TaskLight: A Groupware System to Facilitate Requesting and Managing Help in Teams

by

Stuti Vishwabhan

S.B. Computer Science and Engineering, Massachusetts Institute of Technology, 2020

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Department of Electrical Engineering and Computer Science
May 14, 2021

Certified by .....
David R. Karger
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

TaskLight: A Groupware System to Facilitate Requesting and Managing Help in Teams

by

Stuti Vishwabhan

Submitted to the Department of Electrical Engineering and Computer Science
on May 14, 2021, proposal for thesis research in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

People working in collaborative settings often have to request help from their colleagues due to their expertise or authority. Previous research in request management found that a performer (a person who is asked to conduct a task) would benefit from meta-data about the request. In this work, we use requesters' motivation to help their performers and their goodwill to construct informative requests; in turn, performers are able to better and more efficiently finish tasks.

We propose a collaborative request management tool called TaskLight, where requesters can provide and curate contextual information for performers. Performers are able to prioritize and manage requests based on information provided and also easily collect additional information through nuanced discussion and negotiation around requests, if necessary. The design of TaskLight is inspired by preliminary interviews with individuals who were part of collaborative settings, in order to understand their current practices of requesting help and using tools. We further investigate how different models of request can assist performers' attention management via a field study. We demonstrate the diverse use cases of TaskLight through implementations of requests such as collaborative writing, getting approvals, and making group decisions. We derive insights via a user study from our first deployed version of TaskLight and use it as a stepping stone for future direction of work.

Thesis Supervisor: David R. Karger
Title: Professor of Computer Science and Engineering

Acknowledgments

I'd like to thank the CSAIL Haystack Group, who taught me so much about HCI research and who always shared their ideas to help me with this project. I'd also like to thank all the people who participated in my interviews and studies for spending time in sharing valuable feedback. I'd like to especially thank my main group of advisors for this project: David Karger, Michael Muller, and Soya Park for the wonderful guidance and support they provided me throughout the entire project. Lastly, I'd like to thank my friends and family, especially my mom and dad, for always supporting me.

Contents

List of Figures

List of Tables

1. Introduction

People often rely on others to help them get things done. This is not always an easy feat; in fact, collaborating with others to accomplish something can be mentally demanding and stressful and can easily become more complex as more individuals become involved [41].

When people collaborate with each other, they ask each other for help to do things that we call *tasks*. These tasks can be hard to request, and can also be hard to perform and complete for various reasons. In this work, we seek to explore the needs of users regarding the coordination of tasks with the eventual goal of designing a system that users will find helpful for collaborative task management; that is, having requester-performer relationships among colleagues, where requesters ask performers to fulfill some task.

Through formative interviews with knowledge workers in collaborative settings, we identified 3 main problems concerning collaborative task management. First, we found that for many interviewees, collaborative task management is spread too thin. There's a disconnect between task-related discussion and task lists, which causes channels like instant messaging and email to subsume all task management leading to conversation overload and challenges of tracking individual task status. Alongside email and instant messaging channels, participants tend to have their own personal management practices which fail to integrate with the shared practices of the people they're working with. Second, we found that collaborative task management can feel heavyweight and overwhelming. Interviewees remarked that keeping a shared task management space can feel like an expense especially when they are collaborating in *lightweight* settings; that is, settings outside of their main work. Third, we found

that interviewees find collaborative task management impersonal and that assigning a task to someone does not fit into their sentiment of asking for help from someone. In our work, we utilize the words *task* and *request* synonymously with both taking on the meaning of asking for help rather than assigning work.

Following the formative interviews, we wanted to create a data model of information that addresses the second concern identified: how can we make task management feel less heavyweight and overwhelming? We conducted a lab study to brainstorm what information would improve collaborative task management for performers and make tasks feel more clear-cut. Inspired by previous research on how adding structure to shared information can offer benefits to make the information more digestible [19, 36, 34], we came up with a set of attributes we imagined would be helpful for performers. We compared our design with two existing models: free-form text and project management (PM) tools. We sent three different versions (free-form text, PM tools, TaskLight) of daily requests over a 5-day period to university undergraduate students, who were asked to complete the tasks by a particular deadline. We found that TaskLight users submitted assignments more on time, had fewer inquiries about the tasks, and reported less mental load compared to those in the other groups. Additionally, in post-lab surveys, when asked to compare request models, participants preferred the TaskLight model of structured requests and found the provided information helpful and easy to digest.

Motivated by the preliminary interviews and lab study, we developed TaskLight, a tool that provides structured guidance for requesters to request help to their performers and for performers to easily digest the request and voice their opinions on the request. TaskLight supports this in 3 main ways. First, TaskLight provides features for task consolidation by easily connecting with shared objects to make requests feel less dispersed. Second, TaskLight offers features for easing task authoring by recommending useful default fields to make the process of sending a request less overwhelming for the requester. TaskLight also offers features for easing task tracking by providing selective disclosure of required actions, so that requesters and performers alike more readily know what to do. Third, TaskLight includes features for making

task management more personal. We propose the notion of *collaborative task construction* where requesters and performers can share and collect required information of requests, have structured discussion and follow up on the request progress, and offer room for adjustment and negotiation around requests, as opposed to requesters simply *deciding* request specifications and informing it to performers. This suggestion model not only helps the discussion become more tractable and specific, but also adds personal dimension to the request as it reflects human conversation and changes surrounding the request.

To gain insight on how TaskLight fares as a tool for collaborative request management, we conducted a deployment user study over a 4-day period with pairs of university undergraduate students/alums. We sent requests via email to one member of the pair, who then authored a request in TaskLight and assigned their partner as a performer of the request. Members of the pair alternated the role of requester and performer, so that they experienced both roles. Examples of requests sent that are supported by TaskLight include the following:

1. Collaborative writing: These requests ask performers to review particular parts of a shared object (i.e., paper writing, designing a slide deck).

2. Getting approval: These requests are ones performers commonly receive like approving forms or signing documents.

3. Group decisions: These requests depend on the responses from multiple performers, creating a flow of inter dependencies that must be fulfilled to complete the request.

Results from our study show that users find TaskLight to be an intuitive tool that consolidates task information well, allows for easy authoring and tracking of tasks, and makes tasks feel personal. One participant remarked, *"The platforms that are used to communicate with other people, and a lot of the time, the things that I do use are essentially equivalent to speaking to them in person right like just sort of freeform messages...And I didn't feel like TaskLight got in the way of that kind of*

communication and it does provide a lot of features. That sort of helps people. Like builds a request, like all of the fields that specify, you know when do you want this done by and things like that. And that felt intuitive to us[e]". All participants stated that they would use TaskLight again for small-scale group projects with colleagues.

We conclude our work on a heuristic evaluation of TaskLight and a discussion about opportunities for more intelligent task notifications and balancing shared work space with personal task management.

2. Related Work

In our review of related work, we consider a series of experimental tools that researchers have built and explored over several decades for collaborative management of shared tasks. We then examine a family of one class of *products* that attempt to solve these problems. However, neither the experimental tools nor the products have been designed for less formal, less routinized tasks. We therefore complete the survey of related work with a consideration of the human difficulties with collaborative management of non-routine requests and tasks.

2.1 Organizing Work and Tasks

There is extensive literature on Personal Information Management (PIM). Early work in this area focused on how an individual might keep track of their own personal tasks [3, 5, 6, 7, 10, 29]. A key feature of these approaches was the subdivision of a large task into smaller components [8, 12, 32, 60].

There were initial hopes that this work could be done in email [55, 61, 46, 33]. Email users regard their inbox as a todo-list. Taskmaster [4] was proposed to facilitate the use case where email users can manage their task in email messages. Previous literature on email management hence led us to understand how individuals manage their task list [50, 38], for example they defer the task to later by marking email as unread. However, email proved to be an ineffective method for managing personal information [4, 3, 64]. An important key to this problem was that email had been "overloaded" with too much functionality and too many specialized tasks [65]. Part of the point of using email as a task infrastructure was the increasing need to work with

others in carrying out larger tasks [16, 35, 43, 44, 51]. As many organizations start shifting their communication channel from email to instant messaging, researchers also advance the task management residing in various channels [59] (e.g., scheduling tasks [13]).

Researchers began to develop specialized environments in which teams could manage these interdependent tasks and assignments. A working version of speech acts theory [52] called `The Coordinator` was created, which focused on task *assignment* and task *completion* in a highly structured environment [66]. Moran and Geyer led research into Unified Activity Management [24, 40, 62], which supported a range of collaborations from lunch-dates to large-scale collaborations [43]. These systems moved task management from general, ubiquitous tools into more specialized tools, taking place away from collaborators' interpersonal communication tools like email and instant messaging. In some cases, the absence of a more interpersonal, less formal channel became problematic [56, 57]. The free-form human conversation related to tasks was compromised when shifting to more specialized tools, making it difficult for people to effectively "talk" about a task in the system. Unifying these experience-domains remained a partially unsolved problem [53]. TaskLight seeks to unify these domains by offering a notion of collaborative task construction, where the requester and performer alike can share thoughts, perform negotiations, and freely "talk" about the task while still being in a specialized environment.

2.2 Project Management Systems

In contrast with the series of experiments described above, project management (PM) tools are widely used in various organizations [48, 39]. It is known that PM tools help teams finish on time because it increases the visibility of the decision maker [1, 39]. However, it is unclear if team members also like to use PM tools. Team members tend to rely on multiple tools including PM tools for request management [54, 41], which implies that there is a disparity between people who use and update the tool (performers) and people who benefit from the tool (project manager) [23]. Another

possible explanation is that team members are not comfortable to share incomplete work or work-in-progress with other team members [63]. It is possible that team members do not want to spam others with all the small progress they are making which could be interpreted as no progress by the requesters' end. In TaskLight, we aimed to build a groupware system that both requesters and performers were motivated to use [30], provide flexibility to handle changes to requests, and give performers power on what progress they'd like to be visible to requesters.

A separate line of tools focuses on request handling in collaborative settings such as corporations that are managing many customer inquiries in parallel [44]. Company emails are often organized using ticketing services (e.g, [67, 20]) where company representatives can organize their requests. Ticketing systems are project management tools specialized in service centers of organizations. Each request of the system becomes a ticket. Employees can customize each ticket and tag them with relevant information about the ticket, such as priority or specific fields (e.g., order number) based on the category of the ticket. Companies can customize their marketing strategy using the tools' rule-authoring interface that focuses on delegating tasks to the right agent, tracking customers' requests, and making sure they are satisfied (e.g., sending a follow-up email: "How would you rate our service?").

TaskLight is inspired by the design of the ticketing system based on how multiple collaborators send and receive requests leveraging meta-data annotated by requesters. While ticketing systems streamline the request management with the structured conversation, it still shares common sentiments of "ordering your colleague" with other project management tools, which does not resonate with the sentiment of requesting or asking for help [27, 41, 56, 57]. This sentiment of "micro-managing colleagues" was revealed in our formative interviews (Chapter 3) as well where participants pointed out that making requests through such structured tools feels unnecessarily formal and boss-like. To maintain balance between leveraging metadata like that which is provided in the ticketing systems while also making the request feel request-like, we came up with metadata that is frequently required for performers, and we recommend requesters to provide the information. We provide features to make the request feel

more request-like by providing a "narrative guide" for when requesters are sending a request. The narrative guide presents how each piece of information will help performers in a text layout that sounds like sending a message to a friend. This way, the structured meta-data leveraged still streamlines requests while humanizing it.

2.3 Request Management

The preceding discussions focused on the design and evaluation of *tools* to help individuals or groups to manage their tasks. Many of the tools were designed for management of workflows [55, 61] or other formalized and even routinized tasks [2, 40]. A less well-understood area of human work-practices involves less formal and non-replicated tasks. Muller et al. referred to this subcategory of task-management as request-management, and showed that requesters and performers (the people who responded to the requests) found this work to be time-consuming and stressful [41], requiring sometimes extensive negotiation of the timeframe and content of the task (see also [56, 57]). PM tools have largely failed to address this problem since the PM model operates on the requester deciding task specifications for the performer as opposed to having a collaborative task construction process between both parties.

