers more freedom and flexibility than waterfall and other predecessors, and ultimately to deliver the most market-responsive products with minimal cost [42, 67].

While there are several variations of Agile, such as scrum, lean, kanban, and more, the overall framework is united by four core values, outlined in the The Agile Development Manifesto: "individuals and interactions over processes and tools", "working software over comprehensive documentation", "customer collaboration over contract negotiation", and "responding to change over following a plan" [37]. From these values, agile software development focuses on three main areas: people, process, and outcomes [14, 62, 83].

However, inquiry into its process has revealed how Agile in practice diverges from its promised ideals. In the decades since Agile's popularization in the 2000s, scholars have identified ways that Agile structurally prioritizes the cost-efficiency of a product over its utility to its intended users, and their ultimate satisfaction [76]. One indicator of Agile's prioritization of products over people comes down to how the Agile process tracks progress and values success, operationalized as Key Performance Indicators (KPIs). KPIs have been widely discussed in the management science [18, 56, 92], information science [77, 104], and computer science [12, 75] literature. With roots as a management tool, KPIs have become an integral part of many Agile development processes as a method of quantifying performance [43]. However, as sociologists of measurement noted, the very act of quantifying success in fact changes its terms [22, 63]. Rankings and measurements can exhibit "convergent reactivity," in which institutional processes and work practices are reshaped around the measurement target [36]. Put another way, choosing how to measure performance is always a social process. As the popular adage Goodhart's law goes, "When a measure becomes a target, it ceases to be a good measure" [93].

Which proxies chosen as signals of technological success matters immensely. For example, the widespread use of user engagement as a proxy for "digital well-being" produces a broad, individualized definition of the concept that does not take into account its culturally and environmentally specific meanings [30]). Referencing 2021 U.S. Congressional hearings on Facebook's role in political radicalization, Posner argues that when a company's sole focus is to maximize user engagement there is no room to question potential holistic impact of its technology on society [76]. Building a product towards individual engagement is not inherently wrong but can lead teams to focus on 'scale', defined as amassing the largest quantity of active and engaged users possible. Then, when Agile teams set out to build systems that advance societal aims, incentives "to scale" can overshadow attention to a diverse range of stakeholders.

Part of Agile's narrow vision and quantification of success is due to its core focus: developing software at the quickest rate at minimal cost. Teams are encouraged to pursue technologies that demonstrate "stickiness", another word for engagement, and thus generate profit, over ones that are socially productive [51, 103]. The imperative towards profit is even more palpable for teams backed by venture capital (VC), which exercises an enormous influence in the modern tech industry. Shestakofsky's ethnographic study of one such startup showcases how venture-backed organizations confront "valuation lag – a temporal and imaginative gap between a venture capital firm's investment in a company and its ability to realize returns" [88, p. 17]. In response, startups experience "drags" in which they attempt to reshape their work around the primarily speculative goals of their VC backers: goals that come into conflict with ideals of "social good" [88]. Our experiences reveal how a push to increase the number of active users also drives developers to move at a pace that disconnects from those who cannot keep up. We identify competing priorities for Agile that complicate commitments to PD as envisioned as a future path for community-industry partnerships. Therefore, if technologists wish to create systems that incorporate community needs and voices, then we must evolve our frameworks for Agile and PD.

Both "participation" and "tech for good" have been proposed by researchers as ways to balance out Agile's intentional or unexamined prioritization of engineering expertise. These definitions are in many cases overlapping, as participation can be used as a method in "tech for good" contexts; our project has some elements of both of these interrelated movements within the tech sector. Hence to re-imagine Agile as a process, it is necessary to first understand the existing critiques of both.

2.2 Critiques of the Tech for Good Movement

"Tech for good" is a porous label that is often conflated with terms like "civic technology" and "public interest technology" [82]. Despite the term's malleability, Rider broadly defines it as "the intentional building of digital technologies to have a positive impact on the world" [82, p. 5005]. While "tech for good" initiatives have gained traction, studies have found that, again and again, technical projects that aim to benefit and/or include community groups can end up causing harm and perpetuating existing inequities (e.g. [6, 44, 82, 90]).

As Rider notes, many tech for good projects have emerged from a critique of engineering culture, from both outside the sector and among its practitioners. Engineering culture has been a subject of much social scientific analysis. In the information and communication industry of the 1990s and early 2000s, studies of engineers demonstrated how flexibility, networking, and scrappiness became key aspects of workplace culture in the face of postindustrial precarity [61, 69, 101]. From the 2010s to the 2020s, more recent studies of engineers underscore a near religious emphasis on work [21], the racism and sexism dominant in the tech industry [20], and the ever-present push of venture capital funding and outsourcing [54, 88]. Tech workers themselves have their own critiques of economic inequality and their role in it [32] as well as ethical concerns about the companies for which they work [106].

Rider argues that tech workers often grapple with these critiques through "repair work" – a process through which they use "tech for good" projects to heal their affective ties to an idealized vision of their career [82]. However, while these projects may have benefits for the engineers at the helm, they are often ineffective at best or harmful at worst to the nonprofits and civic organizations whom they purport to serve. Ethnographic studies exploring the trajectory of such endeavors within schools, libraries, nonprofits and government institutions have found that ideals and outcomes are often misaligned [6, 44, 90]; in some cases, the charisma of the technological object itself allows development to continue in the face of repeated failures [6]. Through common sense ideas about the promise of user access [44], and gendered, racialized ideas of who those users should be [6], institutions continue to perpetuate the notion that simple, agilely developed technical solutions can solve messy, complex social problems. This paper builds upon the critique of tech for good initiatives by mapping where and when engineering values diverged with those of community partners in our project and also offer specific alternatives to developing technologies that aim to hold commercial computing accountable to social needs.

2.3 The Promise and Limits of Contestation as Participation

The idea of "participation" in design is rooted in Scandinavian design practice and has become a popular tactic among technologists who hope to incorporate "user" voices into their work [45]. While thousands of articles in the ACM library reference participatory methods, a distinct literature has emerged outlining both their challenges and their risks. A primary challenge is that the individuals who are tasked with speaking on behalf of communities come to represent the whole [33]. Furthermore, the history of injustices against marginalized groups such as Black Americans makes trust between researchers and participants rightfully fraught in design scenarios [47]. CBOs working with technology researchers face barriers to enacting their own visions for technology development, or "hostile ecologies," given that the tech landscape, the economic resources available, and the concept of innovation itself are often in opposition to their priorities [97]. These challenges can result in "participation washing" via which participatory language excuses the ethical failures of the project [91]. Participatory methods also risk putting severe affective burdens on the individuals they purport to serve [33], and at their worst can result in predatory inclusion [31].

As this past work has shown, one of the primary challenges participatory projects face can be captured through "pseudo participation," when participants have the illusion of agency with no actual control within the process [72]. Key strategies from the literature seek to address these challenges by distributing control over a technology's development. First, researchers place an emphasis on contestation and/or contestability. Contestability, as defined by Alfrink et al. [4], involves building "adversarial debate" into the lifecycle of a sociotechnical system [4, p. 3]. Purported as a potential remedy to automated harms, contestation is often figured as a way for researchers, participants, developers, and various stakeholders to challenge machine learning models at various points in the development and rollout process [4, 50, 102]. Beyond being just a tactic, Sawhney and Tran [85] observe that many PD projects exist within "ecologies of contestation," in which conflicts between various parties are essential aspects of the process and important objects of study for researchers. While the literature on contestation provides practical examples of how to implement these processes, the idea of "contestable" systems runs the risk of reproducing the problems that critiques of participation raise – in the case of high stakes automated systems, such as ML approaches to mental health [50], contestability could risk becoming a perfunctory process that does not challenge the power structures embedded in the actual building of automated systems.

Related to but distinct from contestation is disagreement, which presents a harder to measure but potentially more productive framing. In their study of a crowdsourced policy-making process in Finland, Aitamurto and colleagues [3] found that disagreement can play a constructive role in these policy development processes, especially for idea generation. In their work with neurodiverse children, researchers did not shy away from conflict in sessions and instead analyzed how these conflicts were both destructive and constructive to the design process [39]. Much like its role in this study, disagreement led to the possibility of reimagining how conditions could be different to best serve everyone involved.

Refusal is another tool available to users when engaging with computational systems. HCI has drawn from Feminist and Indigenous refusal practices to highlight the importance of saying no when developing technologies [1, 2, 17, 23, 41, 100, 108]. As DeVrio and colleagues outline, refusal can take on various forms: impairing technologies, depriving systems of needed data, and confusing algorithms by inputting false or unexpected data [27]. Zong and Matias build upon this body of work by proposing "data refusal from below", an approach that calls for attending to "refusal by people harmed by data systems who have low direct power over technology creation" [108, p. 4].

