Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2014)

# Wish: Amplifying Creative Ability with Expert Crowds

**Anand Kulkarni[1], Prayag Narula[1], David Rolnitzky[1], Nathan Kontny[2]**
[1] MobileWorks, Inc., [2] Draft
[1] {anand | prayag | dave}@mobileworks.com, [2] nate.kontny@gmail.com

**Abstract**

We present *Wish*, a system that uses expert members of an online crowd to amplify a user's ability to carry out complex creative tasks. When presented with an incomplete or rough draft of creative content (for example, writing, programming, or art), Wish finds and recruits an expert to suggest a possible realization of the user's vision.

Wish contributes a new approach to crowdsourcing complex work. Rather than combining pools of unskilled workers to producing complex output, Wish uses the crowd to identify and recruit a pool of experts, then assigns a single high-quality expert to carry out the task. This approach retains the benefits of using a crowd – scalability, speed, correctness, and responsiveness – while simplifying the process of crowdsourcing complex work.

We demonstrate Wish in the context of three prototype tools. *Draft* enables users to consult a crowd of authors for suggestions on writing. *Hack* lets developers convert pseudocode into working code by pulling results on demand from a crowd of programmers. *Sketch* lets individuals convert rough sketches into fully-refined art by consulting artists. We illustrate how novice creators can use Wish to amplify their ability, how expert designers can use Wish to explore design spaces and improve the speed of creation, and how new users can use Wish to gain feedback from experts.

## Introduction

Since the first days of human-computer interaction, software has been viewed as a tool to amplify a user's ability to produce complex creative output (Licklider 1960, Bush 1945). However, many efforts in human computation today focus on improving the intelligence of software, not of humans (Von Ahn and Dabbish 2004). In this paper, we explore whether crowd computing can be used to make humans effectively smarter and more creative. How can crowd computing be used to improve our ability to carry out specialized, creative work?



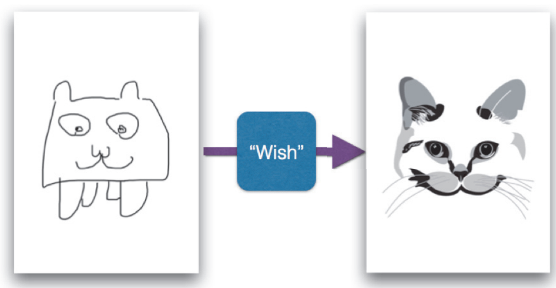


*Figure 1. Wish augments design interfaces with the ability to obtain refinements from experts. A nonexpert user can convert a napkin sketch into a refined illustration by clicking a button.*

Crowd labor on microwork platforms is not well suited to this purpose. It has historically been used to solicit unskilled workers for brief periods of time, using workflows, redundancy, voting and the wisdom of crowds to attempt to produce complex work from raw human brainpower. These approaches have been relatively lacking: while structured and repetitive work is easy to achieve on an open undifferentiated marketplace, both the precise workflows for decomposition of complex work and the workers best suited for a task have proven hard to identify.

To produce this result, Wish identifies and recruits *one* member of a pool of expert crowd workers from within a general crowd to participate in the creative process and complete the user's work. If no such expert is available, one is dynamically recruited into the crowd. We demonstrate multiple examples of complex creative output generated using the system, including transformation crude sketches into refined visual art, high-level outlines into personalized blog posts, and function comments into working code.

Behind the scenes, Wish uses an expert-identification algorithm based on *query incentive networks* to rapidly identify an expert contributor in the crowd and solicit results. This is a departure from traditional approaches to crowd computing; rather than coordinating large numbers of untrusted contributors to create complex content, Wish focuses on finding a single high-functioning individual to solve the task in an unconstrained fashion. The role of the crowd is to search for and find the most capable and competent member, rather than to combine efforts of the crowd to decompose and verify results. This approach provides a new strategy for crowdsourcing complex output that may be applicable to areas beyond creative production.

