Taking a HIT: Designing around Rejection, Mistrust, Risk, and Workers' Experiences in Amazon Mechanical Turk

Brian McInnis[1], Dan Cosley[1], Chaebong Nam[2], Gilly Leshed[1]
Information Science[1], Law School[2], Cornell University
{bjm277, drc44, cn277, gl87}@cornell.edu

ABSTRACT

Online crowd labor markets often address issues of risk and mistrust between employers and employees from the employers' perspective, but less often from that of employees. Based on 437 comments posted by crowd workers (Turkers) on the Amazon Mechanical Turk (AMT) participation agreement, we identified *work rejection* as a major risk that Turkers experience. Unfair rejections can result from poorly-designed tasks, unclear instructions, technical errors, and malicious Requesters. Because the AMT policy and platform provide little recourse to Turkers, they adopt strategies to minimize risk: avoiding new and known bad Requesters, sharing information with other Turkers, and choosing low-risk tasks. Through a series of ideas inspired by these findings—including notifying Turkers and Requesters of a broken task, returning rejected work to Turkers for repair, and providing collective dispute resolution mechanisms—we argue that making reducing risk and building trust a first-class design goal can lead to solutions that improve outcomes around rejected work for all parties in online labor markets.

Author Keywords
Crowdsourcing; trust; risk management; design; rejection

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

On its ten-year anniversary in November 2015, Amazon Mechanical Turk (AMT) continues to thrive as an effective online labor market, but one that raises concerns about worker welfare. A layer of technology separates Amazon's crowd workers (Turkers) from the Requesters for whom they complete work. This separation makes it possible for Requesters to coordinate large crowd workforces, but it also means that each transaction with a worker is mostly anonymous [38, 34, 30], abstract [1, 3], and legally ambiguous [4, 16, 41]. These conditions raise concerns about fairness [27] and abuse [44].

These concerns are exacerbated by AMT's *hands-off* approach to the labor market. AMT's participation agreement[1] classifies Turkers as independent contractors, free to accept any task they qualify for (§3b). At the same time, Requesters have the right to reject a Turker's completed work without payment (§3a) while AMT, providing only the venue for an exchange (§2), is not involved in resolving any labor disputes (§3f). When a Turker's work is rejected, the result is lost pay, time, and reputation, and AMT's stance gives workers little recourse. These policies, and other aspects of the AMT platform we detail below, make the practice of crowd working risky.

In this paper, we focus on how Turkers manage the risks of rejected work. Based on 1,092 comments collected during an experiment asking Turkers to comment on Turker-relevant aspects of the AMT participation agreement, we identified 437 that dealt with challenges, experiences, and practices around the risk of work rejection. Although respondents realize that some work is legitimately rejected, many rejections are seen as unfair. Problems with task clarity, design, and implementation can lead to rejections; many rejections include little rationale; some rejections seem arbitrary or malicious. No matter what the reason, Requesters are often non-responsive to Turkers who question the rejections—a position they can adopt because of AMT's hands-off policy. These aspects of rejection lead to feelings of unfairness, to mistrust in Requesters and AMT, and to perceptions of AMT work as risky.

This, in turn, leads workers to adopt strategies to minimize risk: avoiding new and known bad Requesters, sharing information about their experiences with other Turkers, and choosing tasks with clear, concrete descriptions and evaluation criteria. These risk-averse strategies, though rational given the current structure of the market, affect both the kinds of problems AMT can solve and the quality of living and learning Turkers can gain. This in turn harms the long-term prospects for individual workers, Requesters, and the market as a whole to grow and innovate toward the "Future of Crowd Work" envisioned by Kittur et al. [29].

Our contribution is twofold. First, we present an empirical analysis of how AMT's design and policies affect Turkers'

CHI'16, May 07-12, 2016, San Jose, CA, USA
© 2016 ACM. ISBN 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858539

[1] https://www.mturk.com/mturk/conditionsofuse

experiences, emphasizing risk and trust as key analytical constructs for characterizing these experiences. In this we highlight seven key risks that Turkers face around rejected work, along with a number of practices Turkers use to manage those risks. Second, we use this analysis to propose design ideas to reduce those risks and build trust. Realizing that immediate change by AMT itself is unlikely, we focus on designs that could be realistically prototyped either via existing sites where Turkers provide mutual aid such as TurkerNation and Turkopticon, or integrated into individual Requesters' task designs and validation workflows. We consider designs that could improve outcomes for both Turkers and Requesters in the short term, with the long-term goal of recasting the relation between Turkers and Requesters away from suspicion and toward their shared interests in the market.

RISK, TRUST, AND THE DESIGN OF THE AMT MARKET

Both Requesters and Turkers face risks in the AMT labor market. The mostly anonymous relationships in AMT present opportunities for mischief like employer fraud [4], worker carelessness [28], and collusion [11]. The psychological distance created between Turkers and Requesters by anonymity can also contribute toward feelings of mistrust, by dehumanizing the work relationship. For Turkers, this distance is perceived as demotivating and isolating [33], while for Requesters it is perceived as an excuse to forget that Turkers are real human beings who deserve fair labor practices [1, 3, 4, 16]. This distance also reduces the possibility for regular, predictable social interaction that can support cooperation [10, 25].

While there are a number of approaches across disciplines to defining, analyzing, and controlling risks, *trust* is a classic and crucial concept in risk management [2, 9, 14, 26]. Trust is defined as "a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another" [37, p. 395]. In labor contexts, it helps reduce the "uncontrollable complexity" of the current conditions in exchange for the future benefits of cooperation [26]. Thus, parties who trust each other are more tolerant of risks, while those with mistrust will be less tolerant.

Policy Asymmetries Breeds Risk and Mistrust

Trust and cooperation, however, are hard to develop in asymmetrical relationships [18], and the design of both the AMT policy environment and platform give more power to Requesters than Turkers. On the policy side, the employment status of crowd workers as independent contractors (AMT participation agreement §3b) is an important factor. Some aspects of this status benefit Turkers, for instance, allowing them to choose jobs rather than being assigned to jobs by Requesters. However, the agreement also gives Requesters the right to reject work without payment for any reason (§3a) while still retaining ownership rights to the rejected work (§3b). Further, when a Turker has a question or complaint, they can contact a Requester through the AMT site, but the Requester has no obligation to respond [27]. These aspects of the policy are structured for Requester power and Turker compliance in ways that our participants described as causing real hardship and mistrust of both Requesters and AMT.