Much like disagreement, friction offers a perspective on "the awkward, unequal, unstable, and creative qualities of interconnection across difference" [99]. When two entities encounter one another, it is friction that fosters a pathway for further movement and action. As Tsing describes, "a wheel turns because of its encounter with the surface of the road; spinning in the air it goes nowhere" [99, p. 33]. In this generative turn, friction echoes approaches like contestability in that it identifies disagreement but builds upon these ideas by encouraging translation into action and reconciliation. A fricative approach to social mobilization does not assume agreement, homogeneity of thought, and resolution as explicit goals; instead, as Tsing writes, the point is to "appreciate how we can use diversity as well as possible" [99, p. 21]. It is through embracing productive friction that differing opinions can lead to development decisions that meet many kinds of needs.

Community-collaborative, participatory approaches provide opportunities for such input. These encompass a range of methods for engaging with communities in computing research including Action Research and Community-Based Participatory Research [26, 48, 55]. Engaging with communities offers new design potentials and pathways towards equitable development, which can be particularly impactful for marginalized groups who are often left out of critical conversations (e.g., [46, 74, 89]. These participatory approaches offer a way for people to be involved upstream, as sources of user insights and after deployment, as a practice of contestation or disagreement. Agile, as typically applied, provides methods for incorporating user feedback that can come from PD. But how can practitioners pull these things together in a way that preserves the best elements of each without engineering logics of "success," measured through adoption, commercial viability, or a mix of the two, overriding commitments to community-defined priorities? We build upon these participatory efforts to articulate a vision for how to integrate community-collaborative approaches into industry Agile methods, especially when faced with breakdowns from divergent priorities. Doing so, however, demands a restructuring of both practices.

3 Case Study: Co-Developing a Community Client Intake Form

3.1 Data Overview

Our research imagined that three groups of decision-makers would need to share power to set goals for developing and deploying a client-intake form that could augment or replace existing analog processes and increase efficiency and efficacy for community health workers managing the frontlines of the pandemic. First, CBOs that consisted of community health workers (CHWs), their leadership team, and a technology team that could operate as the CBOs' own internal concierge tech support team. Next, engineering resources, staffed by both full-time engineers working for a major software products and services company and an external, contracted engineering team. Finally, the industry-based sociotechnical research team that included the author team. The nature of each group, and their interactions with each other, are summarized as follows:

- *Engineering <> Research*: As the research team, we participated in frequent meetings with members of the engineering teams to discuss findings from our engagements with CBOs. For instance, we relayed that CHWs emphasized the need for the tool to support multiple languages including Spanish so that their multi-lingual teams of volunteers could interact with the system.
- *Research team <> CBOs*: The research team held frequent meetings and monthly interviews with CHWs, the CBO tech team, and CBO leaders. For instance, we attended a monthly virtual meeting among the network of CBOs to share the project's progress and solicit feedback on both the development and collaborative process.
- *Engineers <> CBOs*: The engineering team and CBOs met occasionally, always with the research team present to facilitate exchange and bridge perspectives. Often, these meetings were purposed for the engineering team to share high-level progress reports with core CBO leadership.

The research team was deeply embedded in the development process over a two-year period, from 2020 to 2022. Our responsibilities to the wider project included: 1) interviewing CHWs, prototyping, and collecting feedback for development purposes, 2) attending engineering and CBO meetings, 3) training a Community Tech Team, nominated by the CBO leadership and paid for through a grant from the industry-based partner, to participate in the codesign process, and 4) establishing deeper relationships with CHWs through frequent interviews and focus groups to understand their satisfaction with the collaborative process and their views on the value of codesign to meeting their technical needs. Prior to engaging with CHWs, we obtained approval from our institution's Federally-Registered IRB and Ethics Review Program. For CHW meetings, we obtained informed consent to record and transcribe conversations, emphasizing that participants could pause the recording at any time or retract statements brought to them for review. We draw primarily from our observations from these collaborative meetings and insights from process-oriented discussions with CBOs as data sources for analysis in this paper.

3.2 Data Analysis

Through our roles as both members of the development team and observers of the process, we were privy to both product-specific insights and the relationships among the groups of decision-makers. Based on these observations, we employed an interpretivist situational analysis to analyze our data, which calls for mapping relationships among actors and their positions, commitments, and values [24, 25]. Each member of the research team maintained analytic memos throughout data collection, and after initial immersion in the field notes and transcripts, we individually constructed themes. We then met regularly (weekly during the active phase, then monthly) to compare insights and systematically refine codes and themes through collaborative discussion. In these sessions, we constructed situational maps of key actors and factors, debated interpretations of critical incidents, and iteratively reached consensus on the three focal tensions. This process ensured that the themes of expertise, disagreement, and accountability truly derived from the data rather than from any single researcher's perspective.

Through these iterative processes, we further identified how divergent narratives of success drove the participatory failures that we witnessed, outlined in further detail in Section 4. Additionally, our research team comes from academic disciplines in informatics, engineering, anthropology, sociology, and design. We have drawn from our own experiences with not only marginality but also public health, software development, and community-driven research to inform how we interpret our interactions with each member of the project and our analytical approach. We have particularly drawn attention to and spoken openly with the community partners about how our experiences both connect and diverge, inspired by four tensions of HCI research with marginalized people [65]. In thinking across these tensions of exploitation, membership, disclosure, and allyship, our research team has been intentional about how our partnerships are built upon both similar commitments to equitable public health and distinct experiences as we enact our work.

This work is subject to the limitations of all interpretive work. The findings are the product of the authors' observational and analytic lenses that come with our own biases, even as we sought to address these influences by confirming our interpretations with project members. In our depiction of our case study, we as the research team were not a neutral party throughout the development process. Rather, our own biases and values—including a proclivity for equitable community participation and agency and the close relationships we established with the community health workers—informed our interactions with team members and our resulting interpretations. For example, our research team was invested in studying the social implications and cultural dynamics of technologies. We are measured by our contributions by our impact on academic scholarship so incentives (and pressures) to publish our research in peer-reviewed venues were as present as they are at other research-intensive institutions. And, of course, our prolonged engagement with the CHWs created an empathy bias toward their frustrations. We acknowledge that this positioning may have led us to interpret certain events in favor of the CBOs and with some skepticism about the power of Agile and participatory approaches to overcome industry pressures to produce an artifact. To mitigate this, we deliberately sought feedback on our preliminary conclusions from other stakeholders (e.g. engineers on the team, other community members adjacent to the participating CBOs) and remained conscious of contrasting viewpoints during analysis. By reflecting on how our own commitments (to equitable participation and community agency) steered our focus, we make our positionality explicit and treat our potential biases as data – something to be openly examined rather than hidden.

In addition, we, as a research team, were involved in the very processes and decisions that we critique in this paper. We do not wish to absolve ourselves from any of the breakdowns presented and seek to position our own estimates of failure as object lessons for colleagues invested in the prospects of PD for more inclusive tech [29, 95]. Finally, because our analysis is grounded in a single case study, we do not claim that our visions will provide generalizable solutions to every future partnership. By outlining the specific details that informed our observations, including the organizations and processes involved, we hope that future partnerships between CBOs and large tech companies can identify what pieces and recommendations are transferable to their project's contexts and what can be left behind.

4 Pursuing Success

Given that the primary goal of this work was to use participatory design to build a client intake form in collaboration with community health workers, how well did we, as research and engineering partners, do? To what extent did we materially improve community health workers' ability to attend to and document their duties through a rigorous and robust incorporation of their expertise, as we initially set out to accomplish? From a purely engineering perspective, the various development teams were indeed successful in building an open-source client intake tool. Despite this short-term accomplishment, however, there were participatory failures that led to the collapse of the community-industry partnership with the CBO network. Those failures led to an overhaul of the long-term research project and an explicit focus on community-driven innovation. It remains an open research question as to whether industry-based software and data infrastructure development can incorporate participatory design to create technologies that prioritize democratic, collaborative processes and recognize the expertise of society's marginalized stakeholders as KPIs over more familiar measures of tech success.

