"I really need your help with this work..": A System for Navigating the Tricky Terrain of Managing Up by Leveraging One's Motivation to Get Things Done

"I really need your help with this work..": A System for Navigating the Tricky Terrain of Managing Up by Leveraging One's Motivation to Get Things Done

SOYA PARK, Emory University, Atlanta, USA
STUTI VISHWABHAN, doc.ai, San Jose, USA
MICHAEL MULLER, IBM Research, Cambridge, USA
DAVID R. KARGER, MIT, Cambridge, USA

When people need help from their supervisors or peers, they often have to manage up to get things done. However, unlike managing subordinates (managing down), managing people of equal or higher status (managing up) are not obligated to help. These requests often involve collaborative tasks between requesters and performers. Through interviews, we found that these collaborative tasks require coordination work that is not materialized in existing management tools. We also found that requesters are willing to take on this coordination work to see their requests fulfilled. To address this issue, we propose a system called TaskLight, which allows requesters to handle coordination work themselves. For example, requesters can collect useful context and information for their performers. We conducted two deployment studies and found that TaskLight leads to better outcomes because requesters are able to assist performers more effectively. Our findings demonstrate a new way to reduce the social burdens of managing up and improve collaboration.

CCS Concepts: • **Human-centered computing → Asynchronous editors.**

Additional Key Words and Phrases: managing up, request management, personal information management, help-seeking, group work, groupware

1 INTRODUCTION

People frequently need to ask others for help to achieve their goals. In a workplace context, this direction happens in a top-down manner (managing down, i.e. traditional managing), where managers direct their reports to perform certain tasks. However, individuals often find themselves in situations where they need to direct their supervisors or peers (managing up [36, 47]). This can happen in collaborative projects as work is divided or delegated, or can involve asking someone for a favor such as writing a recommendation letter. Any such request defines a collaborative task. Collaborative tasks are often viewed through a lens of project management, in which work that must be done is assigned—by a project manager or collective decision making—to people who have to do it. However, favors, volunteering, advising, and mentoring differ in that people who have made the request do not have power over the person who fulfills the work. The *requester* is often the one with the motivation to see the task completed [6]; however, they lack various resources, such as knowledge [34, 64], social capital [35, 37], authority, or time to perform the task themselves. On the other hand, the *performer* is the one capable of actually

Authors' addresses: Soya Park, soya.park@emory.edu, Emory University, Atlanta, USA; Stuti Vishwabhan, svishwab@alum.mit.edu, doc.ai, San Jose, USA; Michael Muller, IBM Research, Cambridge, USA, michael_muller@us.ibm.com; David R. Karger, karger@mit.edu, MIT, Cambridge, USA.

ACM 1073-0516/2024/3-ART
https://doi.org/10.1145/3652603

completing the work. Due to the asymmetric nature of these tasks, requesters are sometimes put in a vulnerable position and feel anxious when they seek help [6, 80]. Marginalized users demonstrate a larger need for support relative to their peers. This is further exacerbated by the fact that these users more commonly lack social skills of help-seeking compared to these same peers [37].

In this work, we focus on the managing-up scenario. Central to our approach is capitalizing on the requester's drive to accomplish tasks. We aim to understand how requesters can optimize and refine the managing-up process, with an end goal of developing a functional system. We address two primary research questions:

RQ1 (Understanding Dynamics): How can we utilize requesters' motivation to enhance the managing-up process?
RQ2 (Interface Design): What design elements are pivotal for an interface that supports managing up?

Through our formative interviews, we discovered that requesters perceive existing collaborative task-management tools as inadequate for managing up. While they acknowledge the benefits of such tools for ease of tracking and efficiency [58, 61, 73], they still rely on email or instant messaging [35, 60]. Our interviews shed light on the gap between the tool and the dynamics of managing up, and suggest three design requirements:

- Reducing the burden of coordination work on performers: Collaborative task-management tools require substantial effort to keep track of the status of requests from the performers' end. It can be particularly challenging when the burden of coordination falls on the performer, even though it is the requester who is truly motivated to see the task completed.
- Shifting the burden of coordination work: Requesters are willing to take on the burden of coordination that will allow performers to complete the task. Therefore, they seek tools that enable them to take on the additional efforts *on behalf of* their performers.
- Assisting in bringing attention across different types of coordination: In the context of requesting help, where performers are doing work for requesters, performers often need to ask questions to ensure that they meet requesters' needs. Delays in getting feedback can slow down the pace of the task and make it difficult to maintain focus and engagement. To mitigate these challenges, the system could bring back task participants' attention in a timely manner.

Based on these requirements, we have developed a system that addresses the aforementioned design requirements and supports requesters seeking help from non-obligated performers. Our system leverages the requesters' motivation and uses it as a driving force for the coordination overhead in collaborative-task management. This enables the requesters to take control of the coordination work and utilize their understanding of task requirements and contexts to make the process more efficient. By doing so, the requesters, who have the most motivation to see the task completed, can direct their efforts towards creating a better workflow for performers.

1.1 System Overview

To better understand the struggles performers face when managing up, we conducted a preliminary study to inform the design of interfaces that allow requesters to take on those struggles. The study involved a one-week-long field study with 32 participants acting as performers and receiving requests using three different approaches to request management: instant messaging (free-form text), collaborative task-management tools, and our experimental approach based on the literature of request management [60]. The results showed that participants using the collaborative task-management tools felt the most burdened by coordination work, while those using our approach were better at following request specifications and attending to requests in a timely manner.

We developed TaskLight, a tool that enables requesters to invest effort in task coordination and guide performers through request fulfillment. The process of fulfilling requests involves continual information gathering