Absent trust, Turkers and Requesters turn to strategies to protect against each other [39]. For instance, Requesters often deploy attention checks, elements of tasks that try to determine if a Turker is paying attention, to reduce the risk of sloppy or fraudulent work [28]. However, the additional work of attention checks imposes time costs on Turkers, as well as the potential for honest errors on nonessential parts of the task. Thus, Turkers might reduce these costs and risks by sharing with other Turkers ways to crack the checks [5]. As Turkers and Requesters come to see each other less as mutually beneficial partners and more as adversaries, trust likely declines, and perceptions of risk correspondingly increase.

Platform Creates Information Asymmetries

Like the participation agreement, the design of the platform is asymmetrical in favor of Requesters, who have privileged access to information and tools that Turkers don't have. Although Requesters cannot assign Turkers to tasks, they can screen Turkers using information about them such as past performance or demographics that the platform provides. Requesters can also ask Turkers to log into their Facebook account and use information gleaned from that to screen Turkers. The API also helps Requesters automate many of their interactions with Turkers; for Turkers, on the other hand, automated tools are explicitly forbidden (§3b.i).

Reputation features of the platform are also asymmetrical, meaning that anonymity in AMT affects Turkers differently than Requesters [27, 33, 39]. Turkers build reputations over time through successfully completing tasks, which in turn allow access to more and better paying tasks. New Turkers, or those who stumble initially, find that much of the market is closed to them. By comparison, a Requester with a bad reputation, or a new account, has all of the same AMT tools and resources as a good Requester, or as they would if they deleted their account and restarted with a fresh username. Further, AMT does not provide Turkers with information about a Requester's history, e.g., tenure, past acceptances and rejections, and pay rates. This means that a Requester is far less reliant on their reputation to access the AMT labor market, and Turkers have far less ability than Requesters to use reputation to reduce risks.

Turkers do have access to an AMT dashboard that summarizes the amount of money they have earned, their current approval and rejection rates, the approval status of jobs ("Human Intelligence Tasks", or HITs) they have completed, and the last 45 days of their work history. These work history details provide some information to Turkers about Requesters they have already worked with, but there is little analysis or aggregation. The platform also provides little information about open HITs; notably, important factors participants use for risk management such as time to completion or acceptance rate are nowhere to be found.

Turkers use Mutual Aid to Mitigate Risks

Without access within AMT to information and tools to manage risks, Turkers turn to external platforms where they provide and receive mutual aid toward making informed decisions about their labor market participation [34, 39]. There are several tools Turkers use to make informed decisions about whether to accept a HIT or not. The most widely adopted of these is Turkopticon [27], an online tool that Turkers use to share comments about and rate their experience with specific Requesters based on their communicativity, generosity, fairness, and promptness. There is also an accompanying browser-based plug-in that embeds a summary of the Requester's Turkopticon ratings next to HITs in the AMT interface.

Turkopticon is widely used; however, just as Requesters' attention checks impose time costs, it takes time to evaluate Requesters on Turkopticon. Thus, recent tools automate the collection of mutual aid data by harvesting performance information as Turkers take HITs. Like Turkopticon, Crowd-Workers [5] has a browser-based plug-in that presents Turkers with metrics about HITs including average hourly rate, expected time to payment, approval rates, and reasons for rejection. Unlike Turkopticon, these metrics are automatically collected by the plug-in. Similarly, the TurkBench [21] prototype automatically collects workers' activity to provide visualizations of the AMT market. Both Crowd-Workers and TurkBench are in development and much less commonly used than Turkopticon.[2]

Other external tools aim to support not just task choice, but to support a community of Turkers. For example, the online forum TurkerNation provides Turkers with professional and personal support. Turkers share tips and strategies for working on AMT and discuss how Turking is characterized in the media and by academics. Further, TurkerNation is a space for social support, where Turkers share 'prayers and good vibes' for members who live in challenging circumstances in which income from Turking is how they pay the bills. Martin et al. describe this support as the mutual aid work that makes Turking work [34].

Overall, these mutual aid tools have helped Turkers to avoid some risks, provide some solidarity and community, and exert some power to collectively regulate the market through the ability to avoid bad HITs and Requesters [27]. However, in the same way that AMT's policy and tools are one-sided in favoring Requesters, these mutual aid tools are primarily aimed at Turkers. In doing so, they help manage risks in the short term, but do little to support the management of risk in the market as a whole or building trust between Turkers and Requesters. One notable effort toward this was Dynamo, which was designed as a platform for Turkers to generate and carry through collective action. In its first six months, Turkers generated 22 ideas and initiated two campaigns through the platform. Notably, a campaign called "Guidelines for Academic Requesters" led to useful dialogue between Turkers and academics who use AMT, as well as a creation of best practices. Unfortunately, Dynamo has had limited impact, as it lost momentum as participants lost interest or left, frustrated by friction caused during heated debates [38].

In this context, we are interested in the extent to which design innovations could support the reduction of risk and the building of trust, given the information and power asymmetries and lack of trust that characterize AMT. Our short-term goal is to suggest specific design innovations for better risk management that could be prototyped at low cost in mutual aid forums or task designs. Our medium-term goal is for these design innovations to work toward increasing trust between Turkers, Requesters, and AMT, recasting their relationship to be more cooperative and less adversarial. Our long-term goal is to move toward policies and platforms for crowd work that explicitly address these questions of building trust and managing risk, not from the perspective of one side only, but for all parties in the market.

METHOD

To address these goals, we leverage data from a prior study in which Turkers were asked to comment on parts of the AMT participation agreement that directly affect their work and livelihood [35]. The original study focused on predicting commenting behaviors, but did not examine the content of those comments. However, we did notice in the comments a profound emphasis on rejection and risk that led to the analysis we describe below and, eventually, to the framing we arrived at above.

Data Collection Context

The data were collected through an online discussion website in which participants could comment on parts of the AMT participation agreement as of November 1, 2012. The website design was inspired by RegulationRoom, a tool designed to help lay people effectively participate in complex policy discussions [15]. In RegulationRoom, policies are translated into plain English and broken into subtopics with specific issues and questions to respond to. In the study, we did the same, creating eight subtopics focused on portions of the participation agreement that we saw as most relevant to Turkers (Table 1). Each subtopic was presented in a separate webpage with its own discussion thread. During the study, participants were free to navigate between the subtopics and post comments on each subtopic's discussion thread.