Our collaboration was initially unified by a common goal to build software that could materially improve CHWs' ability to attend to their duties and improve participatory co-development processes that unite industry and social groups for societal resilience. Through an Agile framework, we followed rapid sprints to develop specific features of the intake tool towards a version that could meet a baseline level of CBOs' needs. Along the way, community partners would guide development by providing feedback based on how well each proposed feature could support their work. Success provided the upstream justification for determining timelines and allocating resources to setting priorities for feature requirements; however, as we moved further into development, differences in how we interpreted achieving success arose. Upon reflection, we observed how it is easy to agree on success as a value, but it becomes much more difficult when it comes to agreeing how to enact and evaluate it.

Our analysis, summarized in Table 1, identifies three areas in which these diverging interpretations of success played out: expertise, disagreement, and accountability. Building upon previous discussions around each within the participatory literature (e.g., [28, 60, 105]), our work both situates each within the context of community-industry partnerships and makes connections to wider pursuits of success. The next sections are structured as follows: a) we ground each section with a scene from our observations that establishes collaborative tension, b) we discuss how each area illuminates current limitations in Agile development, and c) we reimagine each area towards principles of future community-industry partnerships that advance social needs.

5 Reimagining Expertise

5.1 Scene 1: Digital vs Paper

A familiar grid of faces popped up on the screen as our group started its weekly standup meeting. Scattered across different time zones, a 4-person contract engineering team, 3 industry-based researchers, and 4 community-based health organization workers waved on camera from their home offices or said hello, with their cameras turned off. Even in this virtual space, a palpable tension quickly rose, as the different teams debated a proposed "Paper 2.0" feature.

Tension	Scene Summary	Current Limitations	Reimaginings
Expertise	Paper 2.0: engineering teams resisted building functionality for creating paper outputs because it was not worth the effort.	Computational biases favor technical knowledge when making engineering decisions. Non-engineers' knowledge is framed narrowly through the lens of use.	Embrace principle of yielding by expanding expertise beyond technical knowledge and use-centric framings.
Disagreement	Platform Migration: a community leader raised concern with a decision to migrate to a different platform on behalf of community needs.	While valuable, existing methods for contestation [66] and refusal [108] rely on systems that have already been deployed.	Embrace principle of friction by fostering opportunities for disagreement at times in development that can lead to changes in decision-making (prefigurative timing) and by insisting those with power are in the room (intentional presence).
Accountability	The Retrospective: time allocated to reflect upon the process was able to identify how to improve but unable to create plans to enact these ideas.	KPIs and product-specific metrics of success lead to scarcity that prohibits accountability.	Embrace principle of realignment by expanding accountability at the level of incentives (e.g. KPIs) and processes (e.g., sprint planning).

Table 1: Summary of Key Themes from Community-Industry Partnership Analysis

This heated discussion began smoldering months earlier. The industry-based research team had brought on the contracted engineers in early 2021 to help cocreate digital tools that focused on what CHWs might need to document their work on vaccine equity drives at the height of the COVID-19 pandemic. The community groups working with vulnerable communities wanted a way to document their shared efforts and analyze the data collected together in the field. In this research, the CHWs were more than customers. They were subject matter experts, teaching us their perspective on what a shared, web-based intake tool for tracking Covid support and complex community health needs might look like. These workers had long relied on paper more than computers for various tasks, such as typical intake and assessment forms, keeping attendance, and taking notes during client visits. Despite the digital age, paper remained a preferred medium for many CHWs. It was simple, reliable, and a familiar part of workers' and clients' experiences with social services.

The proposed Paper 2.0 feature aimed to bridge the gap between the comfort of paper and the efficiency of digital tools. It included functionalities like offline data entry, file attachments, and the ability to scan handwritten forms into the system using a phone. Despite these promising features, the engineering team, with less than a month left under contract with the industry partner, voiced concern about giving Paper 2.0 priority. "Implementing Paper 2.0 will complicate our current system," argued one of the senior engineers, their voice dropping out slightly through the speakers. "We need to focus on streamlining our existing features, not adding more layers." "But the CHWs want this," countered one of the Community Tech Team members who had been trained by the research team and closely involved in the discovery phase of the participatory research. "They've told us that paper feels safer and more reliable. We can't ignore their feedback."

The room fell silent as the team pondered what had been said. On one side were many of the engineers, both full-time employees of the industry partner and the external team of contractors managed by those employees, who saw the technical challenges and potential pitfalls of integrating Paper 2.0. On the other side were community tech team members, CBO leaders, and other parts of the research team who had more opportunities to investigate the real-world needs of the CHWs and the potential benefits of the new feature. "We're not just building software," a Community Tech Team member continued, breaking the silence. "We're creating tools that people rely on to do their jobs effectively. If Paper 2.0 can make their lives easier, isn't it worth the effort?"

The introduction of Paper 2.0 faced several key challenges that needed to be addressed to ensure its successful adoption. The administrative burden was another significant challenge, as using both paper and digital systems led to duplicative work and increased the likelihood of human error. Paper also offered emotional reassurance to CHWs and community volunteers, particularly those who were not tech-savvy. Therefore, creating a digital tool that could offer the same level of comfort and trust was imperative. Lastly, connectivity issues posed a challenge that paper did not. A tool with Paper 2.0 and offline mode capabilities would not have to depend on an internet connection, making it more reliable in areas with poor connectivity. By addressing these challenges, Paper 2.0 aimed to create a digital tool that was efficient, trusted, and could be embraced by CBOs and their clients where unreliable digital infrastructures were part and parcel of the broader realities of limited public sector and clinical support in the neighborhoods and regions of the marginalized communities CHWs served. However, in the end, the Paper 2.0 feature was not built in the eventually deployed intake tool.

5.2 How expertise is currently mobilized

Whose voice is listened to—and therefore who holds power—plays a pivotal role in how development decisions are made. What Holzmeyer refers to as "the politics of knowledge and expertise" is on full display in the previously depicted scene between engineers and community partners [51]. The engineers hesitated to respond to the CHW's calls for implementing paper-based features, voicing concerns about meeting contract deadlines and having deliverables to show for their work. Embedded in their resistance were indications of how they understood what a successful outcome for them was.

Traditional Agile approaches have two limitations when it comes to expertise beyond that of the bench engineer that have influence over how to achieve success. First, there is a computational bias that favors technical knowledge gathered through telemetry or 'common sense' when making engineering decisions. While we are not implying that this knowledge should be disregarded, we are wary of the extent to which this specific form of domain expertise dictates what becomes embedded into technical systems [51]. To this end, Holzmeyer describes "a politics of 'domain expertise"' in which "some people's values, priorities, epistemologies, types of evidence, forms of data and diagnostic or prescriptive frames will predominate – marginalizing or foreclosing other possibilities, especially from systemic perspectives" [51, p. 108-109]. Previous scholarship has criticized design and engineering practices's reliance on I-methodology, or "the use of one's own personal experience to guide decision making", that supplants additional perspectives [38, p. 3]. Oudshoorn et al. similarly depict how development teams set out to build solutions for everyone but ended up basing decisions upon their own preferences [71].

The second limitation of expertise is that non-engineers' knowledge is framed narrowly through the lens of use. Such a perspective is imprecise when seeking to fully understand people and their needs, despite Agile's principle in understanding customers. Previous scholarship has pointed out that technologists often construct "the user" not as multifaceted people, but instead as one-dimensional beings whose only interests lie in technology usage [13]. Sharrock and Anderson point out how design teams strategically invent the user and their needs to justify their own decisions [87]. In the case of "tech for good" projects, an idealized user can also map onto visions about what the project should be or represent [6]. Technologists additionally ascribe morality onto adoption, where a "good" user uses systems as they were intended and a "bad" user is one who rejects [79]. In an HCI context, scholars have advocated for the legitimate perspectives for non-use when designing technologies [15, 16, 84].

Across this body of work is an important lesson for technology builders today: people who interact with technology do not only exist to use, adopt, and consume. Framing people solely through the priorities of the development team yields a one-sided and limited depiction of who uses software. These narratives can enable a techno-idealism [90] and shift focus onto the affective needs of the engineers, rather than those who the technology is actually meant for [82]. In the scene presented above, the contracted engineering team struggled to separate what they imagined a typical user would want—digital outputs—from what the actual CHWs asked for—paper— leading to tension. This strain became most visible when the surfaced needs of community members seemed to be a questionable concern to the engineering team, focused on digitization features as the higher priority.

Valuing expertise through more than just a use-centric lens, then, inexplicably leads to a more robust understanding of how people actually use technology. With this clearer picture in mind, technologies can then lead to impacts that are more closely aligned with peoples' and communities' needs. If in the above scene the engineers agreed to implement Paper 2.0 functionality, community health workers could enjoy trust with the intake tool, comfort with working in an analog workflow, and ease of having to worry about having internet access. In other words, the community partners could also experience a successful outcome.