Several of the preceding themes re-appeared in this work, including the task-decomposition into smaller work-units [8, 12, 32, 60] and the difficulty of managing diverse activities that did have a single organizing principle. While Muller et al. began to describe a new kind of abstraction to unite these disparate tasks and requests [42], this work was primarily a specification and problem-analysis, and did not lead to technological solutions. In this paper, we seek a more general support technology to the common aspects of the diverse sub-categories in [41, 42], and we use the word "tasks" to refer to the general case of such human-to-human assignments, obligations, and negotiations.

Prior study in request management reflects prevalent challenges in managing requests in organizations. Fortunately, existing systems guide and enlighten us on how to help and facilitate request management in different settings. As a notable

example, IntroAssist [27] guides requesters to write informative and well-structured introductory-request emails in order for performers to do the task well. Building on such a system, we propose a new system TaskLight which streamlines making requests in team settings. In addition, we expand a notion of *requester affordances* as we believe requesters are motivated to help performers since performers are providing help to requesters' work. We present interfaces demonstrating that this motivation and goodwill can be spent on the right causes. For example, requesters provide useful information for performers and help them manage their schedule and their attention.

3. Formative Interview

We sought to understand how members of organizations govern request management and what are the decision factors or affordances they need in request management tools. We conducted semi-structured interviews with seven individuals who are part of teams and have decisions over tools and collaboration practices of team works, unlike people who have to adhere to tools that have been required by their organization policy (i.e. *I need to use Trello because my organization requires so*).

Participants We recruited participants via posting on a mailing list of a private university and word-of-mouth. Each interviewee was compensated $15 for their time. Participants (5 females, 2 males) were officers of a university council, a diversity initiative of the university and extracurricular clubs, and working on a team project for courses.

Interview protocol Each interview was 45 minutes long and driven by a script that covers questions related to interviewees' practices of collaboration and their tools. In addition, we presented existing project management tools such as Gantt chart and Kanban board and asked interviewees to compare them to their current practice.

Interviewing self-organizing team members led us to understand what *organic* ways of request management practices look like and what is considered in the decision. We understood the current practices of people working in collaborative settings and identified a list of 3 main needs: 1) task consolidation, 2) flexible, easy-to-parse structure, and 3) personal dimension to a task.

First, participants stated that the fragmentation of tasks over various platforms (shared request management space, personal management practice, messaging) makes it difficult for them to stay up-to-date on all channels and to actively update new ones they haven't formed a habit with. Therefore, a need for consolidating tasks emerges, so that tasks feel less disconnected. Second, participants mentioned that most of their task-related discussion occurs in chat, but that they lose important information and context as the conversation gets easily cluttered. They state that they like to have structure to make information more digestible, but that current tools feel too heavyweight of a structure to maintain. Thus, a need for having flexible, as-needed, and easy-to parse structure arises. Lastly, participants said that current tools supporting request management feel too impersonal. They raise concerns about how assigning tasks to someone feels boss-like when they really want to ask for help from a peer. They also state how they'd like to give their performers privacy as they complete tasks. Thus, having a personal dimension to a task is essential.

3.1 Fragmentation of tasks result in extinction of collaborative task list

Echoing previous work, participants indicated that email and instant messaging are the most dominant channels for request management within teams. Team members send, receive, and follow up with the requests via chat interfaces. However, four participants said that this is not what they intended at the beginning of the project. When they started the team work they built a central workspace or project management space, so-called team master list. The workspace helps team members to keep everyone on the same page and view their team progress at a glance. The positive sentiment of having the shared task management place is universal among interviewees; this ideal setting, however, does not last long, is not maintained on time, and eventually disappears. Participants suggested that because team request management is scattered among too many different platforms, the shared request management space

becomes extinct.

Firstly, participants suggested the difference between personal and collaborative request management. Even if each request is collaborative, once they are assigned or asked a request to perform, they tend to *offload* the request into their personal task management tool. One said they keep moving the request to *"a personal page I have where I keep track of [it]"*. This offloading can be explained by previous work in request management [41], Muller at al. referred to this personal list of requests as a "Masterlist" that helps individuals to manage requests. Our interviewees add that their habit of personal task management also contributes to their behavior of giving up on the shared request list. Individuals develop their own means of organizing requests and it is often highly customized. For example, requests are represented as a customized abstraction that varies for individuals based on the level of details [3]. Participants said the way that individuals manage their tasks outweighs the group norm: *"I don't think all the team members like kind of adhere to that. I think when they finished [a request] they forgot."*

Participants also mentioned the disconnection between task-related discussion and the task list. Team work inherently faces numerous changes, needs to be adjusted accordingly [49, 42, 58, 56], and oftentimes requires volatile coordination of non-routine intellectual work [11]. Hence, it is natural that people have a large volume of task-related conversation, as one interviewee said: *"There's a lot of back and forth with emails and waiting to get the responses basically before we can move on to the next step"*. Participants said their conversation channels eventually subsume all task management such as clarification regarding tasks, tracking, reminders, and follow-ups. One participant said, *"Once they finished critical tasks said like the lower part of the stack, then they would message out saying like: 'This is done, you can start working on the next thing up' "*.

Having tasks fragmented among multiple platforms (collaborative task list, personal management practice, email/chat) makes it difficult for people to keep up with all channels and actively update new channels (i.e. the shared task list) they are not used to. Thus, a need for task consolidation among multiple platforms arises.

3.2 Conversation overload makes parsing through tasks difficult

Not surprisingly, when most task-related discussion occurs in conversation channels, those channels suffer from their own problem of overload and get easily cluttered and difficult to follow, echoing previous work on group discussion [22, 65, 18, 68]. As a result, team members frequently lose important resources related to requests somewhere in their conversation log. One participant said: *"We wanted to organize some social activity for [our living group]. And then we had a meeting and this is actually a follow up to it, a meeting that we had on Zoom. And somebody from the exec committee is asking about the logistics that we discussed on the meeting. I guess it's somewhere buried in these emails."* Some team members become frustrated by the volume of the chat and start to ignore the conversation unless they are addressed specifically: *"That was really hectic.. At one point, it become really overwhelming, so I decided to make a filter; whenever I get an email [about the club], it just mark [the messages] as read."* Because conversation channels are free-form and meant to cover a variety of topics, it becomes difficult to parse through the conversation to find specific, relevant task details. This supports previous research on how structure for a task is needed in order to make tasks more digestible [19, 36, 34].

3.3 "Managing" requests feels expensive

Following the previous section, while structure is needed, adding *only structure* is not sufficient. Although nearly half of the interviewees appreciate the structured view of project management tools, they were not sure if keeping it up-to-date was feasible given the nature of their team. Out of the seven interviewees, four liked to incorporate Kanban boards into their team projects. They said it is an effective way of visually distinguishing requests in different stages. One interviewee also said that the ability to customize columns could be useful to express project-specific stages. Nonetheless,

participants mentioned that it requires too much management of the board: *"I think the problem with the club team is I feel like I'm not a full-time team [manager]. I mean, it's a part of the thing, my responsibility, but managing this board might be expensive, none of our officers working on this all day. So I think it'd be really easy that this board might be get outdated."* Therefore, while structure is needed to make tasks easier to parse, it can also feel too heavyweight and expensive, especially for *lightweight* teams whose main work is outside of their collaborative setting. A more nuanced version of structure is needed. From this finding we identify the missing need of having *flexible, as-needed* structure that can keep requests minimal when they need to be but also provide flexibility to add more details as requests become more complex, supporting heavyweight and lightweight teams alike.

3.4 "Managing" requests feels impersonal

Interviewees also raised concerns about the nuanced sentiment of "asking for help" versus "managing their colleagues". They said requests made to their colleagues are more like asking favors rather than assigning tasks; hence, tracking individual stages or progress is unnecessary as long as *things get done by deadlines*. All of the interviewees said that Gantt charts are not suitable for their team because it does not align with task planning. They stated they initially assigned big chunks of tasks to each team member, let them work on it towards an internal deadline, and combined them later. *"I don't think we like laid that [detailed plans] out as a as a firm rule."* This finding shows that people don't want to micro-manage their colleagues on tasks. Instead, they'd like tools to support performer privacy and to have the notion of a request - asking for help rather than assigning a task.

4. Request Management Field Study

Following our formative interviews, we conducted a field study to understand what kind of data model can make request management feel less overwhelming. From our formative interviews and previous research, we found that many participants rely on free-form channels of text like email and chat for conveying requests to other people. Therefore, we designed our first study condition to be the "text group"; that is, individuals who receive requests of free-form text. Additionally, in our formative interviews, we asked interviewees about their experience using existing project management (PM) tools that are designed to streamline and provide effective views of shared progress. Accordingly, we designed our second study condition to be the "PM group", which comprised of individuals receiving requests with common attributes among PM tools. Our last study condition was the "TaskLight group", which comprised of individuals receiving requests with attributes that we curated ourselves and thought would be helpful to complete the task.

Our hypothesis for this study was that having additional structured information in the tasks would help people accomplish the tasks more effectively and easily. Indeed, this hypothesis was generally supported based on the results of the study. TaskLight users completed tasks more on time, had less clarification questions, and reported feeling less stressed when doing the task compared to users in the other groups. However, users in the PM group visited the request list most often and completed more tasks than other groups. Text users reported perceiving that the requests were less difficult compared to those in other groups.

Participants We recruited participants via posting on community-wide mailing lists of a private university. Each participant was compensated $30 for their time. Initially, 32 participants, who are undergraduate students at the university and self-reported that they recently worked in a team, were recruited and participated in the introductory session. We assigned each participant randomly to one of the 3 experimental conditions. By the end of the study, due to dropouts, 21 participants (19 females, 2 males, mean age=20.0) remained. To break it down by condition, in the text group, 7 participants (7 females, 0 males, mean age=19.9) remained. In the PM group, 9 participants (7 females, 2 males, mean age=19.8) remained. In the TaskLight group, 5 participants (5 females, 0 males, mean age=20.0) remained. We also considered participants to be dropouts if they did not check back to their request list after we sent out the list. Since these were not influenced by condition, it is valid to ignore their existence and continue to treat them as dropouts as having been independently and randomly assigned to the three conditions.

Study protocol To examine whether the TaskLight data model helps performers to manage and conduct requests, we compared TaskLight with the free-form messages and project management tools data models. We divided participants into three groups – the text group (N=11), the PM tool group (N=11), and the TaskLight group (N=10). The study was designed such that over the course of 5 days, participants would be sent requests that they would have to complete by an associated deadline. All three groups received these requests through Google spreadsheets.

For the text group, there was only one column and each request was fully described in free-form text with a deadline of when it should be completed by (ex. "tomorrow" or "this Sunday") in a cell. Requests for the PM tool group were broken down into the following columns: `request description, priority, status, deadline`, and `requester` based on the common attributes found in current PM tools. For the TaskLight group, each request was broken down into the following columns: `title, purpose of request, location of request, anticipated completion time, deadline`, and `steps`. We added the `title` field as a way for

performers to gain a quick grasp of the task at a glance. We added the `purpose of request` field to provide performers a reason as to why the task was important or needed to be done. We added the `location of request` field so that performers have a readily available link to where they can perform the task. We added the `anticipated completion time` field to provide performers an estimate of how long a task may take. These fields were added based on what we thought would be helpful to performers based on findings from previous research and from our formative interviews. The deadline fields for the PM tool group and the TaskLight group were provided as dates (ex. "03/28").