[2] Irani and Silberman [27] reported that Turkopticon had been installed over 7,000 times. At the time of writing Crowd-Workers had 389 users. The TurkBench [21] case study was based on feedback from 4 participants.

Subtopics	Comments about rejection	Total comments
a. Employment status of Turkers as independent contractors	41	228
b. Ownership rights to rejected work	136	150
c. Requesters' right to reject without compensation	167	208
d. How Turkers are paid	1	145
e. Payment delays	22	106
f. Tax law compliance	7	97
g. AMT's hands-off policy	60	80
h. Disclosure of personal information to Requesters	3	78
Total	437	1092

Table 1. Discussion subtopics, number of comments on each subtopic, and number of comments that discussed rejection.

In the end, 391 U.S.-based Turkers recruited through AMT contributed 1,092 comments.[3] 46% of participants identified as female and the average age was 36 years. 57% were employed full-time outside of AMT (the other 43% were students, retired, homemakers, or worked part time), and 78% had some higher education. 55% of the participants reported using AMT as a way to make some "extra money" or to supplement their primary source of income. 22% of participants reported spending between 9 and 14 hours per week working on AMT; 36% devote 14 hours or more.

Data Analysis

We employed standard qualitative analysis techniques. In the first stage, we reviewed all 1,092 comments. Using an affinity diagramming technique, we initially clustered comments to identify common patterns, created a preliminary list of the initial codes/themes, and looked for similarities and contradictions across the comment data. Through several successive loops of coding, discussion, and re-coding, we identified the following main themes: *flexibility, rejection, asymmetric rights, market accountability,* and *professional development*, with 30 specific codes under these themes. The finalized set of codes/themes was then applied to the entire dataset.

Our analysis revealed that *rejection* was a prominent recurring theme, with 437 of the comments addressing it. These comments spanned all areas of the participation agreement, especially around subtopics about ownership of rejected work, rejection without compensation, and AMT's hands-off policy (Table 1). As a result, in the second stage of the analysis we decided to focus on Turkers' experiences, opinions, coping strategies, and solution ideas around rejections.

We organized the comments to first present Turkers' experiences of rejection, then their strategies for managing rejections and the risks associated with it, and finally ideas they suggested to mitigate risk. In reporting on our findings, we support our analysis with quotes selected from Turkers' comments, expressed in their own language and voices to better convey their own experiences [40]. Each quote has a unique identifier linked to a specific participant. Identifiers range from 1–549, capturing all participants, not just those who left a comment.

EXPERIENCING REJECTION AND MANAGING RISKS

Not all rejections are problematic from the Turkers' point of view. Low quality work that was done poorly, inattentively, or maliciously is seen as fairly rejected: "a HIT rejection is reasonable when the work is done without reading directions or is not satisfactory at all" [P32], "…or is obviously scamming the Requester" [P5]. Turkers also realize "we're all humans, and we all make mistakes" [P239] and that sometimes they do bad work.

The real issue is *unfair* rejections that lead to lost wages, time, and future opportunities [1, 4, 16]: "This is a large problem for Turkers. I can do the work with honesty and integrity to get rejected offhand. Not only do I not get paid, but if I do a batch for a bad Requester, and all HITs are rejected, this can potentially do great harm to my ability to do quality HITs in the future, i.e., reduce my approval percentage significantly" [P245].

Below, we summarize our findings based on participants' comments, identifying seven main risk factors that lead to feelings of unfairness around rejection:

1. Flaws in task or interface design
2. Unclear evaluation criteria
3. Unresponsive, arbitrary resolution of rejections
4. Lack of information on Requesters
5. Inexperienced and Unfamiliar Requesters
6. Tasks with poor return
7. Prioritizing efficiency over quality

At a high level, some of our observations within each factor are called out in other work (such as being at the mercy of Requesters around rejected work [27, 34]). At a more nuanced level, our findings add depth to the analysis, focusing on perceptions, experiences, challenges, practices, and behaviors around risky HITs, Requesters and the market. Through Turkers' stories, we also identify the (negative) consequences for Turkers, Requesters, trust between them, and the market as a whole.

Risk Factor 1: Task and Interface Design Errors

Turkers express frustration about rejections caused by task or interface design mistakes, often because they only learn about these errors after putting in the work. "I once had a HIT rejected, stating I entered an 'old code' though I entered the one I got at the end of the survey" [P209]. HITs can behave in ways that Turkers don't expect: "I've had two rejections that were bogus and two that were a matter of the enter/return key submitting the HIT. So, all were from Requester errors" [P106]. Requesters can also forget

[3] Location was the only restriction to access the HIT.

to include task elements or incorporate them incorrectly, and later use them as quality checks: "Requester who made me do a survey and said, 'You did not follow the translation portion,' but there was no translation part" [P52].

Design errors are likely to affect many Turkers: "…Turns out this happened to a lot of people. If the Requester had to submit a formal complaint [before each rejection], AMT would see the problem was on the Requesters end, not the turkers involved" [P209]. Whether individual or wholesale, rejections caused by such design errors lead to feelings of frustration, unfairness, and mistrust.

Risk Factor 2: Unclear Evaluation Criteria

Compounding the problem is that Requesters can reject a task for any reason—and don't have to give one. As in prior work [27, 34], participants felt they deserve an explanation: "it's ridiculous when a Requester can reject work without giving a reason" [P48]. Beyond the frustration, Turkers also seek to improve and avoid being rejected in the next time: "If someone genuinely tries and gets rejected, how will they know exactly what they did wrong to correct it for future work?" [P44]

Further, there are no clear standards for task design and performance evaluation [23]. For Turkers, this means that the Requester might be using standards that seem malicious: "some Requesters like to scam people by making up reasons for not paying such as too few words or bad grammar" [P46] or arbitrary: "Requesters could be rejecting the HIT because they don't want to pay, citing some ambiguous reason like 'invalid' with no supporting documentation. The current system is set up in such a way that the entire burden lies with us Workers, as well as all risk[4]" [P140].

Both the inability to improve their own work quality and the ambiguous standards that Requesters use to evaluate the work are likely to contribute to Turkers' perceptions of risk and mistrust in Requesters.