5.3 How we reimagine expertise

In the scene depicted above, we saw how engineers positioned their technical knowledge to justify decisions of what was worth investing effort into and pushed back against a decision that drew from the CHW's expertise. If a traditional approach holds a computational bias as exhibited in the engineers' hesitation, then our collaborative partnership reveals a different manner of valuing expertise in the development process. Our vision for community partnership in technology development is two-fold, as we call for not just expanding expertise beyond technical knowledge but also yielding decision-making power to these perspectives.

Participatory design and community-based research offer a model for challenging dominant narratives of what counts as expertise and provided inspiration for our own engagement with the CHWs' knowledge (e.g., [48]). Future development must move beyond a computational bias by considering that non-technical perspectives can even be more advantageous to making development decisions towards socially impactful technologies. Part of this shift must also move away from a user-centric framing of people and towards a holistic appreciation of different kinds of expertise that those who will interact with the system can offer. There is significance in challenging dominant narratives of who counts as an authority (Lynch, 2020). Doing so shifts power and lends credence to voices that are not normally accounted for in engineering decisions but, as we argue, provide just as much value when developing technologies for social advancement.

Our reimagining, then, involves a principle of yielding to non-technical forms of expertise. Community-centered approaches have emphasized the importance of involvement, and yielding to the priorities of communities is a meaningful extension for building technologies that seek to meet their needs [26]. Decision making in a development setting is a power-laden act; the people who make and enact a decision thereby materialize their power [19]. Yielding, then, is less about who gets to make decisions and more focused on how they are made—whether done in partnership or isolation. Our vision of yielding disrupts the presumption that engineering teams automatically make decisions solely because they hold technical expertise; rather, community partners' perspectives are equally as critical to building systems that are responsive to people's needs. In a technology development for social good setting, holding computational knowledge should not come with unquestionable power. Instead, its worth should be commensurate with community partners' own expertise.

6 Reimagining Disagreement

6.1 Scene 2: Platform Migration

As the determined MVP deployment date neared, the cadence of meetings with CBOs, engineering team members, and the research team picked up. During one meeting, the engineering team met with research members and four members of leadership among the community partners as part of our regular project check-in meetings. At this point in the project, we specifically came together to discuss the current state of the tool and its future. So far, CHWs had begun working with an initial version of the intake tool to assist their work. One CHW had already transferred her database of client information onto the tool and had been drawing from its capabilities in earnest.

Prior to this meeting, the engineering and research teams had been discussing the possibility of transferring from the tool we had been developing up until this point to an open-source alternative platform. While the open-source platform had its downsides, its allure was that it was further developed than what we had so far. Switching platforms, then, would conserve engineering time and effort. Part of this decision came with the acknowledgement that many CHWs were already using the intake form that had been built for them. Having already spent so much time and effort setting up their databases on our tool, it would understandably be frustrating to ask these CHWs to do that again on another platform in a couple of months. In addition, the transition to the open-source platform would take a few months, meaning the CHWs would be expected to continue interacting with and storing their data through the original intake tool until the new tool would be ready.

Amidst the back and forths between engineering team members weighing the transition decision, one member of the CBO leadership team, Terry, spoke up. From her own history of being exploited by tech companies, she was particularly attuned to collaborations that were partnerships in name but not in practice. Terry was conscious of how often decisions were made on CHWs behalf but without their input. She shared that she struggled to see the point of continuing with the intake form:

> "*Two parts of my brain are working... one part knows that this is all working towards something else and that makes sense, but the other part thinks that it's a waste of time to put any more data or effort into this thing if we are just going to switch again.*"

This was followed by a serious pause in conversation as everyone took in her words. It was a very fair point and instigated a turning point in our conversation. From there, senior engineering team members committed company resources to handling data transfer to the alternative platform.

6.2 How disagreement is currently mobilized

Meaningful community-industry partnerships, in which a range of community perspectives are not only represented but also respected, inherently involve tension, dissent, and adaptation. In the scene above, prior to Terry speaking up, the development team had decided on a direction that served their needs—a shortcut that would save them time and resources. However, it was only through Terry's disagreement that the development team understood the toll their decision would have on the community partners. The team's initial direction held not only tangible consequences for the community partners—adding labor to migrate their data to a new platform, learn that new system, and deal with any problems along the way—but also more symbolic signals that their partnership was less collaborative and more top-down. Her intervention was a turning point: the industry partners committed additional engineering time to the project, to assist with the additional labor a platform migration would require.

What we witnessed in this scene was a disagreement raised amid a pivotal transition that advocated for the needs of the CHWs who would be affected by the decision. As we have outlined in Section 2, HCI has provided ways for end users to challenge and correct engineering decisions (e.g., [5, 85, 102]. Contestability, for example, has become a popular method for protecting users by giving them the ability to "contest, appeal, or challenge algorithmic decisions" [66, p. 1][58, 59]. Especially in the context of machine learning models, governing bodies have understood the value of disputing algorithmic judgements. Both the European Union's General Data Protection Regulation (GDPR) and Australia's AI Ethics Framework recognize contestability as an integral part of developing ethical AI systems [70, 80]. As part of this ethical move towards contestability, making room for dissent or refusal establishes the user as both decision subject and expert [50].

The existing approaches outlined in Section 2.3 are held together by an understanding that "dissensus and confrontation [are] inherent yet productive aspects of democracy, drawing attention to the plurality of viewpoints that fundamentally can never be fully resolved" [60, p. 4]. While these methods are valuable for challenging engineering decisions, a crucial limitation lies in when these moments take place. Contestability, refusal, and related approaches rely on surfaced dissension after a technology has been deployed. Zong and Matias have similarly discussed the potential importance of timing by outlining how refusal can occur reactively to harms that have already taken place or, in alignment with our critique, proactively to prevent future harm [108]. With this focus on timing and structure, our reimagining focuses on how to practically integrate disagreement into an Agile process.

6.3 How we reimagine disagreement

Our vision of disagreement moves away from a common reflex in design and development to march solely towards consensus. We embrace a principle of friction, which has been developed in previous work on civic and community engagement [11, 26, 60]. In those contexts, friction works to reveal underlying value misalignments, catalyze decision making in favor of community partners' priorities, and disrupt traditional power hierarchies [60]. As Asad et al. further describe, "friction becomes precisely the attribute that lets residents and community organizations know when their interests are not being served. Similarly, friction gives them traction to influence the process to arrive at a different—and, ideally, more beneficial—outcome" [11, p. 2023]. When communities are on the wrong end of a power hierarchy, be it in a civic setting or in our setting of industrial engineering, friction provides the means for people to advocate for their interests and align the priorities of the powerful to their own.

Building on this literature, we focus on how friction translates to the context of community-industry partnerships where, unlike in the civic contexts at play in previous literature, the terms of engagement are delimited by the Agile development cycle, and the social contract of democratic governance cannot be assumed. In our setting, we find encounters of friction are even more critical for revealing power hierarchies, giving teams the opportunity to fully understand a decision's benefits and costs. To reimagine friction for these conditions, we focus on two elements: prefigurative timing and intentional presence.

Regarding timing, a notable aspect of this scene is when Terry's disagreement took place—a development inflection point to decide whether to migrate to a new platform or not. In raising this issue, Terry prevented CHWs from spending even more of their time and energy transferring data and learning a new platform as well as the engineering team from building toward a platform that the community did not want. While there is value in users being able to call attention to computational errors, introducing disagreement prior to a system's implementation provides an upstream strategy to the root of technologically mediated harm. Such an approach was vital to previous work that integrated participatory methods into building machine learning tools from the start [95]. Asad and colleagues have conceptualized this strategy as a prefigurative design that "both anticipates an alternative future and changes current circumstances to better resemble this anticipated future" [10, 11]. Therefore, we advocate for teams working toward community-driven technologies to create room for disagreement prior to deployment by timing inclusion and disagreement at key decision-making points in the development process: what we call prefigurative timing.

In addition to the right timing, our scene demonstrates the need for the right people to be in the room: what we call intentional presence. Bringing the research and engineering team, leadership team, and the community partner leadership together in one meeting was neither happenstance nor standard practice for product development. As part of the research team's focus on participatory principles, we advocated for team meetings to include the community partners who would ultimately feel the effects of engineering decisions. Here, the leadership team, who were not typically present at these check-ins, directly witnessed Terry's expressed frustration. Convening our community partners alongside senior personnel with power to commit resources may have been inefficient, but it also provided a path towards generating meaningful change for community members. Friction, therefore, relies upon intentional presence; it is vital for key personnel to come together with community partners.