Participants were invited to a 30 minute long tutorial session a day before the start of the study. Participants were instructed to visit their request list throughout the study period and reach out to us via email if they had any questions regarding requests. In addition, the TaskLight group was introduced to each attribute in the tutorial session. Their spreadsheets were sent after the tutorial session via email. There was no notification made other than when we sent the spreadsheets, and participants were asked to check for and conduct their requests on their own time. The study was conducted for 5 days and each request was made at a different time of the day, but at the same time to every participant. Each request had different deadlines. The following is a list of 9 requests distributed to the participants:

1. Invite authors to a new Google drive Folder.
2. Schedule a meeting.
3. Write a review comparing the university and other colleges' policies on maintaining undergraduate social life during COVID.
4. Find 3 virtual social events that are happening this week at the university.
5. Provide comments to Wikipedia START-class article of Covid Symptom Study.
6. Brainstorm your desired COVID-friendly social events and make a flyer of it. (Dependent on #4)

7. Provide comments to Wikipedia START-class article of Edit-a-thon.

8. Provide comments to my writing on universities' COVID polices.

9. Make the flyer more accessible according to accessibility guidelines. (Dependent on #6)

The prompt for each request of each version can be found in Appendix A.1. Upon the end of the study, we solicited participants to fill out an exit-survey. In the exit-survey, participants were asked to fill out the NASA TLX form [25] to evaluate their subjective workload of managing the requests and compare their request list to another group's version of request lists. The text and PM groups were shown the TaskLight version of request lists, while the TaskLight group was shown the text version of the request list.

4.1 Results

To evaluate how different versions helped with managing requests and delivered clear request specifications, we measured how many tasks were completed, lateness, error rate, and how often the participants contacted us for clarification. When calculating the average number of completed tasks per person, if the participant performed the task by the end of the study, we considered it a "completed" task. This means that tasks that were late or did not follow task specification were still considered to be "completed", just not completed on time or properly. When calculating lateness, we considered tasks that were incomplete (not performed by the end of the study) to be "indefinitely late" since not completing a task at all is worse than completing a task late. When calculating error rate, we considered two things: 1) each person completed a different number of tasks, and 2) a participant can fail to follow task specifications only if they completed a task. Thus, we elected to specify only the average percentage for error rate since the average number would have a separate relative meaning per person. When calculating how often participants contacted us for clarification, we

averaged the number of clarifications per person per study condition and calculated its corresponding percentage.

Table 4.1 shows the result of the study as well as responses from the NASA TLX form [25] provided in our exit survey. We found that PM tool users benefit from attributes like `status`, `deadline`, and `priority` and would have liked a field similar to the `anticipated completion time` attribute. TaskLight users benefit from the attributes like the PM tool group, but have fewer inquiries about the tasks and less stress. Users in the TaskLight group mention that the clear task specification, formatting, and more specialized attributes provided like `purpose of request` and `steps` were helpful in terms of understanding the task and making it easier to parse. We found that text users perceive the requests as easy to view at a glance and feel that the free-form text version of requests is more personalized. Participants from all 3 groups stated that they try to help performers accomplish requests by providing detailed information to them and making themselves available for questions/concerns.

Table 4.1: Results of the field study. Users in the PM group visited the request list most often and completed more tasks than other groups. TaskLight participants completed tasks more on time and had less clarification questions. They also reported less mental load and stress. The text users perceived requests less difficult compared to the others.

	Text	**PM tool**	**TaskLight**
Aver. # of request list visit times per person	11.8	**61**	33.8
Aver. #, % of tasks completed per person	4.9, 54%	**5.8, 64%**	4.4, 49%
Aver. #, % of late tasks per person	5.14, 57%	4.9, 54%	**4.8, 53%**
Aver. % of tasks that do not follow task specification per person	36%	21%	**2%**
Aver. #, % of clarifications requested per person	0.7, 8%	0.7, 8%	**0.2, 2%**
How mentalży demanding (1-Very low, 7- Very high)	3.8 (std=1.9)	3.8 (std=1.4)	**3 (std=0.7)**
How successful	4.4 (std=1.9)	**4.6 (std=2.2)**	4.5 (std=2.1)
How hard	**3 (std=1.4)**	3.8 (std=1.3)	3.25 (std=0.8)
How stressed	3.8 (std=1.6)	4.7 (std=2.3)	**3.25 (std=1.8)**

PM tool users were most up-to-date on requests and had the highest completion rate The PM tool users visited the list most frequently and had the most requests completed compared to the other two groups. Participants utilized different attributes to manage requests. Many participants utilized the `status` attribute to keep track of what their next step in the task was. Some participants said that they used the `priority` attribute to triage tasks: *"I looked at the deadline and the urgency of it to determine if I had to work on it immediately or push off"*. However, one

participant remarked whether having the requester assign the `priority` attribute is actually useful when making requests to other members of the team, especially related to its effect on missing deadlines, i.e. if it's low priority, missing the deadline is okay. *"Priority was useful because it gave me an idea of how the effects of missing deadlines, although in a team setting you don't really want to be thinking like that."*

Although it was not given in the PM tool version, a few participants estimated completion time of the request themselves and used it for scheduling tasks: *"I always have a weekly TO-DO list sorted by deadlines and labeled with the amount of time I predict certain things will take, then schedule which work I need to finish on each day."* At the same time, one PM tool user expressed mixed feelings of being "given" an anticipated completion time by the requester: *"Anticipated time is interesting because it can be both limiting to the effort I put in while encouraging me to put in more time to make sure I reach the requirement. If I reach the limit and I'm not done, seeing the anticipated time may make [me] just sloppily finish the job since it's only supposed to take x minutes."*

There were also participants who did not rely on any attributes but relied on send date to perform the requests: *"I mostly tried doing the task as soon as it saw it, even though it was low priority or the deadline was further away. I also noticed I naturally went through the tasks like a list rather than using the urgency or deadline as a factor."*

Despite frequent visits to their task list, PM tool users tended to not turn in requests on time compared to TaskLight users (but only by a slight margin). One PM user said: *"I portioned out time every day to do the tasks. When tasks were added last-minute, I had difficulty getting the tasks done on time as a result."* Some PM users compared their version to TaskLight and said `anticipated completion time` might have been useful, while others preferred to designate it themselves. Unlike the PM tool group, users in the TaskLight group were given the anticipated completion time and said they made good uses of it: *"I also think that the anticipated completion time is useful because I can try schedule my day around the necessary tasks."* Whereas another participant remarked: *"The time estimate given is not nec-*

essarily reflective of my efficiency, so I prefer to see for myself and make my own estimate".

TaskLight users had fewer inquiries about the tasks. Participants who used TaskLight versions tended to have less need to clarify the requests, which indicates that the TaskLight version gave clearer task specification and easier-to-parse details compared to the other versions. For example, some questions and clarifications we received from the text and PM group users via email asked for information that was automatically provided to TaskLight users. Participants from both the text and PM groups asked us via email to share our gmail addresses even though they had access to it in previously sent emails. This information, however, was provided to TaskLight users automatically in the `steps` attribute for ease of access.

Participants also said the `purpose of request` attribute helped them to tailor and understand the specification better: *"The purpose of the task was most helpful because it allowed me to understand why I was doing a particular task, who the target audience was, and how to prioritize completion of the task"*. Text users said that it would have been more clear to follow the specification if they were given the information: *"Sometimes the tasks were not clear as they could have been which led me to email the team members to clarify".*

TaskLight users spent less mental load and were less stressed. TaskLight users reported the lowest stress level of a 3.25/7 compared to users in the text group (3.8/7) and PM group (4.7/7). Participants said the formatting of TaskLight made it *"easier to read and parse the information"*. Other participants also said the `steps` attribute helped them feel like the request was less intimidating: *"I like this better since the task steps give more detail about the task, making it easier to complete"*. After viewing an image of the TaskLight version of the request list in the exit survey, PM users also made similar remarks on TaskLight: *"They also explain the tasks a lot more in depth, which makes the task much more specific and less brain thinking"*

Text users perceived the requests as easier compared to the others. Text users reported that the overall difficulty of the tasks was a 3/7 which was lower

compared to users in the PM group (3.8/7) and TaskLight group (3.25/7). While many participants found the TaskLight version easy to parse, some found the text version was more easy to see at a glance: *"This is easier to digest than the task list we were given and therefore seems less daunting/hard to do"*. One TaskLight user thought that the text version was very personable when shown the text version of the request list in the exit survey: *"It's very personable and almost personal. It feels like a friend is asking me to help them out"*. Furthermore, participants found the text version better for a quick request: *"It [the PM tool and TaskLight] might be too meticulous of a process for small tasks."*

Participants across all 3 groups stated that they provide help to performers Our exit surveys asked users about help they offer as requesters to performers: "When you work as a team and ask your colleagues for help, what kind of information do you offer them to try to be helpful? How do you help them?" From responses among the three groups, there were 3 main ways in which they tried offering help as requesters. First, they try to be *"as clear as possible when giving them [requesters] information about a certain task"*. They do this by *"giving instructions on how to do things"* and trying to *"lay out all the steps so it is easy for them to follow"*. Second, participants mentioned how they try to point performers to the right resources: *"I try to send them my work or a link to my work"*. Another mentioned, *"I offer information that is relevant to the task that I think they may not know"*. Lastly, participants expressed their willingness to being available and patient when asking for help: *"I wait for any questions they have and help them, maybe by giving them demonstrations, or if they are delayed in something maybe check in and ask if they require help"*. Responses from our exit survey show that there is indeed effort put in by requesters to provide help to performers, especially regarding clear task specification, relevant task information, and availability to resolve questions/concerns.

5. Design Goals and Choices

Making and managing requests can be a complex and challenging process. Our findings from our formative interviews show that request management is fragmented, can be overwhelming, and is impersonal. Our findings from our field study show that people benefit from structure, but also desire a human, personal element to requests. To address these challenges, we decided to create a tool TaskLight. We have identified the design goals for TaskLight below. These individual design goals aim to help requesters ask their request and help performers conduct the request in a way that feels more intuitive and personal.

Provide requesters with guidance and help to author informative requests for performers: Previous work revealed that requesters were anxious and not confident about asking for help [41]. Guidance for writing well-structured requests can help requesters feel more confident [27]; furthermore, requesters can help performers manage their attention [36, 46]. This is supported by research that shows how requesters like to fill out forms as opposed to free-form text [34]. However, these results along with feedback from our formative interviews also show that while people find structure useful, they sometimes find it too expensive to use, and they don't want to be forced to use it, especially for small-scale tasks.

Thus, we created the goal to augment the guided form structure by suggesting fields that performers can benefit from while maintaining flexibility. Allowing requesters to curate information decreases mental load and allows for easier use. Having structure that is encouraged rather than forced can help users keep requests minimal when they need to be but also provide flexibility to add more fields as requests become more detailed and complex. Results from our request management field study

augment this notion. Our field study shows that most participants appreciated the task being expressed as different attributes due to ease of parsing information and more task detail provided.

Give performers the ability to communicate about the task: Not surprisingly, over the course of the work, performers often pose questions and clarifications to requesters and offer their own set of changes to the request [56]. Our study results are indicative of this - participants in all 3 groups posed questions and clarifications about the request to requesters. Supporting performers to easily have a conversation about a task allows for a collaborative task construction process, where the request is not a static artifact from requester to performer, but a dynamic, collaborative one that leads to the formation of a request suitable for both parties [42]. This collaborative task construction process not not only benefits sparking conversation, but also helps reflect changes in team work. Previous research shows how team work is inherently dynamic and faces numerous iterations. As team work changes, those changes must also be represented in the structure of the task. Thus, part of this design goal is to provide performers an ability to directly add or mutate structured information about a task. This supports the collaborative construction process as both parties are able to update the task and reflect team work changes. Allowing for collaborative task construction is not just suitable for handling conversation and sets of changes to the task, but also humanizes the task since it reflects the needs and wants of both parties directly, addressing concerns raised in our formative interviews about "micro-managing" colleagues.