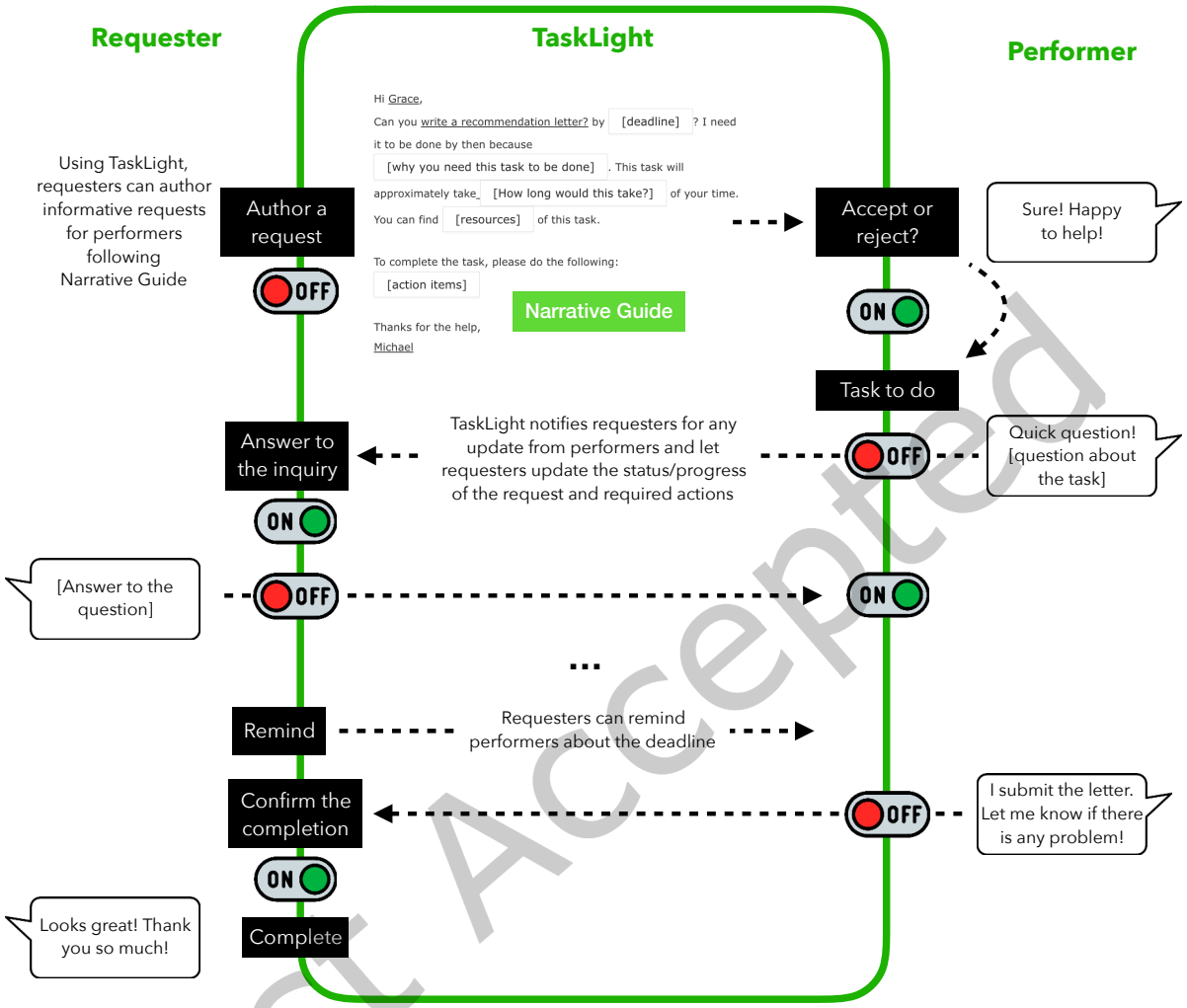


Fig. 1. Request management involves several meta-tasks, such as authoring initial & informative requests, information exchange, clarification, follow-up etc.; losing track of the status of a request can lead a delay or a failure to satisfy the request. Using TaskLight, requesters can invest effort toward such meta-tasks during the request life-cycle and minimize the burden of managing meta-tasks from performers' end.

and management. Through our formative studies, we identified a list of request-relevant information that can benefit performers. TaskLight guides requesters in collecting this information using the `Narrative Guide` feature. Additionally, given the uncertainty and numerous moving parts involved in the life-cycle of requests, it is common for performers to wait and return once requesters clarify specifications. With the `Request with Action Disclosure` feature, requesters can manage attention and highlight the different required actions by performers at each stage of the task.

To assess the effectiveness and usability of TaskLight, we conducted two field studies. The first study (N=14) aimed to understand how TaskLight is utilized in requester-performer interactions for various logistical and

common tasks, such as scheduling meetings and requesting comments. This study helped us evaluate the usability of TaskLight. In the second study (N=24), we compared the effectiveness of our tool with other pre-existing tools. Our results showed that TaskLight enabled requesters to invest effort towards a more streamlined process of managing up, and they could effectively make use of performers' attention. Consequently, this led to a better perceived success rate in task completion.

We conclude our work with a discussion about why managing up requires additional attention by researchers and systems, especially as a potential area to study the design of socially sensitive interfaces and intelligent task-management systems.

2 RELATED WORK

Our review of related work covers a range of experimental tools that researchers have developed over the years to facilitate collaborative management of shared tasks. We begin by examining two domains for organizing tasks: personal and group task management. Subsequently, we delve into request management and its distinct challenges, including the complexities of coordinating multiple individuals with varying levels of motivation.

2.1 Personal Information Management

The literature on Personal Information Management (PIM) suggests how personal information space can facilitate individuals' task management. While some of this work focused on rediscovering informal information [13, 40], much of it dealt with task management as a series of personal actions or concerns [9, 10]. Jones connected information management with the challenge of performing tasks based on the personal actions [39]. Bellotti, Ducheneaut, and their colleagues centered on email as a communication mechanism for personal task management [7, 8, 83], as did Gwidzka [31, 32] and Bergman et al. [11]. Many of these efforts were prompted by the then-current concept that knowledge workers spent the majority of their working time in email, and that email was a probable "habitat" for managing tasks [19, 49, 83]. Researchers hoped to take advantage of the implicit semantics in the semi-structured nature of email ([22, 57, 65]; for similar approaches, see Malone et al. [53]).

Initial hopes were that email could be used for task management, using selected items in the inbox as a to-do list [46, 65, 74, 81]. Taskmaster [8] was proposed to facilitate this use case, where email users can manage their tasks via email messages. Previous literature on email management helped us understand how individuals manage their task list, such as deferring the task to later by marking emails as unread [54, 70]. However, other researchers noted problems with this approach. While Karger et al. shared an interest in semi-structured information [22, 53, 57], they also noted that the semi-structured information in email was insufficient to support human tasks [43]. Karger continued this argument, warning that analysis focused solely on the implicit semantics of email lost a great deal of information ([41]; see also [12]), because information needed for tasks and other actions was distributed across multiple sources, applications, and repositories [42]. Whittaker et al. eventually concluded that the many email-based strategies were inefficient and insufficient [84].

In an influential paper, Whittaker and Sidner proposed a key solution to the problem of email overload: email had become burdened with too much functionality and too many specialized tasks ([85]; a finding that was partially anticipated by Mackay et al. [52]). The idea behind using email as a task infrastructure was to facilitate collaboration with others on larger tasks [20, 51, 62, 63, 71]. As organizations shift their communication channels from email to instant messaging, researchers have started exploring the integration of task management into various communication tools, such as chat [79] and calendars [17]. This highlights the need for a better way of organizing tasks beyond conversation threads and underscores the importance of communication within task management tools. TaskLight enables users to switch between request management and clarification, and guides requesters to craft informative requests so that both requesters and performers can prioritize and organize their tasks effectively.