Risk Factor 3: Unresponsive, Arbitrary Resolution

In principle, rejections can be disputed by communicating directly with the Requester. However, as Amazon has no policy about Requesters' responsibilities in communicating with Turkers and resolving disputes, power is in the hands of Requesters. Some Requesters are responsive: "I've only been rejected a few times. I've had at least one reversed after I did send a message to the Requester for clarification...If you are rejected, I strongly suggest contacting the Requester" [P250]. However, this positive experience was in the minority, compared to negative experiences related to unresponsive Requesters that our own respondents and other studies report [27, 34]. Turkers understand there may be reasons for a lack of response: "For some Requesters they need such a large volume of responses it would be unreasonable for the[m] to respond to each inquiry directly" [P199]. But that doesn't make the lack of response any less unpleasant.

The lack of an official dispute process overseen by AMT, and the concentration of power in the hands of Requesters, exacerbate the mistrust that results from erroneous or unexplained rejections.

Risk Factor 4: Lack of Information on Requesters

Looking at Requesters' history can help mitigate risks, and in fact many Turkers evaluate Requesters by studying their own work histories; they "go through all of [their] previous hits and check one by one" [P118] to see if the Requester approved and paid them or left feedback explaining why not. However, it is hard for Turkers to keep up with even their own history: "To expect us to keep up with every name of every Requester and every HIT title we've done is literally impossible. We don't have a simple way to access those records, and attempting to do so would be extremely time consuming" [P461]. This lack of support drives the use of tools such as Turkopticon and Crowd-Workers.com where Turkers share information about Requesters outside of AMT [5, 21, 27, 39].

This information is clearly useful. Positive interactions with Requesters signal that they can be trusted in the future: "I fell below the highest state minimum wage per hour, so they [the Requester] decided to send me the difference in a bonus payment. I made a note of who they were and I will definitely be working with them in the future!" [P264]. On the other hand, when Turkers have negative experiences, they "tend to avoid HITs from these Requesters" [P223]. Experienced Turkers regularly referenced Turkopticon as a way they could take back some control over bad Requesters. Even if a Turker has never personally worked with a specific Requester before, they can use this shared information to make a reasonable guess about what the work and relationship might be like: "…If you read Turkopticon and TurkerNation, you won't work for bad Requesters" [P77].

Whether individually or collectively, however, this tracking comes with costs. Much of this effort is imposed by the asymmetry of information available about Requesters and Turkers' reputations in AMT; knowing that this disparity exists likely harms trust between Turkers and Requesters. Mutual aid tools also often have an "us against them" character that might work against trust; Turkopticon's web page, for instance, emphasizes that "Turkopticon lets you REPORT and AVOID shady employers[5]".

[4] Note that this is not true; among other things, Requesters bear risks around low-quality work, collusion, and other possible Turker behaviors. We call this out here to point out that our data collection and focus is skewed toward Turkers, and will return to this point in the limitations section.

[5] https://turkopticon.ucsd.edu/

Risk Factor 5: Inexperienced and Unfamiliar Requesters

Shady employers are not the only risk Turkers face around Requesters. Managing a crowd workforce is not easy, and Turkers recognize that Requesters who are new to AMT pose a risk because of their unfamiliarity with the platform and its norms: "I have had work rejected because the Requester didn't know what he was doing, and even told me that in a response message. Yet, he did not reverse the rejection, which hurts me, and he got free data" [P95]. While mistakes happen, Turkers do not trust the system to help new Requesters learn how to correct their errors: "a new requester is ill informed on how to handle the situation with no alternative to rejection" [P24].

Another issue with new Requesters is that they may be old wine in new bottles. The participation agreement explicitly requires that Turkers may only register once with AMT (§3b), but this is not a requirement for Requesters. A Turker shared an experience of being blocked (i.e., banned from a Requester's HITs and put on risk of being removed from AMT) by a Requester they had never worked for: "Requesters can use many names, preventing workers from even knowing who actually issued the block" [P47]. Shady Requesters can also use new accounts to shed bad reputations they accumulate in outside forums: "Changing account names [by a Requester] seems too easy to avoid bad reviews" [P127]. Friedman and Resnick describe the costs around cooperation, reputation, and trust when it is possible to obtain "cheap pseudonyms", as Requesters can by opening new accounts [17].

To minimize these risks, Turkers tend to avoid requesters they are unfamiliar with, whether they are new or simply do not have reviews on outside forums like Turkopticon and TurkerNation. This reduces the ability for Turkers and new Requesters to build trusting relationships while hurting the market as a whole by limiting new entrants.

Risk Factor 6: Tasks with Poor Return

Turkers also manage risk by avoiding tasks that seem likely to have poor returns. For instance, because problems around task design and evaluation criteria described earlier often lead to rejections, Turkers seek HITs with straightforward acceptance standards that are easy to comply with: "I think that standards should be stated very clearly, as to not waste someone's time and so that they are able to complete a task the way the Requester wants them to" [P171].

Estimating the risk of HIT rejection is often weighed against the estimated expected return on a HIT based on what they can infer about the task from its description and from outside forums. Time to complete a task is important in estimating risk, because the stakes rise as tasks get longer: "if a HIT took 25 minutes and was rejected for a simple reason, that's pretty rough for the Turker" [P365]. Weighing the risk against a HIT's potential reward is important, because if a task is rejected then the Turker would have "waste[d] their time on that HIT...when they could have been doing other work that would have earned them wages" [P296].

Thus, Turkers tend to favor concrete, simple, short tasks over longer, complex and less clearly-defined ones, and as with avoiding new Requesters, this can harm both Turkers and Requesters. For Turkers, this can reduce variety, creativity and the opportunity to develop new skills, leading to longer-term risks around stagnation and boredom; for Requesters, it limits the space of potential task designs.

Risk Factor 7: Prioritizing Efficiency over Quality

Turkers also monitor their behavior *while* they work on a HIT to mitigate the risk of rejection. For example, many Requesters impose minimum effort criteria such as response length, and even though Turkers are expressly forbidden to use scripts and bots, they still use these tools to reduce the risk of rejection: "I had used a character counter and had purposely written 50 characters over the minimum to make sure I had done enough" [P145]. Such scripts help Turkers to gauge that their performance complies with task requirements and to determine when they can stop working on a HIT and move onto the next.