Support attention management around actions occurred throughout the life-cycle of requests that require the stakeholder's attention: Results from our formative interviews indicate that users are often overloaded by the amount of information they receive related to a request, especially through chat and email [21]. Some easily become frustrated by the volume and start ignoring the conversations unless they are specifically addressed. Furthermore, there is often a single group thread for all communication about the requests when there does not have to be one. People could directly contact and notify just the relevant people related to

the task. However, with a single group thread, performers face the burden of digesting the messages themselves to see what requires their attention or if it requires their action. Despite having the responsibility to complete the request, performers' attention towards requests is not always needed and requires different types of action. For example, if a performer asks the requester a clarification question, their attention towards the request is unnecessary until the requester responds back. However, in chat or email, it is hard to distinguish and manage their attention on requests that require their attention since everyone is receiving the same notifications. In our field study, we observed that PM users customize the `status` attribute to note their next step or blocker (e.g., confirm a specification with teams, wait for a response from the team about X) and quickly pick up from the state they left. Interface supports can help users by automatically updating and disclosing the status of requests based on users' activity. However, this kind of status is separate from the status attribute provided in PM tools, which only have *shared* information about a task. Accordingly, it is difficult to have *per-person* information about a task. For example, a "blocker" may mean that one person is blocked on a task, but it may also mean that it is urgent for a different person to do something to unblock the task. In such cases, it would be useful for performers to manage their requests based on what requires action from their end as opposed to knowing the general status of a task (i.e. if it is blocked or not). Having a fine-grained status (i.e. Accepted, Waiting for clarification, etc.) for each performer can guide them to point their attention to required action as well as provide insight to the requester on which performer has accepted and started working on the request or which performer is blocked on it, so they can stay informed and take appropriate action for each performer.

Provide support for task consolidation among commonly used platforms: Our formative interviews show how scattered tasks among different platforms can make it difficult for people to stay up to date on what they have to do. Providing integration features with commonly used platforms like Gmail and Google Drive can help consolidate information and provide a seamless way to link tasks between two platforms.

Making tasks feel more personal: Our formative interviews show how people report feeling weird micro-managing colleagues on tasks when they really want to request for help. Users in our field study also reported that the requests sent to the text group felt more human and personal, like *"a friend is asking me to help them out"*. Providing features like text guides with embedded attributes can help humanize structured requests. From our formative interviews, participants also stated that they wanted to provide their colleagues freedom to work on their own. Previous research has shown that individuals feel less comfortable working in front of colleagues due to fear of being *observed* by them [63]. Providing private fields to performers (i.e., priority, personal notes) can help performers feel more comfortable making decisions about the task and conducting the task in their personal way since they do not have to fear being watched by their team members.

6. TaskLight: A Collaborative Request Management Tool

This section introduces the interface of TaskLight. Fig. 6-1 shows the most salient features implemented in TaskLight and how each feature addresses the findings found from the formative interviews and request management study conducted and the design goals. We present 3 main types of features. First, we discuss features to ease task authoring and tracking. This includes providing requesters default useful fields to specify the request and for performers to work on the request. Additionally, we provide status labels and selective task highlighting to help users manage their attention and track required actions. Second, we provide features for task consolidation. This includes a sidebar feature that allows users to draft requests directly from their email and from shared team resources. We also provide a dedicated chat channel for each task that keeps the details and context of the task specific. Third, we offer features for making task management more personal. We introduce our suggestion model where performers can offer their input on a task that requesters can later review. We also present the Narrative Guide to show requesters how each field can be used and how it can benefit performers. Finally, we provide private fields so that performers have freedom to work in their own personal way.

Findings	Design Goals	Features
Managing and keeping tasks up-to-date is too much work	- Guide requesters to author informative requests - Support attention management around actions	- Useful default fields - Tracking required action via status labels, highlighting features
Task details are scattered over too many platforms	- Support task consolidation and integration features	- Drafting feature from email and team resources - Unique chat channel per task
Assigning tasks to others feels impersonal	- Performers can ask clarifications and offer suggestions - Make tasks feel personable and private	- Suggestion Model - Narrative Guide - Private fields

Figure 6-1: How different features of TaskLight address findings from the formative interviews and study conducted as well as the design goals laid out.

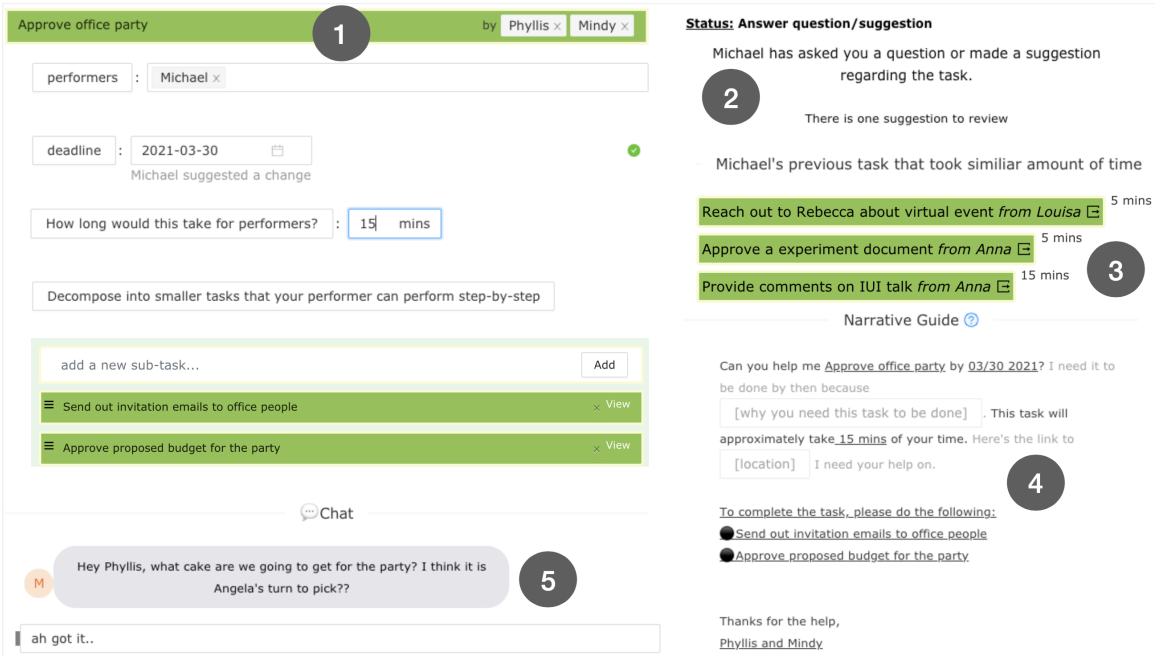

Figure 6-2: The authoring interface of TaskLight for requesters. (1) Request-related fields. (2) Status and required action for the viewer. (3) Helper features for requesters. Based on the field that requester is filling out, the helper feature presents relevant information. For example, in this figure, the user is working on assigning an anticipated completion time and is shown the performer's previous requests that took a similar amount of time, so that they can adjust the time by comparing to those requests. (4) Narrative guide and selection interface of what field to include in the request. (5) Chat interface.

6.1 Features to ease task authoring and tracking

6.1.1 Useful fields for requesters to specify the request and performers to work on the request

To provide guidance to author requests, we added default fields that requesters can use and benefit from. We improved upon the fields we introduced in our field study as participants said that they were useful to perform the requests. For example, the participants remarked that the `priority` of the request is somewhat informative but it could be more informative and accurate when they decided it based on the context of why this request is important. Hence, in TaskLight, the requester does not determine a priority of the request, however, they are encouraged to provide `the purpose of the request` which in turn will be used by performers to determine the priority of the request. As for the `action items` field, Kokkalis et al. work found that even crowd-workers, who do not share the context with the actual holder of the action items, can help derive insights and author context-dependent action items [32]. Our `action items` field is inspired by their work and we provide the guidance proposed in their work.

While the participants acknowledged usefulness of the `anticipated completion time` attribute, they also expressed concerns that requesters might not be able to provide accurate estimation since requesters have less context and expertise, and it would incur too much workload for requesters to come up with the information. To mitigate this, we provide requesters with affordances to find this information more easily. We have a helper feature for each attribute that provides relevant information when requesters are authoring a request and hovering over each field. For example,

- Anticipated completion time: Displays recent requests completed by the performers that both the requester and the performers were involved in for privacy reasons. Based on similarity of the nature of the request, the requester can tune the right anticipated completion time for the current request.

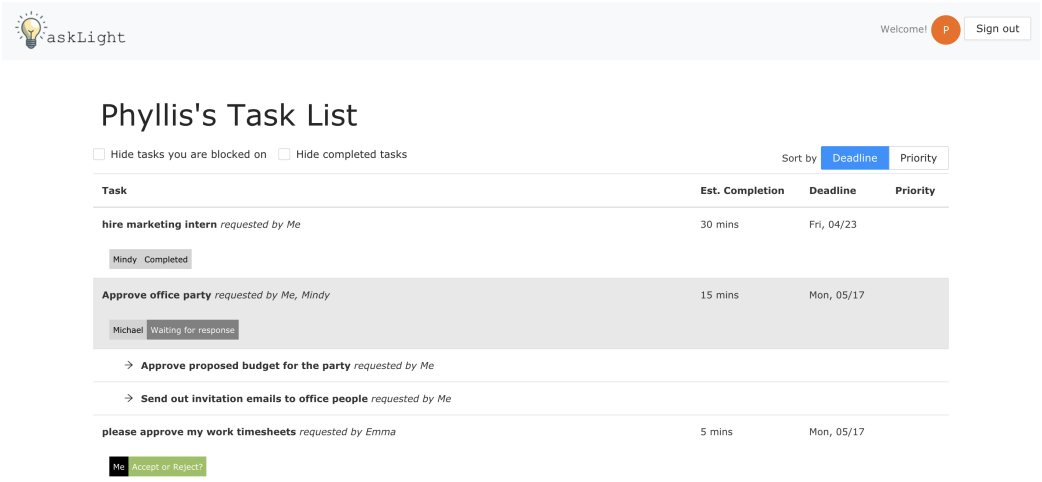

Figure 6-3: The list view allows the user to determine which request to focus on at a glance.

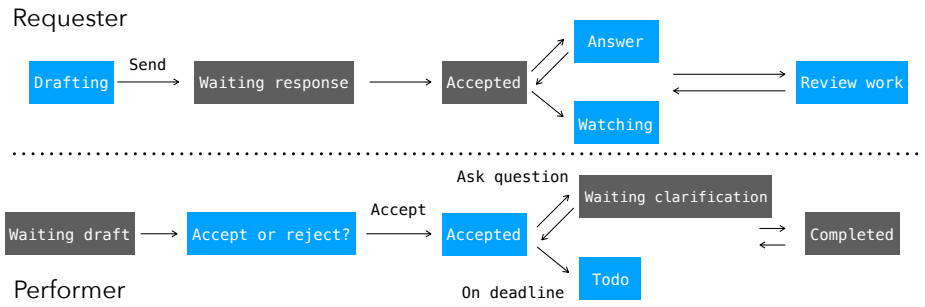

Figure 6-4: The evolution of a requester's status and its mapping to the performer's status

- Action items: Shows a list of related resources or team documents of the current request, so more specific action items based on the related resources can be given. We will present how this could be situated in request authoring in Section 7.1.

6.1.2 Tracking and managing requests with required actions disclosure

Our field study showed that the PM tool group had the highest request completion rate compared to other groups, and many in the group used the `status` attribute to keep track of their request. TaskLight provides requesters and performers status

updates to maintain visibility of the request. We provided a list view (Fig. 6-3) where a user can check which requests require their action (highlighted) and which do not (de-highlighted with gray) based on status of requests. The following is a list of performers' statuses. Fig. 6-4 illustrates how the status of a request evolves from creation to completion and shows the corresponding requesters' statuses as well.

- *Draft to be Sent*: Inspired by a draft feature of an online software collaboration platform [1], this status allows requesters to inform performers that they are working on a request and that performers can expect that the request is under way and schedule their current requests around it.
- *Accept or Reject?*: This status tells performers that there is required action on their end to indicate if they can help the requesters or not.
- *Accepted*: The performer has accepted the request.
- *Rejected*: The performer has rejected the request; the requester must find someone else to perform it.
- *Todo*: The performer should complete the request soon; the requester should be available to answer questions.
- *Waiting for clarification*: Either the performer or requester has asked a question or offered a suggestion and are waiting for a response back.
- *Completed*: The performer has completed the request; the requester is prompted to review the performer's work.