2.2 Collaborative Task Management

In contrast with the experiments described above, several tools have been developed for groups to manage tasks in a centralized manner. In contrast to tools which individual members maintain their own records, these tools maintain relevant information in one place to ensure that all of the members are on the same page.

One category of collaborative task management systems is project management (PM) tools. PM tools allow project managers to assign and delegate tasks to their employees. Most PM tools also support chatting and commenting features on each task for follow-ups. In addition, project managers can use various visualizations, such as Kanban boards and Gantt charts, to track the progress of each task and evaluate their employees' performance [3, 55]. If the request is highly structured and formalized, project managers can even use automation to automatically process the work of coordination in their shared workspace [88]. For example, Github[1] has a clear representation of requests (e.g. pull requests) and of the task's progress (e.g. commits and merges). As a result, project managers can automate their PM space based on the actions in each request.

While PM tools are powerful, previous work suggests such tools only thrive in a limited number of cases [60, 73], which include industry organizations [55, 82] and online communities [66]. In industry organizations, the users of these tools are in authoritative roles, which allows them to maintain discipline among their employees. For instance, the tools are popular with project managers, who can assume their employees will perform as requested, whereas many requesters might not be in a position to take performers' help for granted [92]. On the other hand, in online communities, requesters often are not project managers but still have the upper hand; requesters are mostly volunteers and contributors who do most of the work in this universe and only request performers to review and merge their work. Given that both cases feature requesters with more power than performers, it is natural that performers do the work of coordinating requests, such as updating their progress and making sure they meet the deadline. While help-seeking shares the nature of collaborative tasks, (1) such requests are not routinized or formalized [75] and (2) requesters are less powerful and, simultaneously, *most motivated* to see the task completed. Primary users of PM tools are project managers who have the authority to decide which management tools to use, whereas requesters in help-seeking scenarios have less power to request and performers have less motivation to agree to work in the requesters' working sphere. In our work, we aimed to build a groupware system that both requesters and performers were motivated to use [30], because the tool is a workspace that each participant can benefit from (e.g., access to the useful information or specification of a request).

A separate line of tools focuses on task distribution handling in collaborative settings (e.g. corporations that are managing many customer inquiries in parallel [63]). Company emails are often organized using ticketing services (e.g, [25, 90]) where company representatives can organize their requests. Ticketing systems are project management tools specialized to work 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. Since the tool handles a massive number of requests, these tools are often equipped with automation and workflow.

While ticketing systems are for similar requester-performer dynamics to help-seeking, working spaces for requesters and performers are separated. However, previous work found that requesters' involvement can lead to successful and better request management. For example, requesters can provide useful information for performers [35, 45] and help their attention management [53, 65]. In our formative interview, our participants are willing to help their performers as much as possible to see their request fulfilled. In our study, we explored an interface that allows requesters to offer help throughout the life-cycle of request management, which led to successful request fulfillment by performers.

[1] https://github.com

2.3 Request Management

The act of requesting involves a requester seeking assistance from their peers to accomplish a task. Requesters often require help because they lack the necessary resources, expertise, time, or authority to complete the task independently. Muller et al. classified this type of collaborative task management as request management [60]. Request management presents typical challenges of collaborative work such as being time-consuming, stressful, and sometimes requiring extensive negotiation of the task's timeframe and details [75].

Request management also overlaps with personal task management; once performers receive a request (task) from the requester, they perform the task in their personal information space [23]. In other words, requesters hand off a task to performers and the task could be only done by the performer themselves. For example, when requesters ask their performer to write a recommendation letter, performers engage in the process of collecting, organizing, and presenting information related to a requester's qualifications, character, and achievements. The performer is typically tasked with gathering information about their collaboration experiences, skills, and accomplishments to provide a comprehensive overview of their abilities and potential. This may involve reviewing resumes, transcripts, work samples, and other relevant materials. The recommender then organizes this information into a letter format, highlighting the individual's strengths and making a case for their suitability for a particular opportunity. Unlike previous belief that such handing off between requesters and performers could be streamlined [88], it can be much complicated and unpredictable [76]. This is because personal tasks in request management settings are often subject to various constraints, such as task requirements and performer availability [60].

In request management, a unique challenge arises as tasks are handed off in a reverse direction compared to traditional work settings. Typically, bosses assign tasks to their subordinates who are expected to comply and complete the work. In contrast, in request management, subordinates ask their boss or peers to complete tasks for them. This requires subordinates to manage up and influence people above them whom they don't have power over [47]. While managing down focuses mainly on efficient communication and task completion, managing up involves considering different factors and demands social skills and awareness of *how to work around my boss/peer to get work done without irritating them* [36]. Therefore, literature on managing up suggests that subordinates understand their boss's working style and integrate new requests into their boss or peer's working sphere to reduce friction on their end [4, 14, 28]. Given the different requirements of managing up and managing down, existing collaborative task-management tool (which is for managing down) is unlikely fitting for managing up.

Muller et al. proposed a new approach to unify various tasks and requests [61]. Although their work was a problem analysis, it did not provide technological solutions. IntroAssist [35] is a relevant system from prior work that assists requesters in writing emails to solicit help from strangers. Our paper aims to develop a more comprehensive technology that supports the common aspects of the categories identified in [60, 61]. We focus on managing-up settings and utilizing requesters' motivation to complete tasks while supporting performers' personal task management.

,

3 FORMATIVE INTERVIEW

We sought to understand how knowledge workers manage their supervisors or peers and what factors influence their choice of request management tools. Furthermore, we characterize the current practices of request management and design requirements for our tool.

Study protocol. We conducted semi-structured interviews with seven individuals who regularly engage in managing up during their daily work and make their own decisions regarding tools and collaboration practices related to request management. Each interview lasted for one hour and followed a script that included questions related to the interviewees' collaboration practices and the tools they use. We also presented existing project

management tools, such as Gantt charts and Kanban boards, and asked the interviewees to compare them with their current practices.

Participants. We recruited participants through a posting on a mailing list of a private university and by word-of-mouth. Each interviewee was compensated with $15 for their time. The participants (five women, two men) held positions such as officers of a graduate council, members of a diversity initiative within a graduate program, and members of extracurricular clubs. The group also included individuals working on collaborative projects for courses or in industry.

3.1 Findings

All interviews were transcribed, and analyzed by axial coding of the seven interview transcripts following an open-coding protocol. To yield concepts and themes, the authors discussed the codes through multiple iterations. This involved a series of collaborative sessions where the authors reviewed and consolidated codes, ensuring a thorough exploration of the transcripts. For this reason, we did not calculate inter-rater reliability [56].

RQ1 (Understanding Dynamics) sought to discern the intricacies of managing up and identify ways to harness requesters' motivation to address these challenges. In the following sections, we delve into participants' experiences with managing up and the areas in which they seek assistance.