## Related Work

Historically, crowd work has emphasized the use of non-specialized labor pulled from platforms such as Amazon Mechanical Turk, making use of workflows to combine results from undifferentiated workers (Little, et al. 2010). Non-microtask outsourcing marketplaces like oDesk and Freelancer allow end users to individually interview and vet potential contractors, but leave the problems of expert identification and recruitment to the end user and require direct, 1-1 interaction with experts over the course of engagement (oDesk). More recent work has focused on the potential to use oDesk in an automated manner to producing coordinated expert content over the course of days (Retelny, et al. 2013) ; these approaches use preconstructed

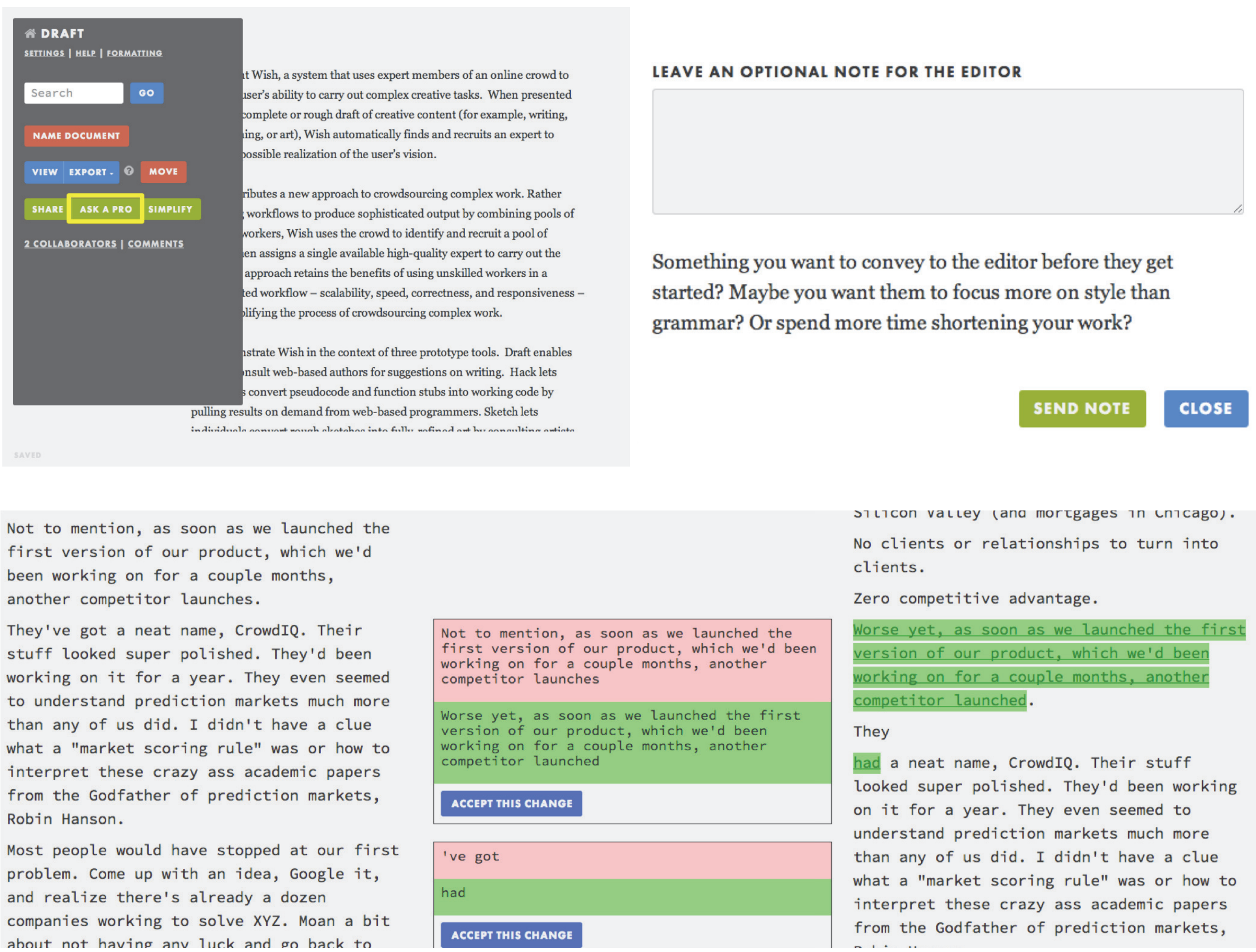


Figure 2: Wish uses an online collaborative editor with version control functionality. In Draft (above), a text editor is augmented as shown to allow a user to request, review and approve or reject edits sourced from the crowd. When submitting a Wish, users may provide an optional set of instructions on how they would like their Wish fulfilled.

teams based on oDesk's taxonomy as opposed to dynamically selected and identified pools. Efforts to produce complex and sophisticated work from undifferentiated workers can succeed in very narrow contexts but identify significant challenges in workflow design and worker ability (Kulkarni, Can, and Hartmann 2012).

Many tools have incorporated non-expert microwork crowds for improving the productivity or ability of the end user, though not the explicit purpose of amplifying creativity (Zhang, Lai, and Bächer 2012; Bigham et al. 2010; Bernstein et al. 2011)]. These tools typically focus on the use of workflows to verify output from unskilled workers.