6.2 Features for task consolidation

Feedback from our formative interviews showed that task lists and typical platforms for communication like email and chat are disconnected despite task-oriented discussions occurring in those channels. We sought a way in which common communication channels and task lists can be connected.

[1] https://github.blog/2019-02-14-introducing-draft-pull-requests/

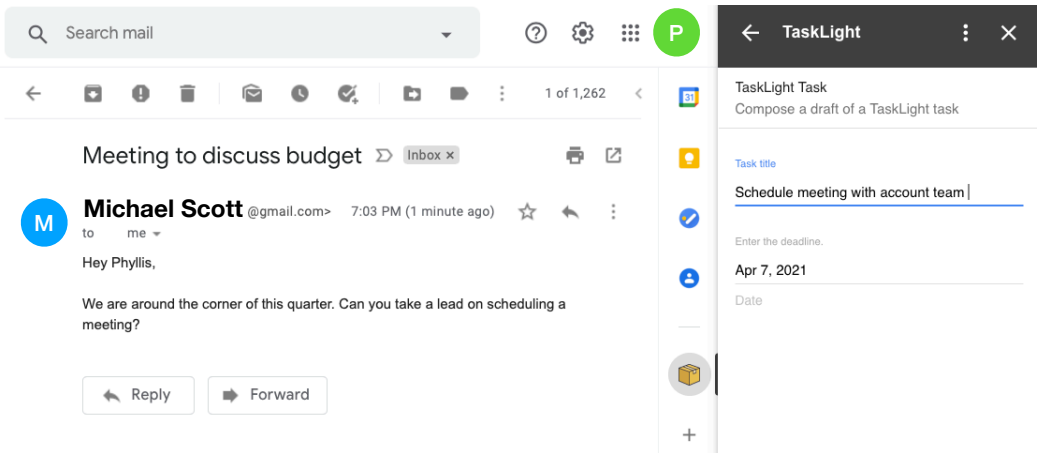

Figure 6-5: TaskLight allows users to draft requests from their inbox: an add-on sidebar is provided for the inbox and for shared documents, so that users can quickly draft a request. Once the request is drafted, it appears in TaskLight with the `status` "Drafting".

6.2.1 Drafting feature from email and team resources

Previous work found that physical paper remains an attractive medium because of its ubiquity and that people like the process of *scribbling* something down [41]. We were inspired by the ability of scribbling down to-do items anywhere and later aggregating them in one list. We provided a sidebar add-on for different clients (email, shared documents, shared slides) that allows users to immediately "scribble" tasks to their task list while maintaining the context of the client. For example, Fig. 6-5 shows the TaskLight sidebar that users can access by clicking on the *box* icon; users can then write a request as a draft while viewing their email message. The draft of their request will automatically appear in their task list. In this way, users can easily create tasks while maintaining the context of other platforms like email, where much task-related information is often found.

6.2.2 Unique chat channel per task

One difficulty of request management is that requests are lost in streams of discussion [68] or scattered in multiple channels [41]. TaskLight includes a dedicated chat channel for each task to include a free-form way of communication. Each chat chan-

nel is unique to the task. In this way, chatting about one task keeps the details and context of the task specific and avoids conversation clutter regarding irrelevant tasks. This makes it easier for individuals to consolidate task information in one place.

6.3 Features for making task management more personal

6.3.1 Suggestion model

TaskLight presents *collaborative task construction* to facilitate discussion around requests. Just like in collaborative writing, request stakeholders can make and review suggestions to make a request more concrete and satisfiable for all parties. Stakeholders of the request include those directly involved in it, the requesters and performers, but may also include individuals who want to be aware of the state of tasks but are not explicitly requesters or performers of it. In TaskLight, requesters have *edit* privilege, performers have *suggest* privilege, and remaining stakeholders have *view* privilege. Together, performers can make suggestions to adjust a request and requesters have control over changes. Remaining stakeholders can witness the changes and receive updates on the task. Using suggestion features, performers can perform common interactions around requests, such as delegate requests to others (i.e. suggestion on the `performers` field), negotiate deadline (`deadline`), tune task specification (`action items`), and solicit more information from requesters (`custom field`). TaskLight also supports quoting individual fields in a chat interface, so they can have specific discussions around a single field of the request. For example, if the performer wants to negotiate deadline, they can hover over the `deadline` field, and click `Quote` which will show the field and its value in the chat, so that both the performer and requester can have a discussion on that specific field. The suggestion model allows performers and requesters to have a human discussion directly on the structure of the request, which adds a personal dimension to it since it reflects the needs of both parties.

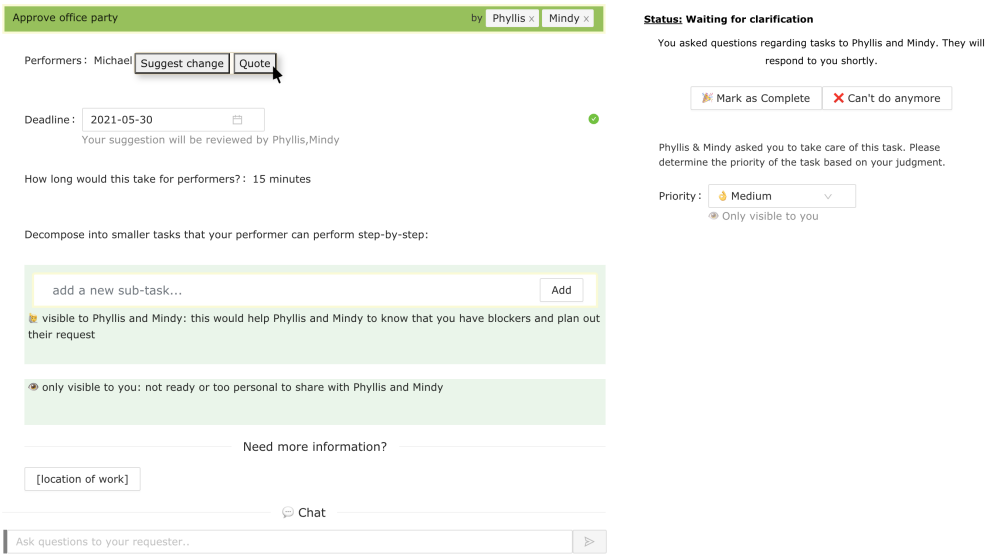

Figure 6-6: The review interface of TaskLight for performers. Performers can make suggestions on requests and ask for more information or clarification if necessary.

6.3.2 Narrative Guide

As another guidance and feedback to requesters, TaskLight supports Narrative Guide. Narrative Guide constructs a "fictional" text of a message (text that will not actually be sent or shown to performers) based on the task metadata that the requester has already filled in (Fig. 6-2). Using Narrative Guide, requesters can visualize how each field can be used and how it can help performers. As the requester updates the value of the field, Narrative Guide is updated accordingly. As the requester authors a request, using Narrative Guide, they may find fields that have not been specified yet, but could be useful to specify. Accordingly, users can easily add more fields by clicking on embedded buttons within the text. For example, if the requester would like to add a location for where the task should be completed, they can click on `[location]` (Fig. 6-2 #4), and the field will be added to the main editor (Fig. 6-2 #1). Narrative Guide encourages requesters to add detailed information as if they're sending a request to a friend.

6.3.3 Private fields

TaskLight allows performers to have private fields of `priority` and `action items`. These features are motivated by findings from our formative interviews on how people desire freedom to work on their own and by previous research on how individuals feel more comfortable working on their own [63]. TaskLight allows performers to set a `priority` for the task they've been assigned that is only visible to them; hence, they can comfortably prioritize tasks according to how they want without feeling like they're being *observed* by their requesters. Additionally, performers can add `action items` that default to only being visible to them. This feature can be used when performers would like to set personal action items for themselves, but would not like requesters to see each step in progress. We've provided performers the flexibility to make individual action items visible to the requester if they desire to share progress or show task blockers.

6.4 Implementation

TaskLight is a React application with Ant Design [2] and Firebase Firestore. For the drafting feature and importing file structures of Gmail, Google Doc, and Google Slides, we used Google Workspace and Sidebar APIs.

[2] https://ant.design/

7. Use Cases

We present use cases of TaskLight in this section of common scenarios in collaborative settings – collaborative writing, getting approval, and making group decisions.

7.1 Collaborative writing: Request drafting & distribution

TaskLight can facilitate request drafting and distribution from a team's shared resources (e.g. Google Drive shared folder). Task distribution is a frequent team activity that occurs during meetings. However, it is each meeting attendee's or meeting organizer's job to remember and transport their assigned task to a task list. TaskLight can make this easier by letting requesters create requests directly from the shared document. Fig. 7-1 illustrates the workflow. The requester can create a request draft on the meeting log and can later finish drafting at the TaskLight authoring interface. In this example, a team is working on comments made by paper reviewers on their submission and dividing who is going to be in charge of which comments. Phyllis (requester) oversees this process and ensures that the team members resolve all the concerns from the reviewers. She creates a draft request from the meeting log directly and continues to work on the request later. TaskLight imports the document structure of the meeting log, and the requester can make action items from that. In TaskLight, Phyllis adds the action items directly by clicking the + buttons in the related team documents section of the request.

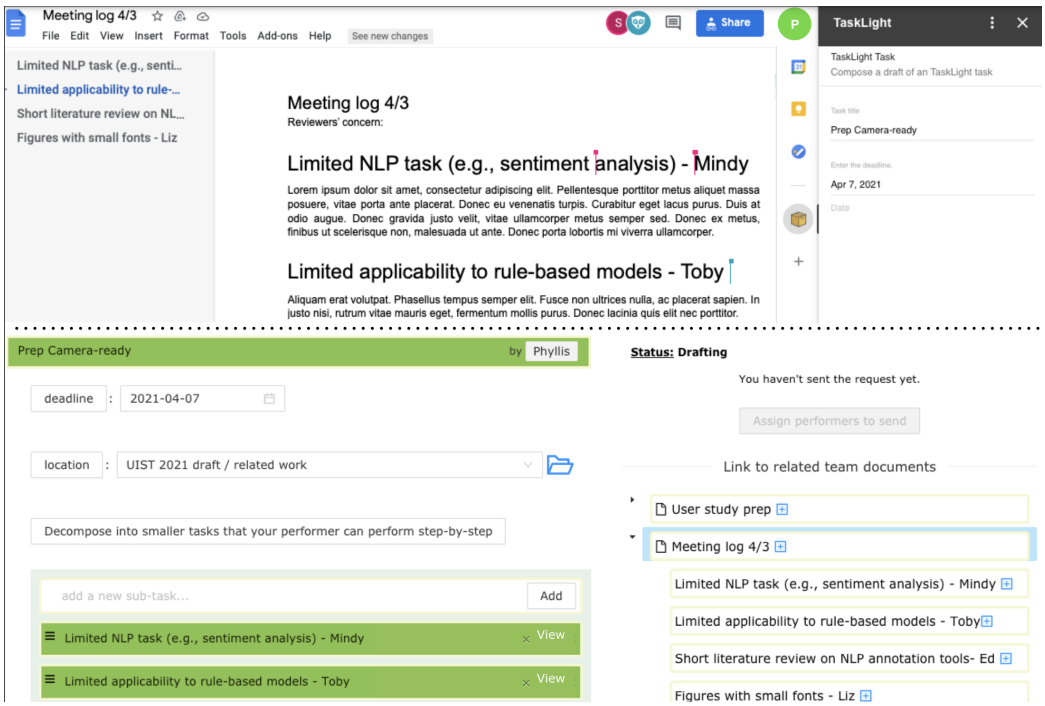

Figure 7-1: (Top) Meeting log used during discussion of request distribution. A requester can create a draft of requests using an add-on interface. (Bottom) In TaskLight, the requester can finish authoring the request by importing the meeting log as action items.