As with HIT selection, Turkers consider cost-benefit tradeoffs during HIT execution. If working on the HIT is inefficient, with the work taking too much time relative to the reward, Turkers will sometimes decide to stop despite the wasted work: "I know that I have personally abandoned HITs that have taken longer than I'm willing to spend for the pay, and that's a loss to me of both time and money" [P151].

This optimization behavior leads some Turkers to think less about doing quality work and focus more on developing skills to maximize their AMT return while minimizing their time and effort on HITs. "Many Turkers put in hours of work, and much of that work is research into avoiding unfair rejections because we can do nothing about it. It's bad for us, and bad for honest requesters who could get more high quality work done, IMO" [P342]. In other words, Turkers bear the costs of developing efficient work skills, and Requesters bear the costs of work done with minimum effort just to avoid rejection. In the long term, the market as a whole suffers from failing to achieve growth, innovation, complexity, and collaboration [29].

WHAT CAN BE DONE?

Our analysis of how Turkers' described their experiences in the context of the AMT participation agreement positions risk as a central construct in Turkers' work. In this section we outline a series of ideas, many directly inspired by Turkers' suggestions, to help Turkers avoid risky HITs, to minimize the costs of rejections, and in the long term to promote trust between Turkers and Requesters. The current lack of trust in the market and the inertia behind AMT lead us to believe changes to AMT itself are unlikely in the short term. Thus, we focus on ideas that researchers, Requesters, and Turkers could explore using individual task designs, existing mutual aid forums, browser plugins, or new external sites, rather than ideas that would require fundamental changes to the design of AMT.

Recognizing and Avoiding Risky HITs

Sounding an Alarm for Broken HITs

- *Primary risks*: Task and interface design errors (1); inexperienced Requesters (5)
- *Suggestion*: Single-click alarm and an appeal process

Turkers are frustrated by rejections that are due to problems with the task design or technical errors. Some proposed a formal appeal process, for example, a single-click device that submits an appeal to a Requester to reverse the rejection: "there should be some type of appeal button that makes the requester know of this. If they see a large number then they may realize an error has been made" [P178]. This idea suggests that such a tool might also serve as a *broken HIT alarm*, immediately notifying the Requester of a potential error in the HIT.

Our own experience supports this idea. At one point in our data collection process, we introduced a JavaScript bug that caused a technical error. Within minutes of launching the HIT, 36 Turkers emailed us about it and our Turkopticon ratings plummeted. By quickly turning off the HIT and responding to each Turker, our Turkopticon ratings recovered and ended the day higher than before. Thus, although in the short term as a Requester we did incur costs by handling these emails, the long term benefit was likely worth it. Further, HITs where Requesters are actively monitoring problems are also likely to be perceived as less risky, as Turkers value responsive Requesters.

This process could be streamlined away from individual email responses in a number of ways. Current email programs could be used to group emails sent by AMT per-HIT based on the subjects of the emails, and Requesters (or, crowd workers via tools like MailValet [31]) could monitor those folders. Or, rather than processing unstructured AMT-generated emails, task interfaces could include a problem report button that had a pre-set list of common types of problems and a freeform text field to give additional information. These could feed into a bug report-style database that would let Requesters monitor problems and respond in the aggregate. This would also address an important API problem in AMT: Requesters can't communicate with Turkers who haven't completed a HIT, making it hard to repair these kinds of technical problems.

Integrate Mutual Aid Systems to Increase the Visibility of Risk

- *Primary risks*: Bad Requesters (4); tasks with poor return (6)
- *Suggestion*: Create data sharing standards across the network of Turker tools

A number of tools have been developed to help Turkers share their activity to provide mutual aid. However, uptake of newer, more automated tools such as Crowd-Workers and TurkBench has been slow compared to the more established Turkopticon, even though Turkopticon requires manual rating of Requesters and does not support sharing HIT-level information that participants described as important in making real-time work decisions. Although part of this might be because Turkers might fear automated sharing of performance data, we suspect that much of Turkopticon's use comes from its established reputation, coupled with the cost of changing practices or using parallel systems with partly overlapping functionality.

What if Turkopticon, Crowd-Workers, TurkBench, and other mutual aid tools joined forces on standards and APIs for sharing data? Giving new tools access to existing data could help provide useful services more quickly, increase the value of adopting them for individual Turkers. These new tools, in turn, might provide additional kinds or sources of data that existing tools could make use of.

For instance, Crowd-Workers might use the Requester-level average ratings available from Turkopticon's current API to help users make decisions about HITs that not enough Crowd-Workers users have attempted to have good estimates of return—or even use those ratings as input into those estimates. Turkopticon, in turn, might be able to aggregate HIT performance data from Crowd-Workers to make comparisons between Requesters to supplement manual ratings. Meanwhile, forums might automatically annotate references to Requesters and HITs with data available from these services.

Such cross-platform development and data management can be tricky to coordinate; further, individual tools might prefer to compete instead of cooperate. But many such tools are developed by academics with an avowed interest in making Turkers' lives better. Serious thought to how these tools can interoperate would be a promising way to do this, and support the kind of longer-term questions about the market proposed by Silberman et al. [39].

There are risks to such an approach—Amazon could adjust the participation agreement or AMT interface to prevent the use of such tools, the tools and mutual aid data might become a target for people looking to manipulate Turker behavior [38], or they might exacerbate the tendency to take only low-risk HITs. But it aligns well enough with current practices and challenges that it is worth exploring, and our hope is that it would increase overall trust by helping Turkers manage their risks and allocate trust appropriately: trusting Requesters and HITs that are good actors, and avoiding those that are not. This, in turn, would empower mutual aid tools to better regulate the AMT market.

Mitigating the Impact of Rejection

Fast Fail to Reduce the Impact of Honest Error

- *Primary risks*: Unclear evaluation criteria (2); efficiency over quality (7)
- *Suggestion*: Automated feedback within the task design

Developing standards for task design is an ongoing conversation within HCI crowdsourcing research [7, 23, 29, 36]. As code repositories and task-templating services are developed, our findings suggest that adding the Turker perspective on task design could be used to soften the impact of rejection. For instance, the earlier idea about a feature for reporting broken HITs from inside the task could be made part of such libraries.