Similarly, there is a rich body of work indicating the value of design feedback and design exploration as a mechanism for improving user performance during creative tasks (Dow et al. 2012). Related work explores the notion of how unspecialized members of online crowds can provide useful design feedback to novices as a means for education and improvement, as well as to other members of the crowd (Dow, Gerber, and Wong 2013; Dow et al. 2012).

Ongoing work pursues a parallel line of inquiry into the use of embedded human experts to carry out software development; these approaches assume fixed expert resources are known and available (LaToza 2013; Little 2012).

On a theoretical basis, the form of search used in Wish is best modeled with the framework of Query Incentive Networks (Kleinberg and Raghavan 2005), which are well-understood from a game-theoretical perspective. Among other examples, MIT's winning entry to DARPA's Red Balloon Challenge (Pickard et al. 2011) illustrates the power of this form of network search for rapidly identifying individuals with appropriate information or skills in a network.

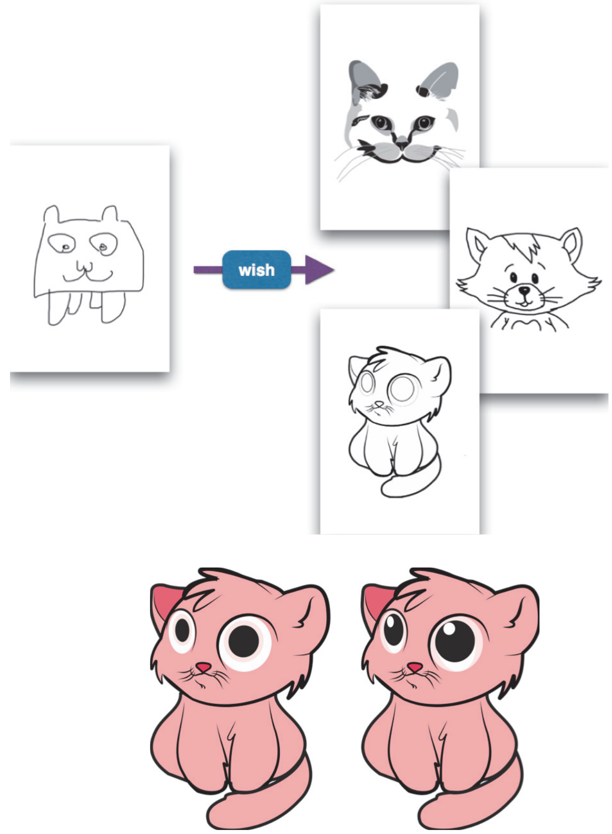


*Figure 3: By repeating the same wish, a designer can explore various options in the design space, shown here in Sketch. When a designer wants to explore a particular design direction, she may iterate on the result of a particular wish with subsequent wishes to see further refinements.*

## Wishing in Action: Scenarios and Outcomes

We discuss several potential scenarios where we believe the functionality offered by Wish can be useful. We illustrate these scenarios in the context of three prototype applications: *Draft*, an online writing application that can copyedit writing and convert outlines of documents into finished papers, *Hack*, an online code editing environment within Github that can convert pseudocode and comments into real working code, and *Sketch*, a drawing tool that can refine simple drawings into more complex ones.

### Exploration of Design Spaces

Individuals of any skill level can use Wish to explore a design space by repeatedly *wishing* and observing the outcomes as various experts attempt to fulfill the same *wish* in various ways. Using Sketch, we can produce a multitude of potential drawings in response to an initial outline, allowing users to see the various outcomes that may result from a given set of choices in the design space. By specifying a level of fidelity to the original drawing as part of the *wish*, users may request more or less substantial refinements over time.

Providing a set of expert-produced options in the design space may help designers make firm design choices. In Figure 3, we illustrate real results from using Sketch to first produce a number of drawings in response to an initial outline, then to iterate on a design, selecting a favored direction in the design space and wishing again for further refinements. As shown, this process can lead the user to add constraints to her *wish* and iterate more quickly towards finished work.

### Amplifying Expertise and Educating Novices

Novice users can use Wish to amplify their effective ability to create difficult content, effectively increasing their expertise when operating a design tool. Figure 1 demonstrates this functionality in Sketch. We produced a crude napkin sketch of a cat and used Wish to request a refinement from an expert in the crowd. The result was a substantial improvement that left few design choices to the end user. If the user sought to be more closely involved in the production process, the user could request a sequence of more minor refinements through a sequence of *wishes*, providing feedback on the desired direction at each step. This is a powerful use case, enabling novice designers to produce expert-level content by piggybacking on an expert's level of skill while retaining creative control.