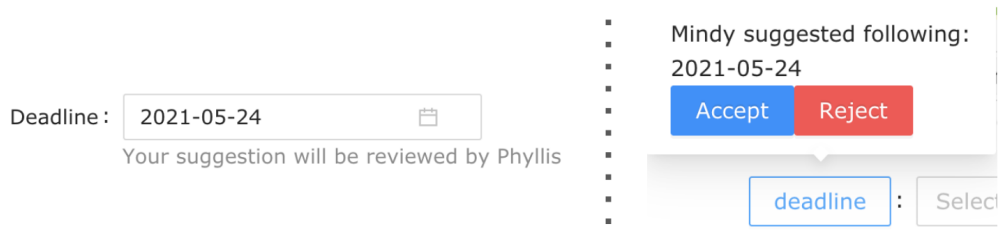

Figure 7-2: (Left) A performer offers suggestions to reflect the changing needs of a team. (Right) A requester reviews the suggestion.

7.2 Getting approval: Task construction and iteration

TaskLight allows users to easily iterate over content by providing features for performers to offer suggestions (i.e. changing deadlines, adding custom fields). Such a feature is integral to request management as teams undergo frequent change. Take for example that Phyllis (requester) would like to hire an intern, but needs to get approval from her boss Mindy (performer). Mindy accepts the request, but would like to push back the deadline since she has a busy week ahead. She suggests a change on the deadline field (shown in Fig. 7-2. She is told that her deadline change will be sent to Phyllis as a suggestion and be reviewed by her. The two can move forward as they construct the task together.

7.3 Group decisions: Disclosing request motivation and stakeholder progress

TaskLight provides an option to expose other performers' statuses to each other to invoke *stigmergic* effects. Stigmergic effects have positive impacts on team work; exposing one's activity motivates other collaborators to work [15]. When there are

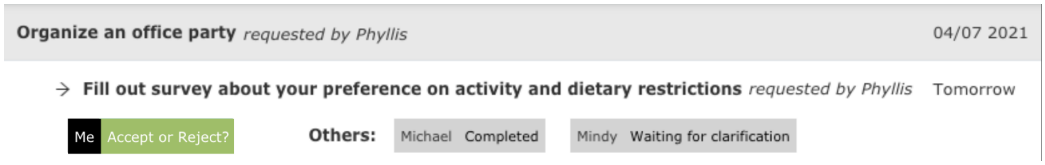

Figure 7-3: A performer's view on requests with multiple-performers requests. TaskLight shows other performers' progress and de-highlights the request without required actions from the user at the time being so that the user can manage their attention.

many performers for a request, requesters can track each performer's progress at a glance, and performers are also motivated to make progress. In Fig. 7-3, the user can see that they are invited to perform the sub-task "fill out survey" of the main task "organize an office party". The interface indicates that there is a pending request for the user; the request is a prerequisite of the party event, which is currently blocked due to in-completion of the sub-task. The user sees other performers have made progress on filling out the survey and can be motivated to do the same.

8. TaskLight Deployment User Study

We conducted a deployment user study to understand how the TaskLight interface fares when people utilize it for collaborative task management. We assigned participants into pairs, where they alternated roles of being requesters and performers throughout the study. Results of our study show that participants found TaskLight to be a tool that consolidates information well, provides good features for request authoring and tracking, and personalizes tasks. All 12 participants mentioned they would use TaskLight again for collaborating on small-scale projects with colleagues.

8.1 Study Design

Participants We recruited participants via posting on community-wide mailing lists of a private university. Each participant was asked to sign up for the study with a friend that they had collaborated with before. Each participant was compensated $30 individually ($60 as a pair) for their time. Fourteen participants, who comprise of students and/or alums at the university, were recruited and participated in an introductory tutorial session. By the end of the study, due to dropouts, 12 participants (10 females, 2 males, mean age=20.67) remained. We considered participants to be dropouts if they failed to request more than two tasks assigned to them. Since we had informed all participants that they would have to finish all the tasks by the end of the study in our introductory session, it is valid to consider teams that did not adhere close to that protocol as dropouts.

Study protocol Participants were invited as a pair to a 30 minute long tutorial session a day before the start of the study. During the session, participants were given a video tutorial on how to use TaskLight, a practice run-through of assigning and performing tasks with their partner, and set-up processes of assigning TaskLight as their homepage for ease of access and the installation of the TaskLight sidebar feature for their Gmail. Participants were instructed that when playing the role of requester, they would receive emails containing the task that they had to send to their study partner. When playing the role of the performer, participants were instructed that they would have to periodically check the TaskLight interface for the appearance of newly assigned tasks to them. The study was conducted for 4 days and each request was sent via email to one partner (Person A or Person B) in the pair each morning. (There were days where multiple requests were sent in the morning - i.e. Day 1: Person A of each pair was emailed request 1 and Person B of each pair was emailed requests 2-4). Emails that contained the requests for participants to send to their partners contained images of the request rather than raw text to prevent participants from directly copying and pasting the request into the interface. Each request was written as a free-form paragraph of text with a specified deadline. In fact, we sent the same requests that were sent to the text group in the request management field study (except the first one) that can be found in Appendix A.1.1.

At the end of the study, we solicited participants as a pair to do a 30 minute exit interview with us. In the exit interview, participants were asked a series of questions about their experience using TaskLight. The questions asked can be found in Appendix A.2.1.

8.2 Study Results

8.2.1 TaskLight Usage Statistics

Table 8.1 shows the statistics of how often participants interacted with TaskLight during the study. Study results show that on average participants visited the Task-

LIGHT interface more than once per day, spent about 5 minutes each time they visited, and actively clicked on an element within the interface about 11 times each time they visited. Among the 12 participants, a range of browsers and devices were used. Among the browsers, Chrome (14) was most used, followed by Safari (2), and then Firefox (1). Among the devices, OS_X (7) was most used, followed by Windows (6), then Linux (3), and then Android (1). Some participants used multiple browsers and devices to interact with TASKLIGHT, which explains why the total count exceeds 12.

Table 8.1: Usage statistics of the deployment user study. Participants in the study logged on 2.5 times on average per day, spent 5.2 minutes on average in each session they were logged in for, and actively clicked on an element in TASKLIGHT 10.7 times on average per session.

Day	Aver. no. of sessions per day	Aver. time spent per session (mins)	Aver. no. of clicks per session
1	2.8	4.8	7.3
2	2.8	2.2	6.2
3	1.75	7.6	19.1
4	2.5	6.2	10.3
All days	**2.5**	**5.2**	**10.7**

8.2.2 Exit interview Feedback and Findings

We conducted 30-minute exit interviews with 6 teams who utilized TASKLIGHT during the study. Overall, participants found the tool to be pretty intuitive, easy to navigate, organized, and usable. *"The overall felt like it was like pretty usable I like pretty much you [knew] w[h]ere everything was supposed to go."*.

Participants felt that TaskLight made requests feel human and more personalized. Participants mentioned that compared to chatting tools, TASKLIGHT offered similar communication but provided more helpful structure: *"the things that I do use are essentially equivalent to speaking to them in person right like just sort of freeform messages...And I didn't feel like TaskLight got in the way of that kind of communication and it does provide a lot of features. That sort of helps people. Like builds a request, like all of the fields that specify, you know when do you want this done by and things like that. And that felt intuitive to us[e]"*. Participants especially appreciated how TASKLIGHT was modeled over the notion of a request: *"I think*

the way that it's phrased in [TaskLight] is helpful in terms of something like being a request that the other person has the power to accept or decline". Additionally, when asked about the effectiveness of the `Narrative Guide`, participants reported that it felt personal and provided good context for fields, and it also served as a good reminder of how to form a helpful request. *"I would look at it [Narrative Guide] and see oh location just click on location and then type it in really quickly...it would just remind me of what I need to put in"*.

Participants shared positive feedback on default fields and tracking mechanisms with desire for improvements. Participants made good use of the default fields, especially `deadline` and `location`, but wished there was a field to provide a general description or notes for the task. The lack of this field led participants to use more free-text features like the `why` field or the chat interface to communicate general task description. *"I think the one thing I wish were a field that I could use is like requirements or sort of like overall description of what to do, we generally sent basically the contents of your email in the chat"*. When asked why the `action items` field was not utilized for that purpose, participants responded that `action items` felt like the task had to be explicitly broken up into smaller tasks that need to be done whereas a general description is : *"not like a thing that you do"*.

For tracking features, participants shared positive sentiment on the overall task list view, mentioning how highlighting/de-highlighting features for blocked tasks were helpful and how the *"task statuses being little boxes is very helpful for me in terms of knowing where to look"*. However, there was desire for more discernible colors for different status labels: *"I didn't think the colors were like distinct or like that there were enough categories of color...having accepted be the same color as completed was confusing for me, because there were actually like things that I needed to do on that particular task"*. Lastly, participants said that they wished there was less screen real estate used for tasks in the list, especially since later in the study, the number of tasks accumulated: *"it's this big list, whereas like the to do list that I'm used to...it's all in a small little column, and I can just glance at it and read"*.

Participants thought TaskLight consolidated task information well and

offered suggestions for more integration. Overall, participants stated that TaskLight felt *"like a central hub for all things that I might have to do"* and categorized information well. Participants especially provided positive feedback on the unique chat channel for each task, stating that it provided *"a distinct chat per project, instead of like you know if you're communicating with somebody like just like over some regular messenger like you don't have a distinct separation like per project"*. One participant shared how the distinct separation per project was especially notable since it created a division between work life and social life: *"it won't really mix like the social aspect of someone's life with the work aspect of someone's life..., which always like is a problem, whenever I'm like sending people information over messenger or email"*.

Aside from the distinct chat interface, participants also made good use of the `location` field, citing how it was a convenient way of linking relevant documents: *"the location thing was also really, really useful. That was probably my favorite feature because it's I especially when I'm thinking about working with my clubs, it's always a hassle to go find look through my folders and then find their correct document for link so and I know like whenever I asked other people to do things it's it's it's an extra like activation energy"*.

During the study, we also asked participants to incorporate a TaskLight sidebar feature that integrated into Gmail, which participants found helpful because they could draft tasks while maintaining the context of email: *"the email was right there and then I could just type it in"*. There was, however, a desire to improve the sidebar feature since the sidebar only supported fields for `title` and `deadline`. Some participants found that the limited number of fields in the sidebar rendered the feature useless since they had to add more details to the task later: *"it'd be really useful if you could also put in information about what you want them to do, because what I would end up doing is putting it in and then going to TaskLight later and then having to open back up the email"*.

Lastly, some participants stated how they could see future integration with other platforms on TaskLight, such as integrating with GitHub for coding-based projects

where users could *"more easily like find specific files or something in a github repository...allows you for line highlighting, so you could sort of, say, like hey this method is bad like fix it"*.

All 12 participants said they would use TaskLight again, but with a few constraints. The top two reasons why participants stated they would continue to use TaskLight are because it: 1) feels centralized and 2) feels like a convenient, intuitive set of features. *"It was just like a very convenient and like set of things that are like you know generally relevant to tasks like that that like I just like haven't had a system provide before"*. That being said, participants cited a few constraints on which they'd continue to use it. First, participants stated that it would be difficult to get everyone in their group to use it, which would hinder their personal usage: *"I don't think many people that I work with would be down to move and learn a whole other app...I think after we get over that initial like bump of like teaching them the interface, it might be something really useful I can imagine it being used by some of my clubs"*.

Second, participants mentioned that the absence of notifications would make it difficult to continue using TaskLight. During the study, some participants ended up using chat platforms in addition to the TaskLight interface to bump or inform their partners on task updates. However, when asked what the content of the message they sent to each other entailed, they responded that it was a simple message like *"I left you something in TaskLight"*. Thus, because our main goal was to consolidate discussion around the task, and discussions about the actual task did not occur in the external chat platforms, we were still successful in achieving our goal. When notifications are supported by TaskLight, the final dependency on chat platforms will not be needed.