A related suggestion from Turkers was that task designs could evaluate their work in situ and help them make better real time decisions about continuing the task. For instance, "if in a survey, a Turker fails an attention check, the survey should be programmed to detect the error, and terminate the survey. In this case, the Turker can return the HIT, and have no damage to their approval rating" [P80]. Scripts for checking attention, instruction compliance, effort (e.g., text length) already exist for making acceptance or rejection decisions, suggesting that on-the-fly detection of many kinds of errors should be relatively easy to implement in existing workflows.

Catching and acting on errors in real time could protect honest Turkers from honest error, while more efficiently punishing bad actors. In some cases—new Turkers, or Turkers who haven't failed for this particular HIT or Requester before—the right resolution might be to suggest that the Turker abandon the HIT and tell them why, to reduce wasted time and to avoid damage to their reputation. For cases where the Turker is more likely to be a true bad actor—has a history of rejections, or fails multiple tasks in short order—the right answer might be to terminate the HIT and reject it immediately in order to (justly) damage the Turker's reputation.

By reducing wasted time, providing immediate feedback, and not collecting unpaid work products, fast fail options would likely improve honest Turkers' perceptions of Requesters while also reducing the number of complaints Requesters receive. There are risks—fast-failing attention checks that rely on agreement with others' responses might provide dishonest Turkers with information about which items are used to determine acceptance—but, just as with the cost-benefit tradeoff Turkers experience, Requesters need to consider the tradeoff between regulating dishonest workers and building relationships with honest ones.

Repairing Rejected Work

- *Primary risks*: Unresponsive resolution (3); efficiency over quality (7)
- *Suggestion*: Learning through repair

Turkers care about the quality of their work and, once a Turker's HIT has been rejected, the experience often leads to feelings of frustration. Turkers expressed the desire for a second chance, wanting to redo or improve their work and correct their mistakes to reverse a rejection: "I think if it seems the worker intentionally didn't do the work right or was just not paying attention, it is ok to reject. However if it seems that the worker was doing his/her job but a few honest mistakes were made, the worker should get a chance to correct it" [P79].

The value of repair to a Requester would likely depend on the task (e.g., its complexity, creativity). Further, with the AMT tools currently available to Requesters, providing an opportunity to repair work is technically challenging. First, a Requester would set up a new HIT just for the Turker to repair the work (which would need a special Qualification granting access to only that Turker). Second, the Requester would wait for the Turker to select the HIT and complete it. Third, the Requester would evaluate the new HIT and, if acceptable, revoke the initial rejection. Costly, though perhaps such a workflow could be made a template in task design libraries.

One alternative might be to use the fast-fail mechanisms described earlier to avoid the explicit processing of rejections and to train Turkers by trial and error. For instance, a rejected photo edit might be returned to the Turker with critical feedback and potential resources, challenging the Turker to repair the work or abandon the task. Such resources could even be designed into HITs, as with the tutorials Dontcheva et al. developed [12] and mechanisms for self-review and peer-feedback inspired by Dow et al. [13, 43]. The question of who bears the cost of such training is thorny, although Kittur et al. point out that providing opportunities for advancement is a key long-term issue for crowd work systems going forward [29].

Establishing Long-Term Relationships and Trust

The suggestions so far are mostly tactical, managing specific challenges around the selection and execution of HITs, although we expect that reducing those frictions in Turkers' work will have a positive impact on how Turkers see Requesters (and vice versa). In this section we introduce two ideas aimed more directly at building better relationships, and consequently increasing trust, between the parties.

Design Support for Collective Interaction with Requesters

- *Primary risks*: Design errors (1); unclear standards (2); unresponsive resolution (3)
- *Suggestion*: Collective interaction mechanisms

Recognizing the infeasibility of individual Turker-Requester contact via email (also discussed by [27]), one solution is to develop technology for a collective resolution system, where a crowd can more effectively communicate with a Requester or with AMT. We see this as a generalization of the "broken HIT button" described earlier: when a problem or dispute arises that affects a number of Turkers in the same way for the same Requester or HIT, it would be useful—but currently impossible in AMT's design—for that group to have a conversation with the Requester about the problem.

As a proof of concept, we might create a discussion space inside an existing mutual aid forum where Requesters are relatively welcome.[7] For instance, the subreddit *HITsWorthTurkingFor* is a regularly updated list of links to "good paying tasks" on AMT. For each HIT, a discussion thread could be created and Turkers could use the post to pose and vote on questions or concerns about the task for the Requester. Unlike email, the nested structure of a subreddit's comments could allow a Requester to quickly spot high priority concerns that attract many sub-comments and votes, as well as reading down the thread to understand the range of other concerns. The Requester could then respond to a whole thread, rather than to individual emails, saving them time while increasing the chances that Turkers see them as responsive, responsible, and trustworthy.

The Turker Task Design Collective (Speculative)

- *Risk Factors*: Design errors (1); unclear task and evaluation criteria (2); inexperienced Requesters (5)
- *Suggestion*: Task design support from Turkers

Ipeirotis and Horton discuss the Requester burden to design a good task, arguing that better task design standards could make crowdsourcing more scalable and effective [23]. New and complex task designs are likely to have problems with design or evaluation. Further, new Requesters are also more likely to inadvertently include errors in their task designs than experienced Requesters. Kittur et al. suggest that better task design standards and support for Requesters should come from the labor market platform [29]. We see this as unlikely in the short term—and risky in the long term because as constituted both the policy and platform favor Requesters over Turkers.

We propose revisiting the idea of Turker support for task design, but with new infrastructure and incentives. *Turkomatic* proposed that Turkers could be used to construct complex crowd task workflows [32], harnessing a crowd to identify, define, and price subtasks as well as to write their instructions. The experiment suffered numerous problems, but by recruiting skilled Turkers to collaborate directly with a Requester, Turkomatic generated complex crowdsourcing designs with little pre-planning.

Some of the problems faced by Turkomatic might be resolved by an infrastructure for cataloguing and refining task design "best practices" and standards. This way the crowd could integrate and update tested frameworks and commit new designs to the catalogue as a Requester identifies them. Turkers already spend time learning about tasks and critiquing their design, but this unpaid design work takes place on worker forums and is in part adversarial research [5, 34].

We propose that experienced Turkers could manage and profit from such a catalogue, benefitting both Turkers by developing error-free task design patterns that are likely to be legible to workers and Requesters by providing tested designs that reduce the amount of unstructured communication around rejections. Further, Turkers might develop new skills and talents as they take on new roles: consultants to understand Requester needs, brainstorming teams to think through novel tasks, and implementers who access the catalogue and assemble or even manage workflows for Requesters.