Less-skilled users can also use Wish for feedback and training, improving their own unaided ability to create. By studying places where refinements are proposed and being shown rapid *in-situ* corrections on a creative product, novices can learn how to perform unaided work more effectively. Figure 2 shows a real example of stylistic and grammatical errors in a *wish* being corrected by an expert in Draft. Hack can be used analogously to teach programming by example by converting individual lines of pseudocode into real code. In educational settings where instructor time is distributed over a large number of students, receiving rapid feedback on work quality from a crowd expert could provide a way to provide focused attention to individual students' work.

### Productivity Enhancement

Experienced individuals can use Wish for productivity enhancement, creating content more quickly than working in an unassisted interface. Figure 4 provides a compelling example of this scenario in Draft. We submitted a brief four-sentence outline of a blog post about a conference visit. Wish produced a completed and detailed four-paragraph post, complete with comments and anecdotes of events and talks. Remarkably, no additional instruction or context was provided to the expert in producing the additional material; it was derived entirely from online research and reasonable guesses. An expert using Wish to produce content might increase her productivity by editing the resulting blog post as an alternative to writing her own entirely from scratch.

## System Architecture

Wish consists of three components:

1) A *collaborative editing application,* such as a text editor or illustration tool,

2) an *application-to-crowd interface*, which allows the editing application to post tasks and retrieve results from expert members of a crowd, and

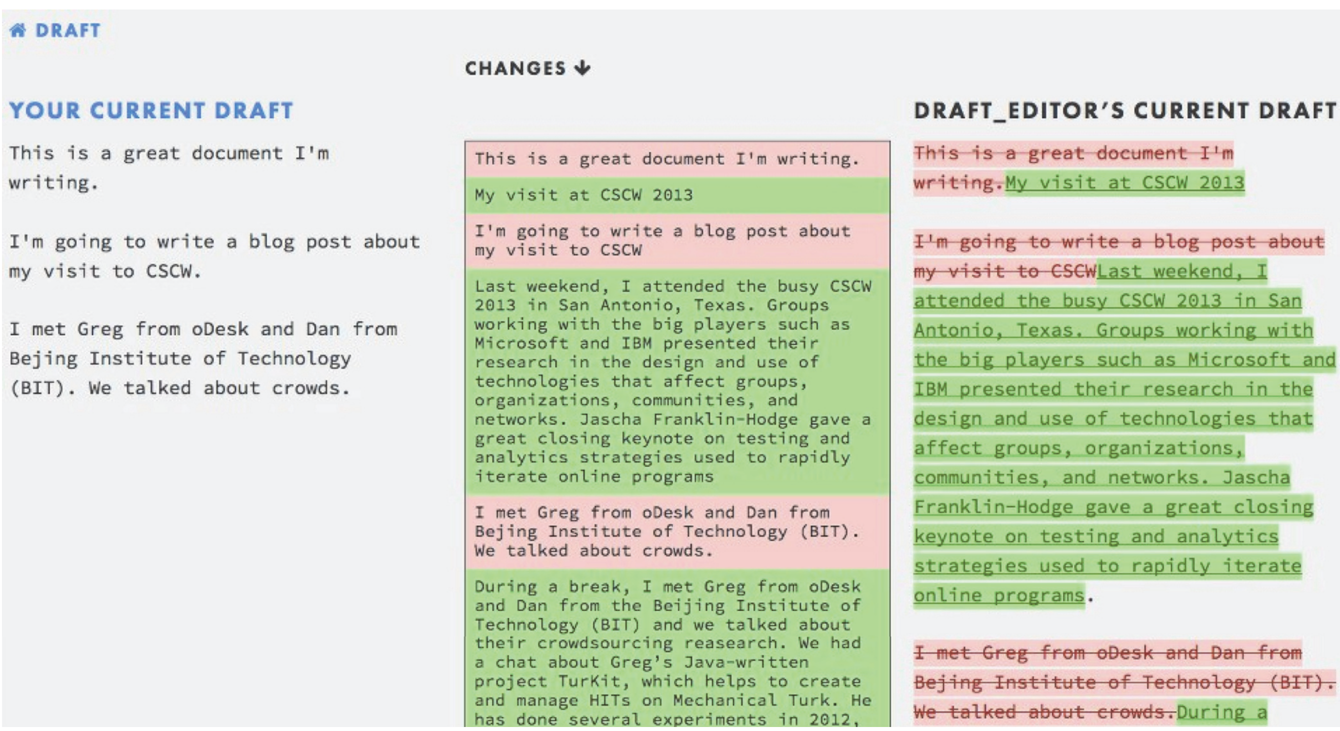


*Figure 4: A wish coming true. A brief outline for a piece of writing is converted to a fleshed-out story by an expert. In this scenario, Wish can be used for rapid production of useful content that can be further refined by the author.*