As we have emphasized in this section, disagreement is not to be shied away from; rather, it can be a productive method toward regrouping community partners' and engineering teams' priorities. A turn to friction, which builds on methods of contestation, refusal, and disagreement, surfaces tensions that teams can then leverage to improve development and achieve success. However, translating these insights into material improvements is not always a guarantee. While friction provides a productive potential, it is only an initial spark. There are crucial conditions necessary for teams to turn a spark into a flame, to translate moments of disagreement into meaningful procedural change. Prefigurative timing and intentional presence are two components that can alter the dynamics of community-industry friction, and in the following section we outline structural requirements for turning that friction into action: accountability.

7 Reimagining Accountability

7.1 Scene 3: The Retrospective

Like many Agile projects, the team had gone through several iterations of two-week Agile sprints and had scheduled a Retro, short for "retrospective" – a meeting in which the team would go through staged exercises intended to elicit reflection on the working environment. The Retro centered around a key prompting question: "*If you had to go back and start again in January, how would you want it to go? What would you do again, what would you do differently?*"

Members of the ISV immediately raised that they had experienced discomfort with their distance from the user. Engineering "*in the dark*", with "*a lot of layers between the developers and the end users*", had left them unsure of how to make specific engineering choices. Several asked to watch users actually use the application, via screen sharing or in the field—as one ISV member said, "*there's how you think people are going to use it, and then what actually happens.*" Members of the CTT largely echoed the ISV engineers' sentiments and concurred that a closer loop between engineering and user experience was important. As one described, "*I think that it makes good sense to have the engineers listen to the people who will be using this tool...there's no telling what they will be able to create with hearing their desires and needs.*" While both parties agreed in principle on a tighter loop between engineers and users, they disagreed in how they imagined the loop to occur. The ISV engineers wanted direct contact with the user base; but the CTT saw a role for themselves as translators or bridges, helping to moderate between the engineers and a user base of what they saw as highly non-technical people.

Several CTT members had had experience in corporate software development, in addition to years of experience with the community health workers who would use the tool. As one explained, "*Most of your CHWs are extremely low-tech. And I know we like to think of doctors and nurses and things as being kind of high-tech, but they're kind of specialized in the specific area, and many of them aren't even comfortable with email.*" For the CHWs to work directly with the engineers, the CTT felt their perspectives were needed to "*help engineers understand certain things that the users may not know how to articulate.*" This kind of translation role was perceived to be protective for the user. One CTT member said: "*I've seen it become very productive when you have a person that can be the voice for the users, so they don't feel like they're talking crazy or what they're saying isn't making sense.*"

However, the retro ended before the engineering team could respond to CTT's vision. This meeting stood out to us because of the unresolved nature of its ending. Despite offering a space to work out failures and identify ways to productively move forward, we were unable to do so in a meaningful way. Disagreements surfaced, but without any plan to change processes or make decisions, the space of the retro was not enough to meaningfully move forward with the failures that were surfaced, on either technical or relational levels.

7.2 How accountability is currently mobilized

Despite identifying a way to improve the collaborative process through CHWs more directly interfacing with the engineering team, the retro was unable to accommodate changes to the development schedule. We draw attention to this particular scene because it depicts our effort to enact pieces of a participatory Agile approach that we thought would lead to success. The space of a Retro should have been the place where CHWs' expertise was foregrounded, where disagreements could be heard—and where the team then took that feedback and adjusted its subsequent processes. However, this last piece did not happen and ultimately contributed to the failure of the project to deliver on community partners' needs. We consider this a failure of accountability, a third tension that links earlier tensions in expertise and disagreement to long-term and sustainable change. Accountability is necessary to establish follow-through after expanded expertise leads to more awareness of disagreements; without such mechanisms, any opportunities to meaningfully improve will collapse.

Better understanding the team structure and the hierarchy that foregrounded the development process offers a lens into whose priorities each stakeholder group adhered to—and why accountability could not be achieved. The main technology team allocated work to two separate groups of contracted engineers, the ISVs. This contract-based relationship meant that each ISV had incentives to meet expectations to sustain their working relationship and employment. The non-ISV engineering team also sought productive development as indicators of their value to the company, as they are subject to regular evaluations of their performance. The intake tool's development, the ISV's retention, and the non-ISV engineers' valuation were thus tied together through the common incentive structure of productivity, measured by whether and how often they shipped a product.

In software development, this impulse towards proven productivity manifests in the form of Key Performance Indicators, or KPIs. KPIs are embedded into the expectations of engineering outputs, tracking and proving that teams are advancing efficiently and effectively in regards to profitability, team quality, product and features, and time and cost [49]. Because of the role KPIs play in reinforcing an individual's utility to the project, they hold a powerful position in engineering decisions. KPIs do not just measure performance; they also shape how organizations and institutions see themselves and play a role in "the production of the future" [57, p. 761].

While there exists potential for embedding community-based values into KPIs, the current construction of success prioritizes product achievements (e.g., adoption, profit, efficiency) over social value. A review of key metrics that Agile teams use similarly pointed out that a majority of agile quality metrics focus on a product's outcomes and development processes [62]. While valuable, outcome-related metrics are "not sufficient in assuring ongoing quality improvement or assessing the complex, iterative sociotechnical nature of [agile software development]-related quality" [14, 68]. Recent qualitative scholarship, for instance, has shown that performance metrics, particularly automated ones, can have mixed effects—often decreasing autonomy among workers, causing stress, and changing how they relate to their jobs [64, 77].

In our work, we observed how existing incentive structures prioritize developing software at the quickest rate and most minimal cost: thus determining who and what gets valued in development decision-making. While exact KPIs were not mandated in contracts with both ISV groups, the understanding was they had to produce features—perhaps even measured by lines of code—by a certain date. This demand to generate results in a time-efficient manner was present across the entire project. For example, in Scene 1 "Paper 2.0", the engineering team's hesitance to build out a paper-based output feature could be attributed to their concerns that this would extend development time—an inefficient detour. In Scene 2 "Platform Migration", the engineers similarly invoked the efficiency of their time and effort as the reason to switch platforms. And in this scene, "The Retrospective", these imperatives hindered Agile's attempts at accountability: there was no time within the Retro to fully hear each other out, let alone time after the Retro to turn those disagreements into action.

Our observations demonstrate how KPIs delimit the forms of accountability possible. Prioritizing product-centered metrics like shipped code prevents a team from attending to their relationships with community partners. Espeland and Sauder [36] concept of the "convergent reactivity" of rankings serves a warning; when a measure becomes the standard for success, work processes can and will be reshaped around that measure. To reorient the incentive structure provided by KPIs towards meaningful community partnership, then, requires a reimagining of accountability.

7.3 How we reimagine accountability

Taken together, our three scenes show a series of missed opportunities to turn moments of yielding and friction into longer-term accountability. We witnessed how the pace and structure of the typical Agile two-week sprints become bound to pressures that structurally do not leave room for anything that deviates from meeting set product-driven goals. Disagreements surfaced (Scene 2) in ways that gave voice to community members' expertise (Scene 1), but there was little opportunity to plan and enact changes in response (Scene 3).

In response to these missed opportunities, we reimagine accountability towards a principle of realignment: the intentional creation of superstructures at the level of incentives and processes that enable a development team to turn moments of yielding and friction into longer-term, sustainable community partnerships. Our reimagining focuses on how engineering teams can be accountable to more than just developer productivity and a product's success and instead foreground the process of maintaining relationships with community partners: a core tenet in participatory design [8]. Realignment is, in our view, how to actualize this expanded vision of accountability.

At the level of incentives, we argue all members of a partnership should discuss what KPIs are important at the beginning of a collaboration and remain open to holistic and qualitative approaches to their assessment. Our principles of yielding (Section 5.3) and friction (Section 6.3) provide potential indicators of successful engagements. It may be tempting to operationalize these principles as KPIs, by counting how many times friction comes up or how often expanded expertise is consulted. However, we argue that moving towards qualitative measurements can provide more insight into these relationships and satisfaction with outcomes. Interviews and discussion can, for example, uncover what led to those moments of friction, or to what extent additional expertise aligns with engineering opinions. Further, communities should be able to define the terms of success of a product that is meant for their use. For marginalized communities, whose terms and conditions for how they can operate are so often set by others, setting what they consider to be successful reasserts their power [52]. Therefore, if we want to build equitable partnership, communities must be able to co-create what determines success.