Managing the shared space is too costly (Burden of coordination work): The cost of maintaining the shared space is one of the key factors in choosing a request-management tool. Although the concept of a shared task management space is universally popular among our interviews, we found that that these do not last long, because they are not maintained on time, eventually disappearing due to the maintenance burden. At the start, they decided on a central workspace or project management space, such as Google Docs, to maintain a list of requests shared among their group. Although nearly half of the interviewees liked project management tools in principle, they doubted that keeping them up-to-date was feasible given the nature of their group. Echoing previous work, participants indicated that email and instant messaging are the most dominant channels for request management. They sent, received, and followed up on requests via chat interfaces, which did not require maintenance.

Four out of the seven interviewees expressed appreciation of Kanban boards, citing their effectiveness in visually distinguishing requests in different stages. One interviewee also suggested that the ability to customize columns in Kanban boards could be useful for expressing project-specific stages. However, interviewees noted that Kanban boards and Gantt charts require too much management to keep up-to-date. As one interviewee stated, *"I think the problem with the club team is I feel like I'm not a full-time [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 get outdated."* This highlights the fact that while coordinating the project is itself a task, this coordination task is not explicitly materialized or assigned, making it unclear who is in charge of such work, even if multiple people are involved in collaboration.

Request clarification and management occur on two different platforms (Need help to manage their attention across different types of coordination): Another reason behind the extinction of the shared task management space is the disconnect between task-related discussion and the request management space. Collaborative tasks inherently face numerous changes and need to be adjusted accordingly [61, 69, 75, 77], oftentimes requiring volatile coordination of non-routine intellectual work [15]. Hence, it is natural that people have a large volume of task-related conversations over email; 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"*. This leads to users juggling different discussion tools and request management tools, which can result in a loss of context and ideas [78]. Notably, performers 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 at the meeting. I guess it's somewhere buried in these emails." Some performers 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 marks [the messages] as read."* This suggests that there is a need to help performers manage their attention and keep up with important information regarding requests.

Even though collaborative task management interfaces often have commenting features, they found that comments made in the task management interface are not as visible as simply posting on a group chat. It is because request clarification is a tangible componentof the interface or integrated into the request life-cycle. As a result, conversation channels eventually subsume all the work of coordinating tasks such as clarifying, tracking, reminders, and following up with tasks. One participant said, *"Once they finished critical tasks [..] 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' "*. In actuality, this process of notifying people that a dependency has been completed is a process that collaborative task management tools are well-equipped to carry out. However, as clarification is not regarded as a task or a blocker in the tools, it makes the tools much less attractive to users given how frequently they need to engage in clarification.

Giving performers additional work feels inappropriate (Requesters are willing to invest additional effort): Interviewees also raised concerns about the nuanced sentiment of "asking for help" versus "managing their colleagues". The interviewees said requests made to their performers are more like asking for favors rather than assigning tasks; hence, tracking and updating individual stages is unnecessary as long as *things get done by deadlines*. All of the interviewees said that Gantt charts are not suitable for managing requests since they require too much commitment and formality from the performers. However, interviewees did like that the charts show the working progress of each request, since doing so improves social translucence [21] of performers' work, and they allow the requesters to see what tasks their collaborator is currently working on. Instead of putting performers in charge of updating their own progress, interviewees sought opportunities to take on the burden of such coordination work themselves. For example, one interviewee shared their strategy of collecting progress and useful information for their performers based on the logs from different applications and messages shared in group chats. The interviewee said *"I'm just aggregating information from like meetings and what we had, you know, said that we're going to [update them on] messenger, what people have updated and the automated emails I got from GitHub whenever somebody pushed a commit."*

Summary: Through the interviews, we found that users have a desire for a system that helps them with the process of managing up with their boss or peers. Some of the critiques of existing systems are unique to managing up, while others apply to managing more broadly. Interviewees indicated that managing up involves the **burden of coordination work**, reflecting the fast-changing and non-routinized nature of requests. Hence, to cope with the life-cycle of managing up, they **need help to manage their attention across different types of coordination**. While they found that managing requests can be challenging, when they are requesters, they are **willing to invest additional effort** to reduce the burden on their performers.

4 DESIGN GOALS AND CHOICES

Our design goals and choices are motivated by our desire to implement a system that requesters can manage up their boss or peers. Our goal is also motivated by shortcomings in existing PM tools. We propose the following three design goals.

Provide interfaces that reduce the burden of coordinating requests: Previous research shows how request management is inherently dynamic and involves numerous iterations. To account for this, we seek to aid performers at various stages in the request-management life cycle; (1) At the beginning, requesters need to

write well-defined and informative requests that communicate information about the requests. Previous work revealed that requesters were anxious and not confident about asking for help [60]. Guidance for writing well-structured requests can help requesters feel more confident [35] and lead to successful request management [50]. (2) Once the initial request is sent to performers, requesters and performers go through a collaborative task construction process, where the request is not simply a static artifact from a requester to a performer, but a dynamic, collaborative one that leads to the formation of a request suitable for both parties [61, 75]. Interface support can allow requesters help performers to clarify and collect required request-related information.

Support bringing attention back to requests across different types of coordination work: A challenge both requesters and performers face is managing the different types of coordination work that arise when handling a request. This challenge can be multiplied if the requesters and performers are handling numerous other requests at the same time. In some cases, however, performers' attention towards requests is not always needed. For example, if a performer asks the requester a clarification question, their attention towards the request is not required until the requester responds back. On the other hand, when their attention is needed, it often requires different types of action. For example, a performer might be blocked from continuing their work until a clarification is complete; however, existing task management tools do not consider clarification as coordination work of requests and separate clarification from the rest of task management. Thus, we propose the goals of (1) integrating different types of coordination work within the request management tool and (2) helping requesters and performers pay attention to requests that need either requester or performer's work when necessary.

Use the motivation of requesters: In our formative interview, interviewees thought that current collaborative task management tools impose the role of a coordinator to performers or do not specify whether who are in charge of the coordination work. Our interviewees said that when they are requesters, they are willing to take on more of the coordination work. Hence, we aim to build an interface so that requesters can take on aforementioned coordination work for performers.

From these design goals, we began by investigating interfaces for performers. Given the motivation and willingness of requesters, it is a sensible decision to understand performers' needs first and design interfaces that enable requesters to help performers meet those needs.

5 FORMATIVE FIELD STUDY

To investigate which format performers prefer to receive their requests in, we conducted a field study and asked our participants to act as performers. Our eventual goal is to design a system in which requesters can author requests in performers' preferred way.