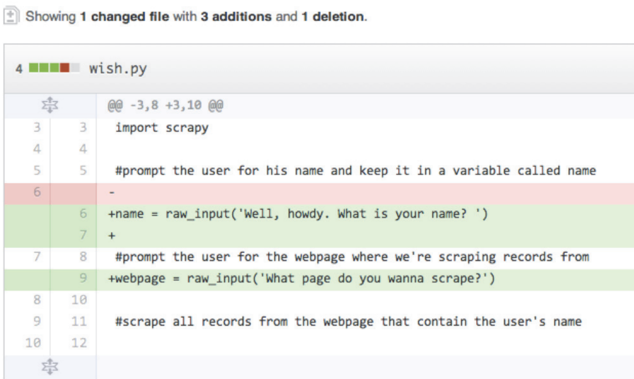


*Figure 5: Hack sends pseudocode from Github to a crowd and returns working code back into the same interface. Github's native interface visualizes changes and allows acceptance and rejection of proposed modifications.*

3) an *expert identification algorithm* used when no experts are available to recruit additional experts into a crowd to solve a task.

## Collaborative Editing Application

As a lightweight interface extension, Wish can be incorporated into any user interface that supports *collaborative* and *version-controlled* editing of a file. When added into these interfaces, Wish takes the form of a simple button within the interface allowing the user to *ask an expert*, sending the current state of the editor to the crowd (Figure 6). Existing controls permit *acceptance* or *rejection* of proposed fulfillments, which take the form of proposed collaborator changes to the document (Figure 2). In the case of a text editor, components of a fulfillment may be rejected or accepted line-by-line; for graphic design, fulfillments must be accepted in sum or rejected entirely.

The requirements for collaboration and version control permit us to incorporate Wish into a tool without modifying the structure of the editing interface (Figure 5). This means Wish can be used to extend a broad and growing class of existing web tools, including online word processors like Google Docs and Office Online, online code repositories like Github, and various online drawing environments. By using files stored in version-controlled repositories like Github or Dropbox, Wish can interact in a slightly more roundabout manner with software tools that do not yet support collaboration explicitly, such as Photoshop.

## Using the Interface to Wish

Users begin by specifying a *wish,* a partially constrained creative product at some stage of the production process. A wish can be a block of text outlining a paper to be written, a partial drawing, incomplete software code, or any other partially-defined piece of a creative product. A wish may be in any stage of the creative process, from a rough sketch to a nearly-completed work. Next, users click the "ask a pro" button within the design interface, automatically generating a request for an expert via the web to produce a *fulfillment* of the wish based on the user's initial constraints. Users are presented with a small text window (Figure 6) in which they may specify any additional instructions for the action to be taken on the wish – for example, "Please finish my drawing of a cat!" or "Can we make the eyes larger?" Such instructions are not mandatory but may narrow the design space in which results lie.

Once the results are returned, the user may accept this fulfillment, reject the fulfillment, or continue to edit the fulfillment and return it to the expert as a new wish. Because this process is embedded in an existing editing interface, users can treat the wishing process as a black-box operation in their design process without being familiar with the underlying crowd mechanics.

## Application-Crowd Interface: Sending Work to the Crowd

Wish uses the commercial MobileWorks platform to fulfill work. Unlike online work marketplaces, MobileWorks retains skill data on workers and actively matches tasks to workers deemed suitable for the task. This online platform further reproduces workplace dynamics like interviews and screening, communication between workers, fair hourly pay, worker-to-worker feedback, and peer and supervisor relationships (Kulkarni 2012).

Wishes to be fulfilled are posted into the MobileWorks platform where they are dynamically assigned to members of the crowd. End users of Wish are not permitted to configure task variables such as payment or high-level task instructions – these are preconfigured by the application and fixed. Boilerplate task text is attached to wishes from a given application explaining that the given artifact should be refined either until completion or until a certain upper time limit is reached (for example, 45 minutes of editing in *Draft)*.

When a new application sends a *wish* task to the crowd for the first time, it is automatically assigned to an experienced MobileWorker in a supervisory role (a "manager") who evaluates its skill requirements and associates these skills with tasks from that interface. In subsequent interactions with the same interface, the manager is bypassed entirely.