Simultaneously, at the level of processes, all members of a partnership should intentionally create space to revisit the terms of collaboration and readjust ways of working before moving forward, even after project initiation. This relational work is, we argue, just as important as tangible outputs like shipped code. For example, in our case, a partnership might decide that a Retrospective should ensure that all members present feel free to express their opinions and implement techniques like anonymous reporting of concerns to set a level playing field. Subsequently, in sprint planning, a partnership might consider allocating enough time to enact changes to the team's development process after each Retro and bake in additional time to consult community partners who could not make it to the initial Retros.

Regardless of what teams decide to do, the adjustments we call for are not complete overhauls of existing practices, but they do offer points of leverage towards structural reimagining of what priorities steer development. Taking inspiration from Docherty and Biega [30], KPIs have the potential to reveal more than just product-based success by depicting a much wider understanding of socially-driven collaborations. In outlining how to expand both incentive structures and processes of Agile through a principle of realignment, we demonstrate how practitioners can restructure their processes to recenter and remain accountable to social and community-based needs.

8 Conclusion

Through a critical reflection upon our involvement in a community-industry software development partnership, we have shown how a specific vision of success—determined top-down and focused on material deliverables over less-tangible processes of engagement and meaning-making—structured how the engineers, research team, and community partners carried out the project. While the collective team held up community participation and collaboration as core values to the work, we saw how easily their importance fell by the wayside in favor of Agile's familiar conceptions of product-based success. In response, our reimaginings of success and conceptual interventions around expertise, disagreement, and accountability seek to establish an alternative framework for technology development that folds community priorities into its structure.

In considering expertise, we draw from an observation of the engineering team's resistance to building a feature that would create paper outputs (Section 5). When it comes to making decisions—and therefore who ultimately holds power in a collaborative effort—there exists a computational bias towards those with technical expertise. As opposed to viewing them as partners, engineering teams can easily designate community members and anyone without such technical knowledge as users and ignore any ways that community members can make contributions beyond that scope. Our broadened vision of expertise, then, involves not only amplifying who counts as an expert but also yielding to alternative expertise when making project decisions.

Rethinking disagreement, we presented a scene in which a community partner challenged an engineering decision to migrate to a different platform by reminding decision makers of the CHW's labor, time, and frustration that would come from this switch (Section 6). As a result, engineering leadership committed resources to alleviate the CHW's undue burden in the platform migration process. We highlight this scene to demonstrate how disagreement can provide productive friction engineering teams and community partners in Agile development. Distinct from current methods for identifying disputes such as contestation, refusal, and traditional user research, the timing of this community partner's disagreement occurred prior to deployment, therefore prefiguratively contending with potential harms.

In examining accountability, we outline how existing processes like sprints and retrospectives can fall short in carrying out changes necessary to maintain equitable partnerships (Section 7). We connect product-centered KPIs that structure the timing and pace of development to this missed opportunity to remain accountable to community partners. Our vision for accountability draws from a principle of realignment which attends to the structures that guide development (e.g., retrospectives and sprints). Realignment provides the capacity for moments of yielding and friction to foster into sustainable and meaningful changes that shift focus onto community partnerships.

Reimagining towards principles of yielding, friction, and realignment offers an opportunity to deepen community-industry partnerships, but implementing our visions faces several limitations. First, market competition incentivizes pursuing profit and efficiency over social responsibility. Our visions are dependent on technology companies to uphold their stated goals of supporting social well-being and require material commitment to reform to be effective. Relatedly, incorporating yielding, friction, and realignment is not a simple nor easy switch. Doing so poses potential risk to engineers who might rely on efficiency to maintain their employment, greater epistemic burden on community partners to be involved in the decision-making processes [73], and disagreements that are potentially irreconcilable. We therefore do not view the shifts we call for as prescriptive, but rather as guiding principles for teams to negotiate based on their specific goals and contexts. Next, our critique and proposals are grounded in one form of community-industry partnership, and our findings likely do not perfectly map onto other collaborations. Smaller technology companies might have more flexibility in incorporating participatory approaches but can also face greater pressure to seek out profit and efficiency. Rather than prescribing generalizable alternatives, we hope that future work can examine how the principles of yielding, friction, and realignment operate in additional forms of community-industry partnership.

With these visions in mind, we outline how the tech industry's goals of improving social conditions are out of reach without restructuring traditional Agile development methods. Drawing from participatory principles provides a pathway towards greater equity, but doing so requires a restructuring of how technologists pursue development. We offer three reimagined ways of such a reassembly to demonstrate how we might make such a future possible. Directly interfacing with the previously mentioned critiques of tech for good projects [82], the principles of yielding, friction, and realignment come together to ensure that success in community-industry partnerships is fairly distributed and does not come at the expense of one another.

Acknowledgments

First, we are grateful to the engineering teams and community health workers involved in this project. Thank you for sharing your time, energy, and experience with us. Additional thanks to those at the Social Media Collective at Microsoft Research for the thoughtful discussions and critical feedback across multiple iterations of this work. Finally, thank you to our reviewers, for thoughtful engagement that pushed our work to a stronger place.