5.1 Study Design

Task. We began by compiling a list of common requests found in the prior work of [60]. We then found appropriate ways of emulating these requests for our participants. For example, one common request for knowledge workers is to provide comments on the writings of colleagues. To emulate this request, we took inspiration from Kaur et al. by providing our participants with drafts of articles we found online [44]. The following is the list of requests distributed to our participants:

- Invite authors to a new Google Drive folder.
- Schedule a meeting.
- Write a review comparing your university's policies on maintaining undergraduate social life during COVID with those of other universities.
- Find three virtual social events that are happening this week at your university.
- Proofread some Wikipedia START-class articles.
- Brainstorm some COVID-friendly social events and make a flier for them.

- Provide feedback about my paper on my universities' COVID policies.
- Make a given flier more accessible according to accessibility guidelines.

Study protocol. We divided participants into three groups – the Text group (N=11), the Project Management (PM) group (N=11), and the Request Management (RM) group (N=10). All three groups received requests from a spreadsheet in Google Sheets. For the Text group, the spreadsheet was formatted as a single column in which each cell contained a different request described in free-form text. For the PM group, each row corresponded to a different request, and the cells in each row contained the following fields drawn from the metadata of the tasks in PM tools: `request description, priority, status, deadline`, and `requester`. For the RM group, each request was also given its own row, with the following fields based on our literature review and formative interviews: `title, steps, the purpose of the request, the resources of request, deadline`, and `anticipated time required`. Except for the priority and status information of the PM group and the estimated completion time of the RM group, all three groups received the same amount of information but in different formats (i.e. free-form vs. structured text). The prompt for each request of each group can be found in the Appendix.

Participants remotely attended a 30 minute-long tutorial session the day before the start of the study. Participants were instructed to access their request spreadsheet throughout the study period and reach out to us via email if they had any questions regarding the requests. In addition, the RM group was introduced to each attribute in their spreadsheet during 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 complete their requests on their own time. The study was conducted for five days and each request was made at a different time of the day, but at the same time to every participant in every group. Each request had different deadlines.

Upon the end of the study, we solicited participants to fill out an exit survey. In the exit survey, we asked them about the subjective workload they experienced when managing the requests. They were also shown a screenshot of the other groups' request spreadsheets and were asked to compare with their own.

Participants. We recruited participants by posting on a private university's community-wide mailing lists. Each participant was compensated $30 for their time. In total, 32 participants (19 women, 12 men, one non-binary, mean age=20) were initially recruited and participated in the introductory session. These were undergraduate students at the university and self-reported that they had recently worked in a team. By the end of the study, due to dropouts, 21 participants remained. We also consider participants to be dropouts if they did not access their request spreadsheet after we sent out the spreadsheet. Since dropouts happened before any condition-specific influence could occur, it is valid to ignore their existence and continue to treat the remaining population as having been independently and randomly assigned to the three conditions.

5.2 Measures

5.2.1 Task completion. We evaluated how effectively each group completed their tasks. To do this, we measured (1) the number of the tasks they completed, (2) how often they turned in their tasks on time, and (3) whether they met all of the requirements for each task.

5.2.2 Interaction with task lists. We also analyzed how the participants interacted with the task spreadsheets. First, we tracked how many times participants requested clarifications from the study facilitator in order to gauge how well participants can understand the requirements for each task. Second, we also counted the number of times each user visited their task spreadsheet. Because the number of times a user checks the task list is correlated with the level of stress and anxiety they have [48, 54], we were able to identify the stress level of each group by the number of times they visited their list.

Table 1. Results of the field study. The RM group followed task specifications significantly better than other groups. The PM group visited their list marginally more than the others (*: $p < .1$, **: $p < .05$).

	Text	**PM**	**RM**
# of task completed	4.9	5.8	4.4
Late	21%	21%	5%
Does not follow task specification	42% **	21% **	10%
# of requesting clarification	70%	70%	20%
# of times to visit the task list	11.8	61 *	33.8
How mentally demanding	3.8 (σ=1.9)	3.8 (σ=1.4)	3 (σ=0.7)
How successful	4.4 (σ=1.9)	4.6 (σ=2.2)	4.5 (σ=2.1)
How hard	3 (σ=1.4)	3.8 (σ=1.3)	3.25 (σ=0.8)
How stressed	3.8 (σ=1.6)	4.7 (σ=2.3)	3.25 (σ=1.8)

5.2.3 Perceived workload. Through an exit survey, we also sought the participants' subjective workload of the conditions. Participants were asked to fill out the NASA TLX form [33]. We did not include questions about physical demand and the pace of the task in our survey because the metrics are not relevant to our task and interest.

5.3 Result

We analyzed our participants' behavior using one-way ANOVA and their perceived workload using one-way Kruskal–Wallis ANOVA (Table 1).

PM users felt the most burden of coordination work: Table 1 shows the result of the study. PM participants had to make more effort to coordinate the work of request management than the other participants. Namely, they visited their spreadsheet five times (p=.02) more than the text group and two times (p=.06) more than the RM group. They needed to visit their spreadsheet frequently to update the status of their request. Furthermore, they found that clarifying and understanding requests was burdensome. Participants indicated that the RM model could help them to share this burden with the requester if the requester is authoring requests according to the RM model: *"This one (RM model) is nicer from the user (performers') perspective, [although] it does seem like it would be more work for the person creating the task list."*

RM users were most successful at bringing back attention to the request when needed: Despite frequent visits to their task list, PM tool users tended to not turn in requests on time as often as RM users did. This might imply that the PM tool is not particularly helpful at managing requests by the deadline. Some PM users compared their version to RM and said `anticipated completion time` might have been useful, while others preferred to designate it themselves. Unlike the PM tool group, users in the RM group were given the anticipated completion time and said they made good use of it: *"I also think that the anticipated completion time is useful because I can try [to] schedule my day around the necessary tasks."* However, participants also pointed out that the estimated time might be not accurate if it is given by requesters: *"The time estimate given is not necessarily reflective of my efficiency, so I prefer to see for myself and make my own estimate"*. This indicates that requesters should be given instructions to provide correct information for performers.