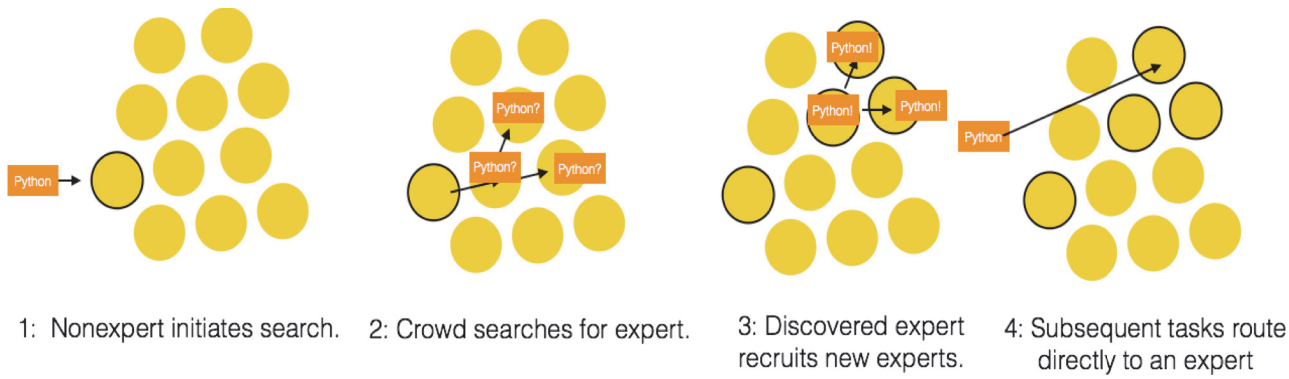


*Figure 6: Searching for new experts via Crowdsearch. If no expert is available for a given activity within the crowd at a given time, a nonexpert manager initiates a search for new expert. Once an expert is discovered, that expert recruits additional experts into the crowd. Subsequent expert tasks route directly to the first available expert in the expert pool.*

If MobileWorks indicates that one or more appropriately skilled experts are known in the crowd, the task is then passed to that pool of experts and the first person to claim the task may carry it out. In the event that a suitable expert is not available at a given time a separate process is invoked to find a new expert. This process uses the crowd to search for and recruit a new expert.

If no suitable expert is known in the crowd or if no expert chooses to claim it, the task is passed to an *expert identification algorithm* which attempts to seek out a new expert and recruit that individual into the crowd for the purposes of completing this task. Over time, this enables MobileWorks to accrue an increasingly large pool of experts within its crowd, reducing both the time required to fulfill wishes and the likelihood that a new expert search will be required again.

Once assigned, finished results are sent directly back to the invoking application. No review or verification takes place on the results if an expert has participated before – once produced, results are returned directly from selected experts. If it is an expert's first time providing results, a crowd manager is tasked with reviewing the result before approving its release.

## Adding New Experts to the Crowd

Wish's expert identification algorithm, *Crowdsearch*, provides a mechanism to augment the MobileWorks crowd with additional experts on demand when none are available. Crowdsearch operates by passing a request for expertise throughout the social network of a crowd until it finds a willing expert. Next, it uses the discovered expert to recruit additional experts to provide a pool of available talent.

Crowdsearch follows three steps:

**1) Broadcasting request.** A single nonexpert member of the crowd is selected in order to lead the search. This nonexpert is given a task to identify possible experts in her personal network with the necessary skills and solicit them to join the crowd for pay. If no experts are known directly by the tasked individual, she is asked to recruit additional individuals who might be better able to find such an individual, and the process repeats with those individuals until the network is exhausted or an expert is found. The initial member of the crowd tasked with recruitment is paid by the hour; additional members are paid only if they (or someone they recruit) finds an expert.

**2) Skills review.** Once an expert self-identifies and agrees to join the crowd, expertise is established by human screening. A trusted manager in the crowd is tasked with carrying out a review of their credentials and existing portfolio of work. If the review is unsatisfactory, then the expert is rejected; otherwise, she is approved.

**3) Pool expansion.** After approval and completion of the task they were recruited for, the first expert is used to establish additional screening criteria that may be applied to subsequent skilled experts, and takes the place of the manager in the screening process. Because experts are likely to know other experts in the same domain, this makes future recruitment easier.

## Incentives and Analysis

Expert workers are presented with the task opportunity from Wish at the point of recruitment. They are incentivized directly via hourly pay and, if needed, a signup bonus. Non-expert workers are compensated hourly for time spent searching for experts, provided their search terminates in successful recruitment of an expert, or successful recruitment of someone who recruits an expert. More sophisticated compensation schemes are possible involving partial pay for all searchers based on network distance from the original crowd (Pickard et al. 2011). These could be used to improve the time needed for this algorithm to converge on a requester.