Third, participants mentioned that they would use TaskLight for smaller group projects with multiple tasks, but not large-scale projects. *"Like when it comes to larger projects for me,...in terms of having to make the sub tasks and stuff like it seems a little bit too much of a burden right now and tasks like to do it, so I prefer to have a software that made [it]...Or, I want to look at it in different views, because, it's*

a really big project and it's just too much to look at it as a list". Participants stated that larger projects entail a larger number of tasks and that TaskLight's current task list view would easily get overwhelmed for something large-scale. However, for smaller group projects, many participants shared that they would feel comfortable using the interface as is because *"it provides more structure and like ways to go back and check your history"*.

Lastly, participants mentioned that they would not use TaskLight for managing personal todos (with no collaborators) because it feels unnecessary. Participants implied that it would be heavyweight to add structure for their personal tasks when they are generally aware of personal things they need to do: *"I use Google calendar which, and I just like schedule things in which works for me and I usually memorize the locations of my documents...so like the linking it was actually just would take longer because I already know where it is and can search it up real quick"*.

Overall, participants are willing to use TaskLight for smaller-scale projects with team members who have learned how to use the interface and are willing to use it.

9. Discussion

We presented our collaborative request management tool TaskLight. Our design was grounded on previous literature regarding task management and request management as well as our interviews, formative study, and deployment user study. Here, we share our evaluation of our design. We also discuss the potential of curated information from requesters as a new way to improve attention management and tighter connection between personal and collaborative task management.

9.1 Cognitive dimensions of notation

We use the Cognitive Dimensions of Notation [9], which provides a set of heuristics for evaluating interfaces, to analyze TaskLight as a tool. From the 14 heuristics, we evaluate our system against a relevant subset of 4: *Viscosity*, *Visibility*, *Premature Commitment*, and *Hidden Dependencies*.

Viscosity (resistance to local change). In terms of task construction, TaskLight is not viscous as the amount of effort required for a user to create a task, add fields to it, and send it is encapsulated as a single click per action. Task iteration (the process of adding changes to the original task via suggestion and/or discussion) however requires a series of steps and tends to a higher viscous side than task construction. Namely, to offer a suggestion, the user must hover, then click `Suggest`, offer a suggestion, and observe a change in their task status. Requesters must then approve the suggestions for the task to be updated. Future work may focus on different permissions requesters can assign performers (i.e. direct edit access v. suggestion access) to make task iteration less viscous for requester-performer relationships of equal power.

Visibility (ability to view things easily). TaskLight offers a task list to view all of the user's tasks while having visibility on the action the user must take on each. Status labels are provided for both the requesters and performers to easily see progress on the task. Task rows become highlighted in white and status labels in particular turn green when action is required, increasing visibility to the user. Clicking on a status label renders the task view, which is originally hidden from users. Here, visibility is slightly compromised as the only indication that users can access the task view is through the change in cursor when hovering over a status label. However, change in cursor from the default to a pointer is familiar to users in modern web practices as an indication for an action to take place.

Premature Commitment (allowing for well-thought decisions). TaskLight first shows the task list before the user makes the decision to see more details on it. Rows that are grayed out in the task list mean that the user is blocked on the task and that their attention is not currently needed on it. This helps users in making decisions related to what tasks they should prioritize and click on. Clicking on the task triggers a detailed view of the task where users observe fields related to the task (i.e. deadline, anticipated completion time). Users are provided this information as they are shown the `Accept` and `Can't do` buttons, so they can make informed decisions on the request.

Hidden Dependencies (visibility of dependent entities). Because TaskLight defines a request as a shared entity between a requester and performer, changes that either user makes reflects on both users' sides. For example, if the performer asks a question to the requester, the performer will see a change in their status as `Waiting for clarification` while the requester will see `Answer question or suggestion`. Therefore, dependencies in the system are reflected well.

9.2 Opportunities for more intelligent task notification

Participants from our deployment user study mention the need for better notifications. Previous work on task notification advances the timing of notifications detecting breaking points of each task [28, 14, 26, 45]. This notifies users about upcoming tasks in a more situated and less interrupting manner [37]. Additionally, feedback from the post-surveys in our field study show that when asked about the help they offer as requesters to performers, most participants care about minimizing workload for their performer. We envision that TaskLight can provide opportunities of improving notifications to lessen burden on performers. In TaskLight, requesters are providing information to solicit and help performers to work on requests. For example, in our field study, participants remarked how anticipated completion time and broken down steps of requests were helpful for them to schedule their day around request management. We can imagine that the information provided by collaborators who likely share context with performers can improve the timing of the notification. As future work, we plan to leverage the curated information of requests into notification. With calendar integration, TaskLight can make a situated notification, such as reminding about the requests that could be done by the next schedule in the user's calendar or warning a user when time until the deadline is equal to the request's anticipated completion time.

9.3 Tension between personal task management and request management

Even if requests *made by colleagues* inevitably have collaborative natures, there is a noticeable gap existing between research discipline and tools [17] as also observed in our field study where we encountered a few occasions where participants liked to keep their own list of tasks. Previous research hints at reasons behind this; people

like to keep work-in-progress just for themselves [63] and customize the way they keep tasks [3]. It is natural for individuals to govern their tasks as they have been doing; in contrast, team collaboration norms are more recent requirements made up for a team. This led us to think that "Is it even right to force members to use a team workspace?" and "Is there truly one workspace that fits all of the team members' preferences?" Modern PM tools tend to assume a single team space, while TaskLight has no notion of a team.

There are different request management methods. Namely, there is a model of email, in which requesters have no access or vision as to what performers are up to. On the other end, there are PM tools, which provide full disclosure and great visibility to requesters but no privacy for performers. Due to the pros and cons, each user might have different preferences of request management tools. We argue that individual users should be supported to pick their own preferred request management tools. We propose a *data stream* between different task management tools, a data structure that could be consumed and understood by different request management applications so individuals can keep managing their tasks in their preferred and personalized manner; at the same time they can take advantage of disclosing their work progress and provide room for requesters to know the performers' progress. To merge and collect individual work in one workspace, we can incorporate a *pull request* model which lets each member add their work to the shared workspace whenever they are ready to [47]. As future work, we plan to investigate this idea in TaskLight, which lets users export and import requests in various mediums with their customized formats, such as to their notepad as plain-text, inbox, or calendar.

10. Limitation & Future work

Although TaskLight targets request management, which is a universal problem in team settings, our interview, field study, and deployment user study are based on populations from academia. We attempted to draw our insight from participants of different team natures, such as officers from student government, members of diversity initiatives, and students doing course projects. It still remains as a future work to see if our finding expands to other populations. We next aim to deploy our system on a larger level in different backgrounds and examine how our system situates for various teams.

We also plan to support affordances on customizing task delegation and scheduling based on the culture of each organization. In conference-paper review requests, there is an implicit policy of "if you decline, then suggest other potential reviewers". In the future version of TaskLight, a requester can customize each request to be managed accordingly. For example, if a performer has declined the past 3 requests, then they need to explain to requesters why they cannot take this request again. Or for critical requests, performers could be asked to give at least a one day notice if they cannot finish the request on time.

11. Conclusion

We presented our tool TaskLight as a solution towards collaborative request management. The design of our tool was motivated by formative interviews on request coordination in teams and a request management field study. We found that current practices of making and constructing requests face numerous challenges ranging from being impersonal, feeling overwhelming, and being too scattered. We seek a form of request management that utilizes flexible, lightweight structure that feels personal and entails less mental load that requesters can use to easily curate information for requests and that performers can also benefit from for accomplishing requests. Accordingly, we design TaskLight as a stepping stone to an interface that can achieve this. Our deployment user study revealed that TaskLight fares well as a request management tool that is intuitive for users and that would be best situated for use in small-scale projects. In the near future, we plan to deploy TaskLight to various teams of individuals to run a full-scale evaluation study. In the meantime, we hope that the needs and findings discovered in our formative interviews and studies can guide designers and developers of request management systems.

Appendix.

A.1 Request management field study

Each version of participants were delivered each request at the same time. For providing comments on writing requests, we enforced participants to give comments based on vocabularies of Kaur et al. [31].

A.1.1 The text group

The following requests were distributed via Google spreadsheets.

- Hi, can you invite our gmail addresses to a new Google drive folder as editors? Please do it by tmrw noon. Thanks!
- Hey, let's have a sync-up. Can you create when2meet or Doodle for next weekday, put your availability and share with the team by tomorrow?
- Hi, can you write a brief review on how [the university]'s policies compare to other colleges' policies on maintaining undergraduate social life? It needs to be between 150 and 200 words. I need this done by Saturday afternoon.
- Hi, we would like to do some coverage on social events at [the university] during Covid. Can you send me a list of virtual social events happening this week? I need 3 of them in the next 2 days.
- Hey, can you take a look at my write-up about the "Covid symptom study" and provide comments? I put a document in our Google drive folder. Can you

do this by Monday?

- Hey, can you help me think of events to help students socialize based on your observation on [the university] & the university Covid policies? Please brainstorm what kind of virtual social events you think students would like and make a flyer out of it. The flyer should have specific details about the social events. I put a flyer template in our Google drive folder. Please send this to me by tomorrow.

- Hey, can you take a look at my write-up about the "Edit-a-thon" and provide comments by tmrw? I put a document in our Google drive folder.

- Hey, I also did some research on other universities' COVID policy. Can you take a look? I put a document in our Google drive folder. Can you do this by tmrw afternoon?

- Hey, I saw your flyer. Thanks! Just wondering if you can make it more accessible. Here's a guideline to make an online document accessible: [link to accessibility guideline] Please make changes based on the guideline and annotate the changes you made by the end of today.

A.1.2 The PM tool group

The PM tool group were provided priority (Medium, High, or Urgent), status (initially set as "Not Started"), deadline and requester column.

A.1.3 The TaskLight group

The TaskLight group were given the following contents (title, steps, the purpose of the task, a location of request, anticiated completion time) and a deadline column:

- Invite [authors] to new Google drive Folder / - Create a new folder at Google drive / - Invite [authors] as editors / We will use this folder for teamwork so make

sure we can edit files in the folder / https://drive.google.com/drive/u/0/my-drive / 5 mins

- Schedule a meeting / - Create a new When2meet for next weekday and put your availability - Share with [authors] / - / https://www.when2meet.com / 5 mins

- Write a review comparing [the university] and other colleges' policies on maintaining undergraduate social life during Covid. Please talk about the following topics in your review: - quality of social life - events organized by the college - Covid standard procedures

  The length of the review needs to be between 150 to 200 words in total. / We'll be brainstorming social events based on your research. / [link to a Google document] / 20 mins

- Find 3 virtual social events that are happening this week at [the university] / Here's a list of good starting points to look for events: - Check out your Facebook and look at upcoming events/ what's happening near you - Check your email to see if there are events planned by your living group - Copy the information of the task at the Google Doc / We are going to plan our own virtual social events and want to get some inspiration / [link to a Google document] / 10 mins

- Provide comments to my write-up on the "COVID Symptom Study" / - / I want someone else to run a quick eye over my write-up and see if it makes sense! / [link to a Google document] / 15 mins

- Brainstorm your desired COVID-friendly social events and make a flyer of it / Based on your earlier research about [the university]'s social life policies, we know what events we can hold and get students to participate! I took a stab at the flyer and listed what information is needed, so you can just fill in! / - / [link to a Google document] / 15 mins

- Provide comments to my write-up about "Edit-a-thon" / - / I'm planning to organize an Edit-a-thon event. I want to write a post about what is Edit-a-thon. Before that, can you take a look at my writing? / [link to a Google document] / 15 mins

- Provide comments to my universities' COVID polices / - / I want someone else to run a quick eye over my write-up and see if it makes sense! / [link to a Google document] / 15 mins

- Make the flyer more accessible / "- Read the guidelines on how to make an online document more accessible - Then modify your flyer according to the guidelines. For example, here is how to add alt-text in Google Slides:

  https://support.google.com/docs/answer/6199477?hl=en - Please annotate

  changes you made as comments in your flyer" / - / accessibility guidelines: [link to an online accessibility guideline] / 5 mins

A.2 TaskLight Deployment User Study

A.2.1 Exit Interview Questions

- Overall, please describe your experience using TaskLight.