This is, admittedly, a speculative vision that would require Turkers and Requesters working more closely together than either the present platform or prevailing attitudes between Turkers and Requesters will support. Still, there are pockets of collaboration such as the academic Requesters discussion on Dynamo that could be used to prove a concept such as this. More generally, we argue that addressing Turkers' experiences of risky work and mistrust in other parties is a must, and to do this, we need to consider designs that bring Turkers and Requesters together and align their interests.

LIMITATIONS AND FUTURE WORK

Our study was intentionally limited to Turkers as an interest group and we designed the policy summaries to elicit their own experiences. This decision limits our ability to discuss Requester or AMT perceptions of the market. Further, our data may be biased toward particularly unhappy Turkers; in our prior study [35], Turkers who had higher feelings of trust and fairness for AMT tended to leave shorter comments that were less responsive to the policy topics. The distribution for trust and fairness ratings also skewed to the low end, with 90% of the participants rating their trust in AMT at or below the midpoint of a 5-point scale.

Thus, one clear path for future work is to include these missing voices. For instance, in our own experience, we didn't like issuing rejections, observing our Turkopticon ratings fluctuate, or seeing our email inboxes explode in response to honest mistakes we made in task design. We suspect that other Requesters have had similar experiences, and although the power and information balances in AMT lead researchers to concentrate their attention on Turkers, there is likely good work to be done around understanding the Requester side of the story.

A logical next step for both research and practice is bringing the voices of Turkers and Requesters together. The platform's layers of separation between Requesters and Turkers means that there is little direct communication and most of that is both task-focused and likely negative: rejections, disputes, complaints. Holding focus groups or online deliberations aimed at helping these parties discuss the issues might help them gain more perspective on the problems others face, identify points of mutual interest, and build trust. Observing these interactions would also be valuable for scholars interested in understanding and designing for the AMT market. For example, as crowd work groups and team-based HITs become a standard part of new online labor markets [36], Turkers and Requesters

[7] This is not always the case; in our own experience, when we attempted to talk with Turkers at the TurkerNation IRC channel, several Turkers exited as we entered, chased off by a "*big bad Requester*".

might come together to design playful small group activities, like brainstorming within a HIT—as the experience could strengthen their effectiveness as a team on the *next* HIT [42].

The collection of performance statistics and the dialogues suggested above would also enable research around Turkers' belief that they help regulate the AMT labor market by coordinating their individual actions and sharing information through mutual aid tools. Knowing how Requesters perceive this pressure and how it affects their decision-making around AMT would be valuable. Likewise, performance data would allow us to investigate our suspicions that risk-averse behaviors in fact impact the market, making tricky, novel, and time-consuming tasks less likely to be adopted. We also worry about other unintended consequences of mutual aid tools. It would be sad if mutual aid in managing risks around bad Requesters, for instance, led to a situation where although serious structural problems remain, the market is "good enough" to keep Turkers participating—but stagnant.

Finally, although we were careful to think about feasibility, the designs we propose impose some cost: on Requesters for developing task designs that integrate fast failing or problem reporting; on Turkers for agreeing to share performance data; on mutual aid forums for collecting and aggregating the data; on Turkers for reporting and Requesters for responding to problems. It may be that even relatively low-cost solutions are impossible to implement because the benefit to any particular actor is not high enough, or because trust is too low.

CONCLUSION

We see this work as making two primary contributions: First, we contribute an empirical analysis of Turkers' experiences in the market and challenges they face, emphasizing risk and trust as key analytical constructs for analyzing these experiences. Rejected work in AMT bears significant losses for the worker: loss of pay, time, and ownership of the completed work, along with lowered approval ratings that limit their access to future tasks. Our findings present the daily challenges that Turkers experience in identifying and responding to the risk around rejected work, through the lens of reading and commenting on the AMT participation agreement. Much of how Turkers select and perform tasks is focused on minimizing the risk of rejection, especially rejections they perceive as unfair; our findings suggest that both their experiences and the tools they use also likely reduce the trust they feel toward Requesters as a whole.

Our second contribution is to consider a number of relatively feasible design suggestions to improve Turkers' conditions. Many of these ideas, such as fast failing and providing in-situ feedback, involve designing tasks to help Turkers do better work or minimize wasted work, reducing the risk of rejections associated with new, complex, or badly designed tasks. Others, such as problem reporting and collective interaction, aim at improving communication between Turkers and Requesters, reducing the risk of unresponsive or arbitrary resolution of rejections while supporting more effective task designs.

Our hope is that by reducing risks in the short term and building trust in the medium term, both Turking itself and the relation between Turkers and Requesters can focus less on suspicion and risk and more on shared interest and satisfying outcomes for all parties.

ACKNOWLEDGEMENT

We thank all the Turker participants, whose thoughtful comments inspired this research. Thanks also to Malte Jung and the Cornell eRulemaking Initiative (CeRI) team for all their help in the project. This work is supported by NSF IIS 0910664, IIS 1422484, IIS 1405634, and HCC 1314778.

REFERENCES

1. Gilles Adda and Joseph J. Mariani. 2014. *Crowdsourcing for Speech: Economic, Legal and Ethical Analysis*. HAL Archives No. HAL-01067110.

2. Pat Barrett. 2014. New development: Risk management-how to regain trust and confidence in government. *Public Money & Management, 34*(6), 459-464.

3. Benjamin B. Bederson and Alexander J. Quinn. 2011. Web Workers, Unite! Addressing challenges of online laborers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 97-106.

4. Lisa J. Bernt. 2014. Suppressing the mischief: New work, old problems. *North Eastern University Law Journal*, *6*, 2: 311-346.

5. Chris Callison-Burch. 2014. Crowd-Workers: Aggregating information across Turkers to help them find higher paying work. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing* (HCOMP '14), Works in Progress and Demonstration Abstracts, 8-9.

6. Jesse Chandler, Pam Mueller and Gabriele Paolacci. 2014. Nonnaïveté among Amazon Mechanical Turk Workers: Consequences and solutions for behavioral researchers. *Behavioral Research* 46, 1: 112–30.

7. Justin Cheng, Jaime Teevan, Shamsi T. Iqbal and Michael S. Bernstein. 2015. Break It Down: A comparison of macro- and microtasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 4061-4064.