Theoretical conditions on network size and topology that can be used to model the chance that such a search will successfully converge on a pool of experts. However, such considerations have proven unnecessary in practice. Crowdsearch capitalizes on two characteristics of crowds that make it likely to terminate successfully in producing a pool of experts. First, crowd participants are well-connected to individuals outside the crowd both through their real-world social networks and through online networks, so even a few dozen searchers will result in access to thousands of potential participants in the network. Global labor platforms like MobileWorks contain individuals across multiple geographies and demographics, providing access to a very large and diverse social network just outside the crowd. Second, experts are likely to know experts with similar skills in their networks. Once a single expert is identified and convinced to join, additional experts may be discovered with ease.

## Discussion, Limitations, and Future Work

Wish suggests that when using crowds to produce complex content, effort may be better exerted in searching for the best individuals in the crowd and trusting them to carry out work rather than combining, supervising, and verifying the outcomes of low-skilled microworkers. However, a number of issues arise concerning whether this approach is better or worse than traditional models of distributed, unskilled crowdsourcing, or as compared to approaches that involve extended engagement of a single expert hired on an online marketplace.

**Single-Point-of-Failure Risk** Wish exchanges the relative robustness of averaging results across multiple users with workflows involving voting and redundancy for the single point of failure associated with a single expert. This presents a real risk for users. What happens if a wish fails to come true – ie, if the result differs wildly from the user's initial expectation, or if the expert fails to complete the task due to abandonment? Since the user has control over whether an outcome is accepted via the version-control interface, it is possible to accept only portions of the expert's contribution. The user may also reject the task outright if it does not fall within the constraints outlined by the end user.

It is possible to imagine the user's role in verification of a Wish being played by another expert in the crowd; to the extent that Wish is a tool for design exploration and enhancement as opposed to delivery of concrete and finished results, the participation of the requester in reviewing results does not seem to be burdensome.

**Execution speed** Although most design software operates in real time, requests to the crowd via Wish are executed and returned between 1 and 24 hours later. If new experts must be recruited, the time required to fulfill a wish may extend by another 24 hours. Models exist for increasing the speed of execution by retaining members of a crowd (Bernstein et al. 2011), and future work should evaluate whether these methods can be applied to retain experts in Wish's pool to reduce the time requirement. At present, the long production cycle limits the design exploration that end users can carry out with Wish in practice, though for many use cases we believe the value of an expert revision outweighs the time required.

**Self-identification of experts** We established expertise by having a nonexpert manager review the credentials and portfolio of a self-identified expert. However, this is an obvious limitation, as an expert could readily overstate her credentials to the extent necessary to fool a nonexpert. Even in places where an expert has elements of necessary expertise, a nonexpert may not appreciate the differences in ability or aesthetic to the same extent an end user does, leading to a purported expert having a smaller level of skill than an end user. This is a potential disadvantage when compared to online labor marketplaces that directly test for expertise using objective skill tests. Indeed, in a previous iteration of Wish, we attempted to produce these tests in advance of recruitment, but found that the time involved in implementing testing systems was prohibitive compared to the quality of expert obtained. Self-identification of experts allows a wider range of experts to be discovered without the need to design tests in advance, but may be less effective at screening for true expertise compared to prewritten objective tests.

**Unmotivated searchers and payment** In theory, a sufficiently large and well-motivated crowd is likely to be able

| | |
|---|---|
| *Observation period* | 12 months |
| *Number of requests* | 2430 |
| *Target completion time* | 24 hours |
| *Total unique experts discovered* | 151 |
| *Crowdsearch initiated* | 8 times |
| *Mean recruitment time (hrs)* | 18.5 |

*Table 1: Performance of Wish and Crowdsearch in a commercial writing tool. Crowdsearch was initiated each time no expert from the existing pool did not claim a task within the completion time. The pool ultimately used over 150 experts. All requests were ultimately fulfilled.*

to locate experts efficiently following the Crowdsearch algorithm as a traditional query incentive network. While we found that the nonexpert initially tasked with searching was highly motivated, we commonly observed other members of the crowd – recipients of a request to search -- giving up their search when they could not identify individuals in their own immediate social network with the requisite expertise. This meant that in practice, the nonexpert initially tasked with the search process did most of the searching; this diminishes the utility of the network model for search. The likely reason is that the initial searcher was paid hourly and under a guarantee of payment, whereas other members of the crowd were only paid contingent on successful recruitment. Future work should experiment with hourly pay for all members of the crowd to maximize the effectiveness of these searchers.