- Features: Providing default fields, Narrative Guide, Status Labels and Highlighting or De-highlighting tasks, Suggestion Feature, Sidebar Feature

  - How did you use this feature?
  - How often did you use it?
  - Prior to TaskLight, how did you perform similar actions?

- What do you think about how the information or features are laid out?

- If you could change one thing about TaskLight, what would it be? And why?

- How would you compare using TaskLight to using other platforms including email/chat? Are there scenarios in which you'd use one over the other when requesting tasks?
- Would you continue to use TaskLight to manage tasks in the future?
  - If yes, why? What did you like about it?
  - If no, why? What's stopping you from using it in the future?

Bibliography

[1] Maruthi Rohit Ayyagari and Issa Atoum. Understanding customer voice of project portfolio management software. *International Journal of Advanced Computer Science and Applications*, 10(5), 2019.

[2] Jakob E Bardram. Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(2):1–36, 2009.

[3] Victoria Bellotti, Brinda Dalal, Nathaniel Good, Peter Flynn, Daniel G Bobrow, and Nicolas Ducheneaut. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 735–742, 2004.

[4] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 345–352, 2003.

[5] Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. The project fragmentation problem in personal information management. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 271–274, 2006.

[6] Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. Improved search engines and navigation preference in personal information management. *ACM Transactions on Information Systems (TOIS)*, 26(4):1–24, 2008.

[7] Ofer Bergman, Noa Gradovitch, Judit Bar-Ilan, and Ruth Beyth-Marom. Folder versus tag preference in personal information management. *Journal of the American Society for Information Science and Technology*, 64(10):1995–2012, 2013.

[8] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322, 2010.

[9] Alan F Blackwell, Carol Britton, A Cox, Thomas RG Green, Corin Gurr, Gada Kadoda, MS Kutar, Martin Loomes, Chrystopher L Nehaniv, Marian Petre, et al. Cognitive dimensions of notations: Design tools for cognitive technology.

In *International Conference on Cognitive Technology*, pages 325–341. Springer, 2001.

[10] Richard Boardman and M Angela Sasse. " stuff goes into the computer and doesn't come out" a cross-tool study of personal information management. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 583–590, 2004.

[11] Marcelo Cataldo, Patrick A. Wagstrom, James D. Herbsleb, and Kathleen M. Carley. Identification of coordination requirements: implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work - CSCW '06*, page 353. ACM Press, 2006.

[12] Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. Break it down: A comparison of macro-and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4061–4064, 2015.

[13] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. Calendar. help: Designing a workflow-based scheduling agent with humans in the loop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2382–2393, 2017.

[14] Casey Dugan, Aabhas Sharma, Michael Muller, Di Lu, Michael Brenndoerfer, and Werner Geyer. Remindme: Plugging a reminder manager into email for enhancing workplace responsiveness. In Regina Bernhaupt, Girish Dalvi, Anirudha Joshi, Devanuj K. Balkrishan, Jacki O'Neill, and Marco Winckler, editors, *Human-Computer Interaction - INTERACT 2017*, Lecture Notes in Computer Science, page 392–401. Springer International Publishing, 2017.

[15] Mark Elliott. Stigmergic collaboration: The evolution of group work: Introduction. *m/c journal*, 9(2), 2006.

[16] Thomas Erickson. From pim to gim: personal information management in group contexts. *Communications of the ACM*, 49(1):74–75, 2006.

[17] Danyel Fisher. *Social and temporal structures in everyday collaboration (dissertation)*. PhD thesis, 2004.

[18] Danyel Fisher, AJ Brush, Eric Gleave, and Marc A Smith. Revisiting whittaker & sidner's" email overload" ten years later. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 309–312, 2006.

[19] Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd. Computer systems and the design of organizational interaction. *ACM Transactions on Information Systems (TOIS)*, 6(2):153–172, 1988.

[20] Freshdesk, 2010. https://freshdesk.com/.

[21] Victor M González and Gloria Mark. " constant, constant, multi-tasking craziness" managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, 2004.

[22] Catherine Grevet, David Choi, Debra Kumar, and Eric Gilbert. Overload is overloaded: email in the age of gmail. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 793–802, 2014.

[23] Jonathan Grudin. Why cscw applications fail: problems in the design and evaluationof organizational interfaces. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 85–93, 1988.

[24] Beverly L Harrison, Alex Cozzi, and Thomas P Moran. Roles and relationships for unified activity management. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 236–245, 2005.

[25] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[26] Eric Horvitz, Johnson Apacible, and Muru Subramani. Balancing awareness and interruption: Investigation of notification deferral policies. In *International Conference on User Modeling*, pages 433–437. Springer, 2005.

[27] Julie S Hui, Darren Gergle, and Elizabeth M Gerber. Introassist: A tool to support writing introductory help requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[28] Shamsi T Iqbal and Brian P Bailey. Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 93–102, 2008.

[29] William Jones, Abe Wenning, and Harry Bruce. How do people re-find files, emails and web pages? *iConference 2014 Proceedings*, 2014.

[30] David R Karger. Creating user interfaces that entice people to manage better information. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1–2, 2011.

[31] Harmanpreet Kaur, Alex C Williams, Anne Loomis Thompson, Walter S Lasecki, Shamsi T Iqbal, and Jaime Teevan. Creating better action plans for writing tasks via vocabulary-based planning. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–22, 2018.

[32] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moontae Lee, Florian Schulze, and Scott R Klemmer. Taskgenies: Automatically providing action plans helps people complete tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(5):1–25, 2013.

[33] Nicolas Kokkalis, Thomas Köhn, Carl Pfeiffer, Dima Chornyi, Michael S Bernstein, and Scott R Klemmer. Emailvalet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1291–1300, 2013.

[34] Kum-Yew Lai, Thomas W Malone, and Keh-Chiang Yu. Object lens: a "spreadsheet" for cooperative work. *ACM Transactions on Information Systems (TOIS)*, 6(4):332–353, 1988.

[35] Wayne G Lutters, Mark S Ackerman, and Xiaomu Zhou. Group information management. *Personal information management*, pages 236–248, 2007.

[36] Thomas W Malone, Kenneth R Grant, and Franklyn A Turbak. The information lens: an intelligent system for information sharing in organizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, 1986.

[37] Gloria Mark, Daniela Gudith, and Ulrich Klocke. The cost of interrupted work: more speed and stress. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 107–110, 2008.

[38] Gloria Mark, Shamsi T Iqbal, Mary Czerwinski, Paul Johns, Akane Sano, and Yuliya Lutchyn. Email duration, batching and self-interruption: Patterns of email use on productivity and stress. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 1717–1728, 2016.

[39] Kate McCready and Claire Stewart. Sheriff, irs auditor, psychotherapist, hostage negotiator all in one: Project management for everyone. *At the Helm: Leading Transformation: The Proceedings of the ACRL 2017 Conference, March 22–25, 2017, Baltimore, Maryland*, page 597–603, Mar 2017.

[40] Thomas P Moran, Alex Cozzi, and Stephen P Farrell. Unified activity management: supporting people in e-business. *Communications of the ACM*, 48(12):67–70, 2005.

[41] Michael Muller, Casey Dugan, Michael Brenndoerfer, Megan Monroe, and Werner Geyer. What did i ask you to do, by when, and for whom? passion and compassion in request management. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1009–1023, 2017.

[42] Michael Muller, Casey Dugan, Aabhas Sharma, Werner Geyer, and Thomas Erickson. A stick with a handle at each end: Socially implicated work objects for design of collaborative systems. In *Proceedings of 15th European Conference on Computer-Supported Cooperative Work-Exploratory Papers*. European Society for Socially Embedded Technologies (EUSSET), 2017.

[43] Michael J Muller, Werner Geyer, Beth Brownholtz, Eric Wilcox, and David R Millen. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 375–382, 2004.

[44] Michael J Muller and Daniel M Gruen. Working together inside an emailbox. In *ECSCW 2005*, pages 103–122. Springer, 2005.

[45] Karen Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah L McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe. An intelligent personal assistant for task and time management. *AI Magazine*, 28(2):47–47, 2007.

[46] Soya Park, Amy X Zhang, Luke S Murray, and David R Karger. Opportunities for automating email processing: A need-finding study. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[47] Ei Pa Pa Pe-Than, Laura Dabbish, and James D. Herbsleb. Collaborative writing on github: A case study of a book project. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, page 305–308. ACM, Oct 2018.

[48] G. Pisoni and M. Hoogeboom. Investigating effective dynamics of virtual student teams through analysis of trello boards. In *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, page 639–643, Nov 2019.

[49] Daniela Retelny, Michael S. Bernstein, and Melissa A. Valentine. No workflow can ever be enough: How crowdsourcing workflows constrain complex work. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–23, Dec 2017.

[50] Bahareh Sarrafzadeh, Ahmed Hassan Awadallah, Christopher H. Lin, Chia-Jung Lee, Milad Shokouhi, and Susan T. Dumais. Characterizing and predicting email deferral behavior. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, page 627–635. ACM, Jan 2019.

[51] Peter Scupelli, Sara Kiesler, Susan R Fussell, and Congrui Chen. Project view im: a tool for juggling multiple projects and teams. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1773–1776, 2005.

[52] John R Searle. *Expression and meaning: Studies in the theory of speech acts*. Cambridge University Press, 1985.

[53] Shilad Sen, Werner Geyer, Michael Muller, Marty Moore, Beth Brownholtz, Eric Wilcox, and David R Millen. Feedme: a collaborative alert filtering system. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 89–98, 2006.

[54] O. Seppälä, T. Auvinen, V. Karavirta, A. Vihavainen, and P. Ihantola. What communication tools do students use in software projects and how do different tools suit different parts of project work? In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, page 432–435, May 2016.

[55] Preethi Srinivas. Modeling clinical workflow in daily icu rounds to support task-based patient monitoring and care. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, pages 105–108, 2015.

[56] Lucy Suchman. Do categories have politics? the language/action perspective reconsidered. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW'93*, pages 1–14. Springer, 1993.

[57] Lucy Suchman. Speech acts and voices: Response to winogradet al. *Computer supported cooperative work (CSCW)*, 3(1):85–95, 1994.

[58] Lucy Suchman. *Human-machine reconfigurations: Plans and situated actions*. Cambridge university press, 2007.

[59] Carlos Toxtli, Andrés Monroy-Hernández, and Justin Cranshaw. Understanding chatbot-mediated task management. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–6. ACM, Apr 2018.

[60] Rajan Vaish, Keith Wyngarden, Jingshu Chen, Brandon Cheung, and Michael S Bernstein. Twitch crowdsourcing: crowd contributions in short bursts of time. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3645–3654, 2014.

[61] Gina Venoglia, Laura Dabbish, J.J. Cadiz, and Anoop Gupta. Supporting email workflow, 2001. Last accessed 16 September 2017.

[62] Jürgen Vogel, Werner Geyer, Li-Te Cheng, and Michael Muller. Consistency control for synchronous and asynchronous collaboration based on shared objects and activities. *Computer Supported Cooperative Work (CSCW)*, 13(5):573–602, 2004.

[63] Dakuo Wang, Haodan Tan, and Tun Lu. Why users do not want to write together when they are writing together: Users' rationales for today's collaborative writing practices. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–18, Dec 2017.

[64] Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. Am i wasting my time organizing email? a study of email refinding. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3449–3458, 2011.

[65] Steve Whittaker and Candace Sidner. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 793–802, 1996.

[66] Terry Winograd, Fernando Flores, and Fernando F Flores. *Understanding computers and cognition: A new foundation for design*. Intellect Books, 1986.

[67] Zendesk, 2007. `https://www.zendesk.com/`.

[68] Amy X Zhang and Justin Cranshaw. Making sense of group chat through collaborative tagging and summarization. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–27, 2018.