RM users followed the task specification more closely: Participants who used RM versions tended to follow task specifications better than others even with fewer clarifications, which indicates the data model of RM was more well-aligned to follow requesters' instructions than the other versions. There was a statistically significant difference in how well the groups followed the task specifications ($F(2, 18)=4.88$, p=.02). Specifically, the RM group users were significantly more attentive to specifications than the Text group (p=.01) and the PM group (p=.03). Participants said the `purpose of request` attribute helped them to tailor and understand the

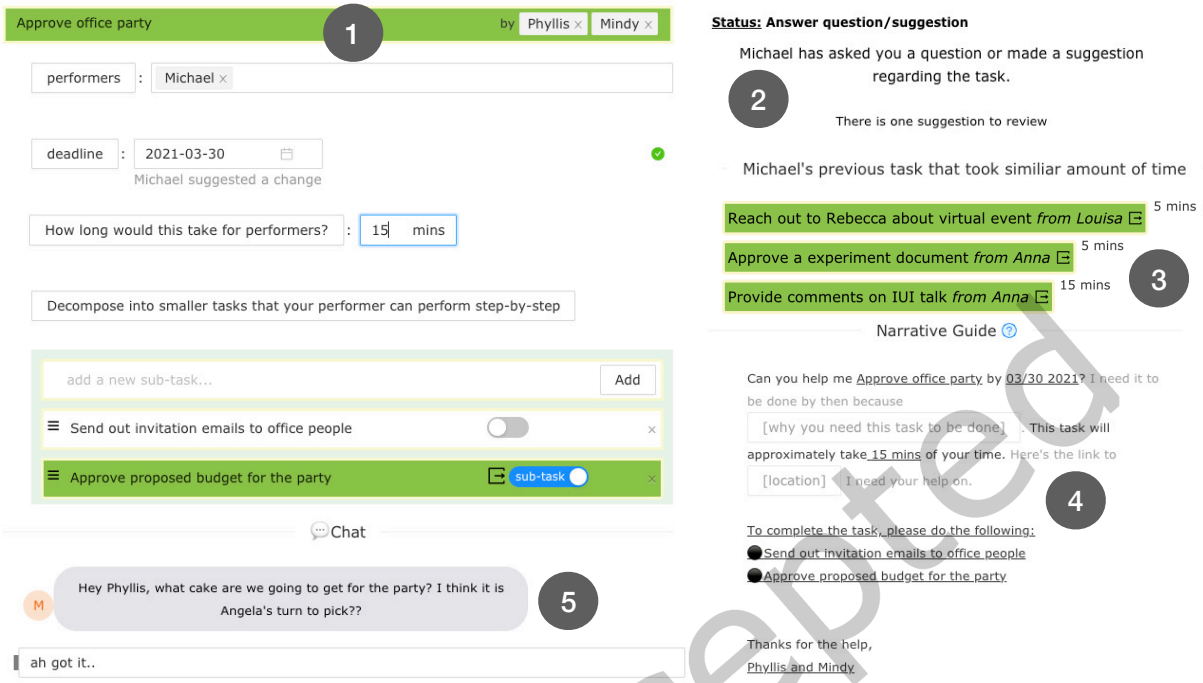


Fig. 2. The authoring interface of TaskLight for requesters. (1) Request-related fields. (2) Status and required action for the viewer. (3) Helper features for requesters. Based on the field that requester is filling out, the helper feature presents relevant information. For example, in this figure, 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.

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 easier to follow the specification if they were given the information: *"Sometimes the tasks were not as clear as they could have been which led me to email the team members to clarify"*.

Summary: The study revealed that project managers faced a disproportionate coordination burden, and users of the RM model were more adherent to task specifications, benefiting from features like anticipated completion times and clear task purposes. These insights directly influenced the development of TaskLight, which aims to streamline task management by providing clearer task context by requesters, balanced coordination efforts, and more accurate time-estimation tools to enhance efficiency and reduce managerial overhead.

6 TASKLIGHT: A REQUEST-MANAGEMENT SYSTEM THAT ALIGNS INCENTIVES OF REQUESTERS TO MANAGE UP

RQ2 (Interface Design) sought to identify an interface design that capitalizes on requesters' motivation to enhance managing up. Drawing from our prior exploration, we crafted the TaskLight interface tailored for two distinct user groups. For requesters, the design emphasizes their role in coordination and underscores ways to assist their performers (Fig. 2). Conversely, for performers, the focus is on maximizing efficiency (Fig. 6).

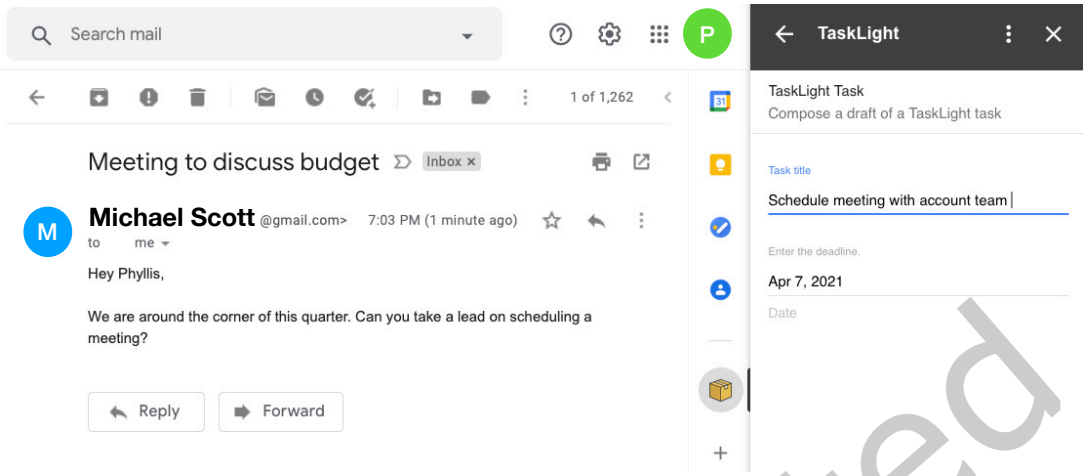


Fig. 3. TaskLight allows users to aggregate requests exchanged in other tools to TaskLight: an add-on sidebar is provided for the inbox and for shared documents, so that users can add the associated request to TaskLight.

6.1 Interfaces for Requesters

The requesters' interfaces offer various functionalities that aim to take on coordination and eventually help performers with work on the requests. It focuses on the two types of coordination work that cost the most effort for performers – (1) finding necessary information for task completion in place and (2) managing their attention, so they pay attention to requests that need performers' work and ignore ones that need requesters' input for the time being.

6.1.1 Guidance for authoring informative requests. TaskLight provides a shared workspace where requesters can contribute their effort when they are making each request. As a starting point, requesters can author informative requests that could be useful for performers to conduct the task. To provide guidance and feedback in authoring informative requests, TaskLight supports Narrative Guide. Narrative Guide constructs the text of a message based on the task metadata that the requester has already filled in (Fig. 2). Using Narrative Guide, requesters can visualize how each attribute of requests can be used and situated for performers. As the requester updates the value of the attribute, the Narrative Guide is updated accordingly. As the requester authors a request, using Narrative Guide, they may find attributes that have not been specified yet, but could be useful to specify. Accordingly, users can easily add more attributes by clicking on embedded buttons within the text. For example, if the requester would like to add `resources` for where the task should be completed, they can click on `[resources]` (Fig. 2.4), and the field will be added to the main editor (Fig. 2.1).

6.1.2 Consolidating out-of-band communication. Previous work revealed that collaborative tasking happens across multiple workspace tools and communication channels like email and chat. This can make task management more difficult for both requesters and performers.