References

[1] Sara Ahmed. 2017. No. https://feministkilljoys.com/2017/06/30/no/.
[2] Sara Ahmed. 2021. Complaint! In *Complaint!* Duke University Press.
[3] Tanja Aitamurto, Peter G Royal, and Jorge Saldivar. 2023. Disagreement, Agreement, and Elaboration in Crowdsourced Deliberation: Ideation Through Elaborated Perspectives. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–10.
[4] Kars Alfrink, Ianus Keller, Neelke Doorn, and Gerd Kortuem. 2023. Contestable camera cars: a speculative design exploration of public AI that is open and responsive to dispute. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–16.
[5] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI magazine* 35, 4 (2014), 105–120.
[6] Morgan G Ames. 2019. *The charisma machine: The life, death, and legacy of one laptop per child*. Mit Press.
[7] Bogdan-Alexandru Andrei, Andrei-Cosmin Casu-Pop, Sorin-Catalin Gheorghe, and Costin-Anton Boiangiu. 2019. A study on using waterfall and agile methods in software project management. *Journal of Information Systems & Operations Management* (2019), 125–135.
[8] Sherry R Arnstein. 1969. A ladder of citizen participation. *Journal of the American Institute of planners* 35, 4 (1969), 216–224.
[9] Harkirat Kaur Aroral. 2021. Waterfall process operations in the fast-paced world: project management exploratory analysis. *International Journal of Applied Business and Management Studies* 6, 1 (2021), 91–99.
[10] Mariam Asad. 2019. Prefigurative design as a method for research justice. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–18.
[11] Mariam Asad, Christopher A Le Dantec, Becky Nielsen, and Kate Diedrick. 2017. Creating a sociotechnical API: Designing city-scale community engagement. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2295–2306.
[12] Mohammed Badawy, AA Abd El-Aziz, Amira M Idress, Hesham Hefny, and Shrouk Hossam. 2016. A survey on exploring key performance indicators. *Future Computing and Informatics Journal* 1, 1-2 (2016), 47–52.
[13] Liam J Bannon. 1995. From human factors to human actors: The role of psychology and human-computer interaction studies in system design. In *Readings in human–computer interaction*. Elsevier, 205–214.
[14] Jo Barata, Sharon Coyle, et al. 2016. Socially-Constructed Metrics for Agile Quality: An Action Research Study. *Australasian Conferences on Information Systems* (2016).
[15] Eric P.S. Baumer, Morgan G. Ames, Jed R. Brubaker, Jenna Burrell, and Paul Dourish. 2014. Refusing, limiting, departing: why we should study technology non-use. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI EA '14)*. Association for Computing Machinery, New York, NY, USA, 65–68. doi:10.1145/2559206.2559224
[16] Eric PS Baumer, Morgan G Ames, Jenna Burrell, Jed R Brubaker, and Paul Dourish. 2015. Why study technology non-use? *First Monday* (2015).
[17] Eric PS Baumer and M Six Silberman. 2011. When the implication is not to design (technology). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2271–2274.
[18] David Beer. 2015. Productive measures: Culture and measurement in the context of everyday neoliberalism. *Big Data & Society* 2, 1 (2015), 2053951715578951.
[19] Tone Bratteteig and Ina Wagner. 2012. Disentangling power and decision-making in participatory design. In *Proceedings of the 12th Participatory Design Conference: Research Papers-Volume 1*. 41–50.
[20] Coleen Carrigan. 2024. *Cracking the Bro Code*. MIT Press.
[21] Carolyn Chen. 2022. *Work pray code: When work becomes religion in Silicon Valley*. Princeton University Press.
[22] Angele Christin. 2018. Counting clicks: Quantification and variation in web journalism in the United States and France. *Amer. J. Sociology* 123, 5 (2018), 1382–1415.
[23] Marika Cifor, Patricia Garcia, TL Cowan, Jasmine Rault, Tonia Sutherland, Anita Chan, Jennifer Rode, Anna Lauren Hoffmann, Niloufar Salehi, Lisa Nakamura, et al. 2019. Feminist data manifest-no. *Cit. on* 119 (2019).
[24] Adele E Clarke. 2007. Situational analysis. *The blackwell encyclopedia of Sociology* (2007), 1–2.
[25] Adele E Clarke. 2021. From grounded theory to situational analysis: What's new? Why? How? In *Developing Grounded Theory*. Routledge, 194–235.
[26] Ned Cooper, Tiffanie Horne, Gillian R Hayes, Courtney Heldreth, Michal Lahav, Jess Holbrook, and Lauren Wilcox. 2022. A systematic review and thematic analysis of community-collaborative approaches to computing research. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–18.
[27] Alicia DeVrio, Motahhare Eslami, and Kenneth Holstein. 2024. Building, shifting, & employing power: A taxonomy of responses from below to algorithmic harm. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1093–1106.
[28] Mark Diaz and Angela DR Smith. 2024. What Makes An Expert? Reviewing How ML Researchers Define" Expert". In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, Vol. 7. 358–370.
[29] Catherine D'Ignazio, Alexis Hope, Becky Michelson, Robyn Churchill, and Ethan Zuckerman. 2016. A Feminist HCI Approach to Designing Postpartum Technologies: " When I first saw a breast pump I was wondering if it was a joke". In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2612–2622.
[30] Niall Docherty and Asia J Biega. 2022. (Re) Politicizing digital well-being: beyond user engagements. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–13.
[31] Joseph Donia and James A Shaw. 2021. Co-design and ethical artificial intelligence for health: An agenda for critical research and practice. *Big Data & Society* 8, 2 (2021), 20539517211065248.
[32] Robert Dorschel. 2022. A new middle-class fraction with a distinct subjectivity: Tech workers and the transformation of the entrepreneurial self. *The Sociological Review* 70, 6 (2022), 1302–1320.
[33] Paul Dourish, Christopher Lawrence, Tuck Wah Leong, and Greg Wadley. 2020. On being iterated: The affective demands of design participation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.
[34] Anthony Dunne and Fiona Raby. 2024. *Speculative Everything, With a new preface by the authors: Design, Fiction, and Social Dreaming*. MIT press.
[35] Sheena Erete, Eric Corbett, Natasha Smith-Walker, Jay L Cunningham, Erin Gatz, Tina Park, Tam Perry, Lauren Wilcox, and Remi Denton. 2025. Towards Equitable Community-Industry Collaborations: Understanding the Experiences of Nonprofits' Collaborations with Tech Companies. *Proceedings of the ACM on Human-Computer Interaction* 9, 2 (2025), 1–31.
[36] Wendy Nelson Espeland and Michael Sauder. 2007. Rankings and reactivity: How public measures recreate social worlds. *American journal of sociology* 113, 1 (2007), 1–40.
[37] Martin Fowler, Jim Highsmith, et al. 2001. The agile manifesto. *Software development* 9, 8 (2001), 28–35.

[38] Sarah E Fox, Amanda Menking, Jordan Eschler, and Uba Backonja. 2020. Multiples over models: interrogating the past and collectively reimagining the future of menstrual sensemaking. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 4 (2020), 1–24.

[39] Christopher Frauenberger, Katta Spiel, Laura Scheepmaker, and Irene Posch. 2019. Nurturing constructive disagreement-Agonistic design with neurodiverse children. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–11.

[40] Anne Galloway and Catherine Caudwell. 2018. Speculative design as research method: From answers to questions and "staying with the trouble". In *Undesign*. Routledge, 85–96.

[41] Patricia Garcia, Tonia Sutherland, Niloufar Salehi, Marika Cifor, and Anubha Singh. 2022. No! Re-imagining data practices through the lens of critical refusal. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–20.

[42] 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.

[43] Moises Goldszmidt, Derek Palma, and Bikash Sabata. 2001. On the quantification of e-business capacity. In *Proceedings of the 3rd ACM conference on Electronic Commerce*. 235–244.

[44] Daniel Greene. 2021. *The promise of access: Technology, inequality, and the political economy of hope*. mit press.

[45] Judith Gregory. 2003. Scandinavian approaches to participatory design. *International Journal of Engineering Education* 19, 1 (2003), 62–74.

[46] Oliver L Haimson, Dykee Gorrell, Denny L Starks, and Zu Weinger. 2020. Designing trans technology: Defining challenges and envisioning community-centered solutions. In *proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[47] Christina Harrington, Sheena Erete, and Anne Marie Piper. 2019. Deconstructing community-based collaborative design: Towards more equitable participatory design engagements. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–25.

[48] Gillian R Hayes. 2011. The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)* 18, 3 (2011), 1–20.

[49] Jonas Heimicke, Tibor Mellert, and Albert Albers. 2020. Performance Evaluation of Agility in Product Development using targeted KPIs. In *ISPIM Conference Proceedings*. The International Society for Professional Innovation Management (ISPIM), 1–9.

[50] Tad Hirsch, Kritzia Merced, Shrikanth Narayanan, Zac E Imel, and David C Atkins. 2017. Designing contestability: Interaction design, machine learning, and mental health. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 95–99.

[51] Cheryl Holzmeyer. 2021. Beyond 'AI for Social Good'(AI4SG): social transformations—not tech-fixes—for health equity. *Interdisciplinary Science Reviews* 46, 1-2 (2021), 94–125.

[52] Peter S Hovmand. 2014. Introduction to community-based system dynamics. *Community based system dynamics. New York: Springer* (2014), 1–16.

[53] Saffron Huang, Divya Siddarth, Liane Lovitt, Thomas I Liao, Esin Durmus, Alex Tamkin, and Deep Ganguli. 2024. Collective constitutional ai: Aligning a language model with public input. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1395–1417.

[54] Lilly Irani. 2019. Chasing innovation: Making entrepreneurial citizens in modern India. In *Chasing Innovation*. Princeton University Press.

[55] Barbara A Israel, Amy J Schulz, Chris M Coombe, Edith A Parker, Angela G Reyes, Zachary Rowe, and Richard L Lichtenstein. 2019. Community-based participatory research. *Urban health* 272, 2 (2019), 272–282.

[56] Marius Jones and Peter Kalum Schou. 2023. Structuring the start-up: How coordination emerges in start-ups through learning sequencing. *Academy of Management Journal* 66, 3 (2023), 859–893.

[57] Robert Joppen, Sebastian von Enzberg, Jan Gundlach, Arno Kühn, and Roman Dumitrescu. 2019. Key performance indicators in the production of the future. *Procedia CIRP* 81 (2019), 759–764.

[58] Naveena Karusala, Sohini Upadhyay, Rajesh Veeraraghavan, and Krzysztof Z Gajos. 2024. Understanding Contestability on the Margins: Implications for the Design of Algorithmic Decision-making in Public Services. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–16.

[59] Daniel N Kluttz, Nitin Kohli, and Deirdre K Mulligan. 2022. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. In *Ethics of Data and Analytics*. Auerbach Publications, 420–428.

[60] Matthias Korn and Amy Voida. 2015. Creating friction: Infrastructuring civic engagement in everyday life. In *Proceedings of the fifth decennial aarhus conference on critical alternatives*. 145–156.

[61] Gideon Kunda. 2006. *Engineering culture: Control and commitment in a high-tech corporation*. Temple University Press.

[62] Eetu Kupiainen, Mika V Mäntylä, and Juha Itkonen. 2015. Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies. *Information and software technology* 62 (2015), 143–163.

[63] Michèle Lamont. 2012. Toward a comparative sociology of valuation and evaluation. *Annual review of sociology* 38, 1 (2012), 201–221.

[64] Karen Levy. 2022. *Data driven: Truckers, technology, and the new workplace surveillance*. Princeton University Press.

[65] Calvin A Liang, Sean A Munson, and Julie A Kientz. 2021. Embracing four tensions in human-computer interaction research with marginalized people. *ACM Transactions on Computer-Human Interaction (TOCHI)* 28, 2 (2021), 1–47.