**Uncommon Expertise** We demonstrated Wish's expert-recruitment methods in relatively easy-to-find domains of expertise like writing, graphic design, and programming. Even when no expert was immediately available, the crowd of workers was able to identify new experts for this topic with these abilities within at most one degree of some member of the already-recruited crowd.

While obscure specialties are certainly available in the broader crowd, it may be difficult to find them quickly enough to deploy an expert within an interface before a task is no longer relevant for an end user. Crowdsearch seems unlikely to work well when searching for experts in more obscure areas of knowledge and ability. These are compelling areas for future work. Can the operator of scientific software *wish* for an expert in chemistry to assist in a completing a chemical assay? Can a mathematician writing in LaTeX *wish* for a specialist to finish his mathematical proof? Future work should evaluate how difficult it is in practice to discover and recruit experts in more obscure areas of knowledge.

**Advantages: Wish as a Model for Expert Crowdsourcing** Wish's model for expert consultation provides many of the same advantages in scale and flexibility as unskilled crowd computing, while retaining the ability to provide a sophisticated level of work. We believe that it can be applied to produce high-quality expert work beyond the three applications shared here. The expert crowds used in Wish are fault-tolerant – no individual expert needs to be available at any given time in order for the system to successfully operate, and even if all known experts disappear from the crowd, a new expert can be found through a new crowd search shortly thereafter.

We have tested this viability in practice over the past year. Draft is available on the web (www.draftin.com) and has successfully processed thousands of requests for expert writing assistance over the past 12 months (Table 1). During that time, Crowdsearch has proven to be a stable mechanism to recruit and retain a pool of expert workers. As such, Wish can serve as a model approach for crowd computing problems that require the finesse of an expert with the flexibility of a crowd.

## Acknowledgments

Wish has been implemented as part of two commercially-avavilable tools. Draft is available on the web (www.draftin.com) and has successfully processed thousands of requests for expert writing assistance. Additionally, the commercial MobileWorks service (www.mobileworks.com) is built on Wish's expert-finding and task-routing algorithm, allowing end users in various domains to solicit expert work from the crowd. We are grateful to the various contributors who have helped these platforms succeed.

## References

Licklider, J.C.R., "Man-Computer Symbiosis", *IRE Transactions on Human Factors in Electronics*, vol. HFE-1, 4-11, Mar 1960.

Bush, Vannevar. "As We May Think." *The Atlantic.* July 1945.

Von Ahn, L.; Dabbish, L. 2004. "Labeling images with a computer game". *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*.

Little, G., Chilton, L. B., Goldman, M., & Miller, R. C. 2010, July. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation* (pp. 68-76). ACM.

oDesk. http://www.odesk.com

Retelny, D., Robaszkiewicz, S., To, A., & Bernstein, M. Enabling Expert Crowdsourcing with Flash Teams. *Crowdconf 2013.*

Zhang, H., Lai, J. K., & Bächer, M. (2012, July). Hallucination: A mixed-initiative approach for efficient document reconstruction. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., & Yeh, T. (2010, October). Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 333-342). ACM.

Bernstein, M. S., Brandt, J., Miller, R. C., & Karger, D. R. (2011, October). Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 33-42). ACM.

Dow, S. P., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D. L., & Klemmer, S. R. (2012). Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Design Thinking Research*(pp. 47-70). Springer Berlin Heidelberg.

Dow, S., Gerber, E., & Wong, A. (2013, April). A pilot study of using crowds in the classroom. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 227-236). ACM.

LaToza, T. D., Towne, W. B., van der Hoek, A., & Herbsleb, J. D. (2013, May). Crowd Development. In *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on* (pp. 85-88). IEEE.

Little, Greg. 2012. HumanScript. http://humanscript. blogspot.com/

Kleinberg, J., & Raghavan, P. (2005, October). Query incentive networks. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on* (pp. 132-141). IEEE.

Pickard, G., Pan, W., Rahwan, I., Cebrian, M., Crane, R., Madan, A., & Pentland, A. (2011). Time-critical social mobilization. *Science*, *334*(6055), 509-512.

Kulkarni, A., Gutheim, P., Narula, P., Rolnitzky, D., Parikh, T., & Hartmann, B. (2012). Mobileworks: Designing for quality in a managed crowdsourcing architecture. *Internet Computing, IEEE*, *16*(5), 28-35.

Dow, S., Kulkarni, A., Klemmer, S., & Hartmann, B. (2012, February). Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1013-1022). ACM.

Kulkarni, A., Can, M., & Hartmann, B. (2012, February). Collaboratively crowdsourcing workflows with Turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1003-1012). ACM.