To minimize work for performers, requesters can collect exchanges made across multiple channels into the unified context of the TaskLight client. We provide a sidebar add-on that allows several communication channels (email, shared documents, shared slides) to be included in the TaskLight interface. Fig. 3 shows an example of the TaskLight sidebar, which users can access by clicking on the *box* icon. Users can then write a request associated with the email message and the draft of their request will appear in TaskLight.

6.1.3 Helping to bring attention to requests with tracking and managing requests with required actions disclosure. One of the complexities of working on a request is that different actions may be required depending on what state either the requester or performer is at within the life cycle of the request. This is especially apparent if the group uses different tools for managing requests or clarification, but can also occur within a tool as to-dos get lost in a stream of chat messages [73, 91]. In our formative interview, our interviewees hinted that they were concerned about the visibility of comments in task-management interfaces and so instead they opted to post task clarifications in a group chat.

We seek to improve managing attention for both requesters and performers by unifying updates on task status and clarification. We provide a rich set of possible statuses to indicate the various actions required of either requesters or performers (Fig. 5). Hence, instead of needing to pay attenion to multiple channels, requesters and performers can focus their attention in the singular context of TaskLight. The following is a list of possible statuses:

- *Draft to be Sent*: Inspired by a draft feature of an online software collaboration platform [2], this status allows requesters to inform performers that they are working on a request. Performers will know the request is under way and schedule their requests around it.
- *Accept or Reject?*: This status implies to performers that there is a required action for them 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.
- *Active*: The performer should complete the request soon; the requester should be available to answer questions. During this stage, requesters can *bump* this request and ask for updates to their performers.
- *Waiting for clarification*: Performers have asked a question or offered a suggestion and are waiting for a response back.
- *Completed*: The performer has completed the request; the requester needs to review the performer's work.

TaskLight provides requesters status updates to manage attention to themselves and their performers. For each update from performers on a request, TaskLight automatically updates to the next corresponding status. For each message from performers on a request, requesters got notified and asked to update the status of the requests. When requesters update the status, TaskLight updates a list view (Fig. 4) where a user can check which requests require their action (highlighted) and which do not (grayed out) based on status of requests. Therefore, performers can quickly navigate to requests that need their attention.

6.2 Interfaces for Performers

Performers' interface is drastically simplified compared to requesters' making it quick and easy for performers to get tasks done by leveraging aids provided by their requesters.

6.2.1 Structures of requests. Performers are provided relevant information for each of their requests. We improved upon the list of information we introduced in our field study as participants said that they were useful to perform the requests. For example, while the participants acknowledged the 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, requesters offer performers with more accurate estimation based on previous requests they exchanged. We have a helper feature (Fig. 2.3) for each attribute that provides relevant information when requesters are authoring a request. For example,

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

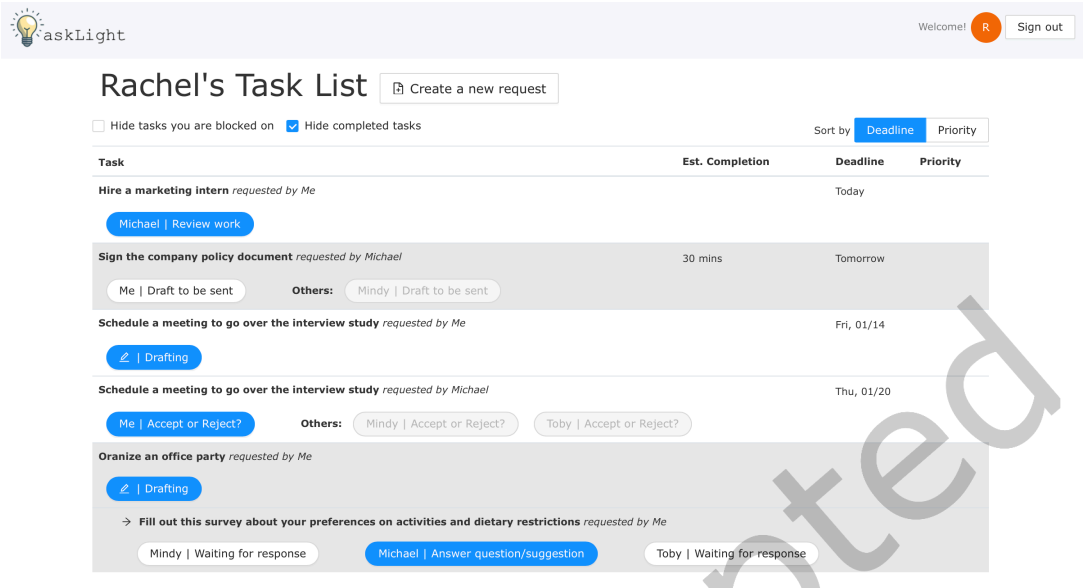


Fig. 4. The list view allows the user to determine which request to focus on at a glance.

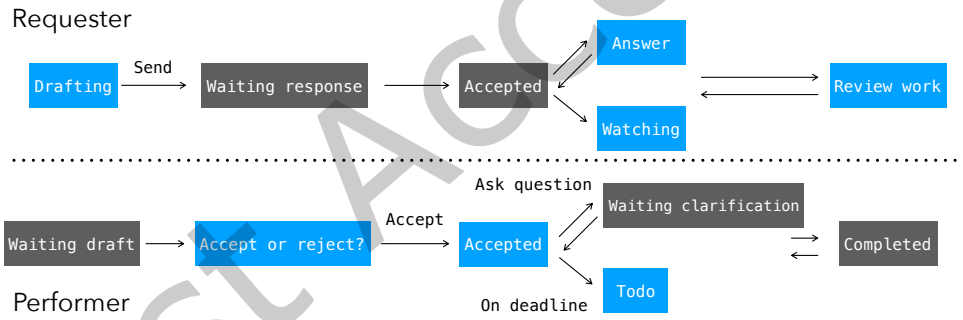


Fig. 5. Mapping of requesters and performers status and their evolution

- Deadline: Show how many requests the performer has and how long each request is expected to take on the selected date, so the requester can be mindful of selecting the deadline based on their performers' load.
- Anticipated completion time: Displays recent requests completed by the performers that both the requester and the performers were a part of for privacy reasons. Based on the similarity of the nature of the request, the requester can tune the right anticipated completion time for the current request.
- Action items: Shows a list of sub-tasks linked with related resources or team documents of the current request. We will present how this could be situated in request authoring in Section 6.4.1.

As for the `action items` field, Kokkalis et al. 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 actions items [45]. Our `action items` field is inspired by their work and we provide the guidance proposed in their work.