[66] Henrietta Lyons, Eduardo Velloso, and Tim Miller. 2021. Conceptualising contestability: Perspectives on contesting algorithmic decisions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–25.

[67] Alexander Maedche, Achim Botzenhardt, and Ludwig Neer. 2012. Software for People: A Paradigm Change in the Software Industry. In *Software for People: Fundamentals, Trends and Best Practices*. Springer, 1–8.

[68] Likoebe M Maruping, Viswanath Venkatesh, and Ritu Agarwal. 2009. A control theory perspective on agile methodology use and changing user requirements. *Information systems research* 20, 3 (2009), 377–399.

[69] Gina Neff. 2015. *Venture labor: Work and the burden of risk in innovative industries*. MIT press.

[70] Australian Government Department of Industry Innovation and Science. [n. d.]. AI Ethics Framework. https://www.industry.gov.au/data-and-publications/building-australias-artifcial-intelligence-capability/ai-ethics-framework.

[71] Nelly Oudshoorn, Els Rommes, and Marcelle Stienstra. 2004. Configuring the user as everybody: Gender and design cultures in information and communication technologies. *Science, Technology, & Human Values* 29, 1 (2004), 30–63.

[72] Victoria Palacin, Matti Nelimarkka, Pedro Reynolds-Cuéllar, and Christoph Becker. 2020. The design of pseudo-participation. In *Proceedings of the 16th Participatory Design Conference 2020-Participation (s) Otherwise-Volume 2*. 40–44.

[73] Jennifer Pierre, Roderic Crooks, Morgan Currie, Britt Paris, and Irene Pasquetto. 2021. Getting Ourselves Together: Data-centered participatory design research & epistemic burden. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[74] Malvika Pillai, Ashley C Griffin, Clair A Kronk, and Terika McCall. 2023. Toward community-based natural language processing (CBNLP): cocreating with communities. *Journal of Medical Internet Research* 25 (2023), e48498.

[75] Viara Popova and Alexei Sharpanskykh. 2010. Modeling organizational performance indicators. *Information systems* 35, 4 (2010), 505–527.

[76] Miriam Posner. 2022. Agile and the long crisis of software. *Logic (s) Clouds* 16 (2022).

[77] Gary W Pritchard, Pam Briggs, John Vines, and Patrick Olivier. 2015. How to drive a London bus: measuring performance in a mobile and remote workplace. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1885–1894.

[78] Brian Randell. 1979. *Software engineering in 1968*. University of Newcastle upon Tyne, Computing Laboratory.

[79] David Redmiles, Hiroko Wilensky, Kristie Kosaka, and Rogerio De Paula. 2005. What ideal end users teach us about collaborative software. In *Proceedings of the 2005 ACM International Conference on Supporting Group Work*. 260–263.

[80] Protection Regulation. 2018. General data protection regulation. *Intouch* 25 (2018), 1–5.

[81] Microsoft Research. 2020. Societal Resilience. https://www.microsoft.com/en-us/research/group/societal-resilience/

[82] Karina Rider. 2022. Building ideal workplaces: Labor, affect, and identity in tech for good projects. *International Journal of Communication* 16 (2022), 18.

[83] Lazaros Sarigiannidis and Prodromos D Chatzoglou. 2014. Quality vs risk: An investigation of their relationship in software development projects. *International Journal of Project Management* 32, 6 (2014), 1073–1082.

[84] Christine Satchell and Paul Dourish. 2009. Beyond the user: use and non-use in HCI. In *Proceedings of the 21st annual conference of the Australian computer-human interaction special interest group: Design: Open 24/7*. 9–16.

[85] Nitin Sawhney and Anh-Ton Tran. 2020. Ecologies of contestation in participatory design. In *Proceedings of the 16th Participatory Design Conference 2020-Participation (s) Otherwise-Volume 1*. 172–181.

[86] Douglas Schuler and Aki Namioka. 1993. *Participatory design: Principles and practices*. CRC press.

[87] Wes Sharrock and Bob Anderson. 1994. The user as a scenic feature of the design space. *Design Studies* 15, 1 (1994), 5–18.

[88] Benjamin Shestakofsky. 2024. *Behind the startup: How venture capital shapes work, innovation, and inequality*. Univ of California Press.

[89] Siang-Ting Siew, Alvin W Yeo, and Tariq Zaman. 2013. Participatory action research in software development: indigenous knowledge management systems case study. In *International conference on human-computer interaction*. Springer, 470–479.

[90] Christo Sims. 2017. *Disruptive fixation: School reform and the pitfalls of techno-idealism*. Princeton University Press.

[91] Mona Sloane, Emanuel Moss, Olaitan Awomolo, and Laura Forlano. 2022. Participation is not a design fix for machine learning. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. 1–6.

[92] A Paul Spee and Efstathios Tapinos. 2012. The relation of key performance indicators and strategy development revisited. In *Academy of Management Proceedings*, Vol. 2012. Academy of Management Briarcliff Manor, NY 10510, 13240.

[93] Marilyn Strathern. 1997. 'Improving ratings': audit in the British University system. *European review* 5, 3 (1997), 305–321.

[94] Cella M Sum, Jiayin Zhi, Amil NT Cook, Patrick James Cooper, Arturo Lozano, TJ Johnson, Jason Perez, Rayid Ghani, Michael Skirpan, Motahhare Eslami, et al. 2025. " You're in a Ferrari. I'm Waiting for the Bus": Confronting Tensions in Community-University Partnerships. *Proceedings of the ACM on Human-Computer Interaction* 9, 2 (2025), 1–28.

[95] Harini Suresh, Rajiv Movva, Amelia Lee Dogan, Rahul Bhargava, Isadora Cruxen, Angeles Martinez Cuba, Guilia Taurino, Wonyoung So, and Catherine D'Ignazio. 2022. Towards intersectional feminist and participatory ML: A case study in supporting feminicide counterdata collection. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 667–678.

[96] Harini Suresh, Emily Tseng, Meg Young, Mary Gray, Emma Pierson, and Karen Levy. 2024. Participation in the age of foundation models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. 1609–1621.

[97] Udayan Tandon, Vera Khovanskaya, Enrique Arcilla, Mikaiil Haji Hussein, Peter Zschiesche, and Lilly Irani. 2022. Hostile ecologies: Navigating the barriers to community-led innovation. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–26.

[98] Emily Tseng, Meg Young, Marianne Aubin Le Quéré, Aimee Rinehart, and Harini Suresh. 2025. " Ownership, Not Just Happy Talk": Co-Designing a Participatory Large Language Model for Journalism. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. 3119–3130.

[99] Anna Lowenhaupt Tsing. 2004. *Friction: An ethnography of global connection*. Princeton University Press.

[100] Eve Tuck and K Wayne Yang. 2014. R-words: Refusing research. *Humanizing research: Decolonizing qualitative inquiry with youth and communities* 223, 2014 (2014), 248.

[101] Fred Turner. 2009. Burning Man at Google: a cultural infrastructure for new media production. *New Media & Society* 11, 1-2 (2009), 73–94.

[102] Kristen Vaccaro, Karrie Karahalios, Deirdre K Mulligan, Daniel Kluttz, and Tad Hirsch. 2019. Contestability in algorithmic systems. In *Companion Publication of the 2019 Conference on Computer Supported Cooperative Work and Social Computing*. 523–527.

[103] João Viera Magalhães and Nick Couldry. 2021. Giving by taking away: Big tech, data colonialism and the reconfiguration of social good. *International Journal of Communication* 15 (2021), 343–362.

[104] Jennifer Wang and Angela Moulden. 2021. AI trust score: A user-centered approach to building, designing, and measuring the success of intelligent workplace features. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[105] Susan C Whiston. 1996. Accountability through action research: Research methods for practitioners. *Journal of Counseling & Development* 74, 6 (1996), 616–623.

[106] David Gray Widder, Derrick Zhen, Laura Dabbish, and James Herbsleb. 2023. It's about power: What ethical concerns do software engineers have, and what do they (feel they can) do about them?. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 467–479.

[107] Ed Yong and Sabrina Tavernise. [n. d.]. 'The Interview': Ed Yong Wants to Show You the Hidden Reality of the World.

[108] Jonathan Zong and J Nathan Matias. 2024. Data refusal from below: A framework for understanding, evaluating, and envisioning refusal as design. *ACM Journal on Responsible Computing* 1, 1 (2024), 1–23.