8. Lydia B. Chilton, John J. Horton, Robert C. Miller and Shiri Azenkot. 2010. Task search in a human computation market. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (HCOMP '10), 1-9.

9. Stacie Conchie and Calvin Burns. 2008. Trust and risk communication in high-risk organizations: A test of principles from social risk research, *Journal of Risk Research,* 28, 1: 141-149.

10. Karen Cook, Russell Hardin and Margaret Levi. 2005. *Cooperation without Trust?* Russell Sage Foundation Series on Trust. New York, NY: Russell Sage Foundation.

11. Djellel Eddine Difallah, Gianluca Demartini and Philippe Cudré-Mauroux. 2012. Mechanical Cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *CrowdSearch 2012 workshop*. Lyon, France.

12. Mira Dontcheva, Robert Morris, Joel Brandt and Elizabeth Gerber. 2014. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the Computer Supported Cooperative Work Conference* (CSCW '14), 3379-3388.

13. Steven Dow, Anand Kulkarni, Scott Klemmer and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '12), 1013-1022.

14. Timothy C. Earle. 2010. Trust in risk management: A model-based review of empirical research. *Risk Analysis: An International Journal,* 30, 4: 541-574.

15. Cynthia R Farina, Dmitry Epstein, Josiah Heidt and Mary J Newhart. 2014. Designing an online civic engagement platform: Balancing more vs. better participation in complex public policymaking. *International Journal of E-Politics* (IJEP) 5, 1, 16–40.

16. Alek Felstiner. 2011. Working the crowd: Employment and labor law in the crowdsourcing industry. *Berkeley Journal of Employment and Labor Law*, 32*,* 1.

17. Eric J. Friedman and Paul Resnick. 2000. The social cost of cheap pseudonyms. *Journal of Economics & Management Strategy*, 10, 2: 173-199.

18. Mark Granovetter. 2005. The impact of social structure on economic outcomes. *The Journal of Economic Perspectives*, 19, 1, 33-50.

19. Neha Gupta, Andy Crabtree, Tom Rodden, David Martin and Jacki O'Neill. 2014a. Understanding Indian crowdworkers. In *Proceedings of the Computer Supported Cooperative Work Conference* (CSCW '14).

20. Neha Gupta, David Martin and Jacki O'Niel. 2014b. Turk-life in India. In *Proceedings of the International Conference on Supporting Group Work* (GROUP '14).

21. Benjamin V. Hanrahan, Jutta K.Willamowski, Saiganesh Swaminathan, David B. Martin. 2015. TurkBench: Rendering the market for Turkers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15).

22. Panagiotis G. Ipeirotis. 2010. Analyzing the Amazon Mechanical Turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17, 2, 16-21.

23. Panagiotis G. Ipeirotis and John J. Horton. 2012. The need for standardization in crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11).

24. Langdon Winner. 1999. Do artifacts have politics? In *The Social Shaping of Technology* (2nd. ed.), Donald MacKenzie and Judy Wajcman (eds.). Open University Press, Buckingham, UK, 28-40.

25. Edward Lawler. 2013. Being on the edge of chaos: Social psychology and the problem of social order. *Contemporary Sociology: A Journal of Reviews* 42, 1: 340.

26. Niklas Luhmann. 1998. Familiarity, confidence, and trust: Problems and alternatives. In D. Gambetta (ed), *Trust: Making and breaking cooperative relations* (pp.94-107), Oxford: Basil Blackwell.

27. Lilly C. Irani and M. Six. Silberman. 2013. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 611-620.

28. Aniket Kittur, Ed Chi and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 453-456.

29. Aniket Kittur, Jeffrey Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matthew Lease and John Horton. 2013. The future of crowd work. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '12), 1301-1318.

30. Jörn Klinger and Matthew Lease. 2011. Enabling trust in crowd labor relations through identity sharing. In *Proceedings of the American Society for Information Science and Technology* (ASIST '11).

31. Nicolas Kokalis, Thomas Köhn, Carl Pfeiffer, Dima Chornyi, Michael S. Bernstein and Scott R. Klemmer. 2013. EmailValet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '13), 1291-1300.

32. Anand Kulkarni, Matthew Can and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with Turkomatic. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '12), 1003-1012.

33. Jennifer Marlow and Laura A. Dabbish. 2014. Who's the boss?: Requester transparency and motivation in a microtask marketplace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 2533-2538.

34. David Martin, Benjamin V Hanrahan, Jacki O'Neill and Neha Gupta. 2014. Being a Turker. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '14), 224-235.

35. Brian McInnis, Elizabeth Murnane, Dmitry Epstein, Dan Cosley, and Gilly Leshed. To appear. One and done: Factors affecting one-time contributors to ad-hoc online communities. To appear in *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '16).

36. Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine and Michael S. Bernstein. 2014. Expert crowdsourcing with Flash Teams. In *Proceedings of the ACM symposium on User interface software and technology* (UIST '14), 75-85.

37. Denise M. Rousseau, Sim B. Sitkin, Ronald S. Burt and Colin Camerer. 1998. Not so different after all: A cross-discipline view of trust. *The Academy of Management Review,* 23, 3: 393-404.

38. Niloufar Salehi, Lilly C. Irani, Michael S. Bernstein, Ali Al Khatib, Eva Ogbe, Kristy Milland and Clickhappier. 2014. WeAreDynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 1621-1630.

39. M. Six Silberman, Joel Ross, Lilly Irani and Bill Tomlinson. 2010. Seller problems in human computation markets. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (HCOMP '10), 18-21.

40. Robert E. Stake. 1995. *The art of case study research*: Sage Publications, Inc.

41. Stephen Wolfson and Matthew Lease. 2011. Look before you leap: Legal pitfalls of crowdsourcing. In *Proceedings of the American Society for Information Science and Technology* (ASIST '11).

42. Robert I. Sutton and Andrew Hargadon. 1996. Brainstorming groups in context: Effectiveness in a production design firm. *Administrative Science Quarterly,* 41, 4: pp. 685-718.

43. Anbang Xu, Shih Huang and Brian Bailey. 2014. Voyant: Generating structured feedback on visual designs using a crowd of non-experts. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '14), 1433-1444.

44. Jonathan Zittrain. 2009. The future of the Internet and how to stop it. *Yale University Press*.