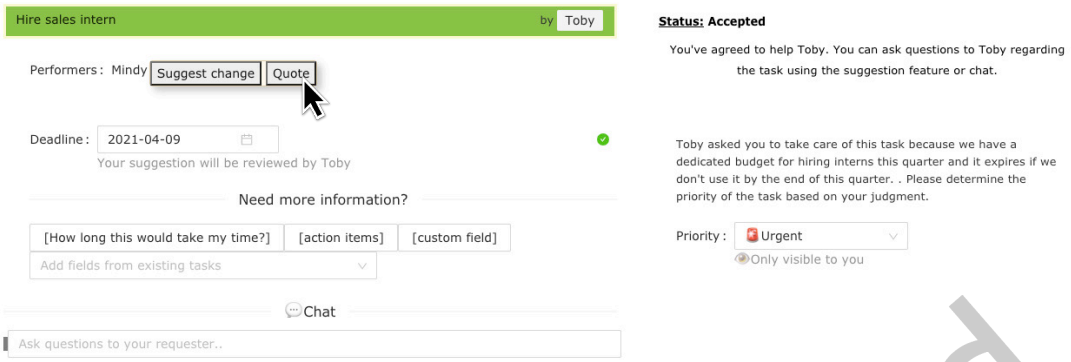


Fig. 6. The review interface of TaskLight for performers. Performers can make suggestions on requests and ask for more information or clarification if necessary.

6.2.2 Efficient request clarification. In order for performers to clarify information about a request in TaskLight, the interface supports two features – chat interface and suggestion features on request fields. Both features trigger a status change of the request (Fig. 5) to help users manage attention – using one of two features will change the status to *Waiting for clarification*, indicating that the requester should address their performer's question. In the meantime, the performer waits and attends to other tasks that they can work on at the moment.

In each request within TaskLight, there is a chat interface that could be used to clarify information about the request. As another method to clarify or negotiate some aspects of a request, TaskLight presents a *suggestion feature on request fields* (Fig. 6). Just like in collaborative writing, requesters and performers can make and review suggestions to make a request more concrete and satisfy the people involved in the request. In TaskLight, requesters are analogous to users with *edit* privilege and performers are users with *suggest* privilege. Together, performers can make suggestions to adjust a request, and requesters have control over changes. Using suggestion features, performers can perform common interactions around requests, such as delegate requests to others if they can not help their requester (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* and referring to 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 a 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.

6.3 Implementation

TaskLight is a React application with Ant Design[3] 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.

6.4 Examples

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

6.4.1 Collaborative writing: Moving requests to be on performers' radar. During collaborative writing, a leading author needs help from their colleagues to work on different parts of their write-up through discussion. In this

[3]https://ant.design/

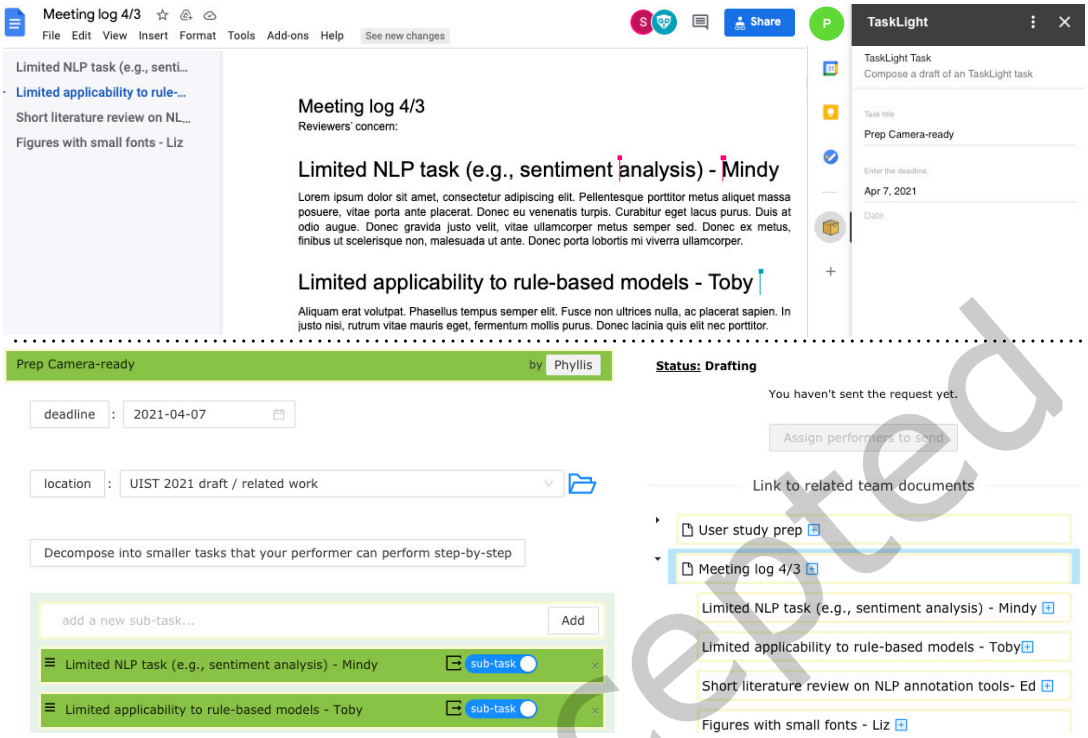


Fig. 7. (Top) Meeting log used during discussion of request distribution. A requester can create a draft of requests using an add-on interface. (Bottom) In TaskLight, the requester can finish authoring the request by importing the meeting log as sub-tasks.

example, a group of writers is working on comments made by paper reviewers on their submission and deciding who is in charge of which comments. Phyllis (requester), the leading author, oversees this process and ensures to resolve all the concerns from the reviewers. They are junior researchers working on a paper with senior researchers. During a meeting, they discussed who can offer help on which part and divided up tasks. Normally, it is each meeting organizer's job to remember requests made to them. Phyllis wanted to make sure if everyone gets the requests and help them remember the task. Fig. 7 illustrates the workflow. Phyllis decided to create requests on the meeting log. TaskLight imports the document structure of the meeting log. Phyllis creates requests based on the meeting log imported by TaskLight, adds specifications, and sends requests to their performers. This way, the requests will be under performers' radar and linked with the specification, since requests are added to their list in TaskLight.

6.4.2 Getting approval: Request construction and iteration. Take for example that Phyllis (requester) would like to hire an intern, but needs to get approval from her boss Mindy (performer). Phyllis decides to send the request to Mindy through TaskLight and adds all the related documents (e.g., intern call from the company, intern's resume) for Mindy for Mindy's convenience. TaskLight allows users to easily iterate over content by providing features for performers to offer suggestions (e.g., changing deadlines, adding custom fields). Such a feature is integral to request management as teams undergo frequent change. Mindy accepts the request and realizes that she must first check if there's enough funding for the intern. She wants to remind herself and communicates

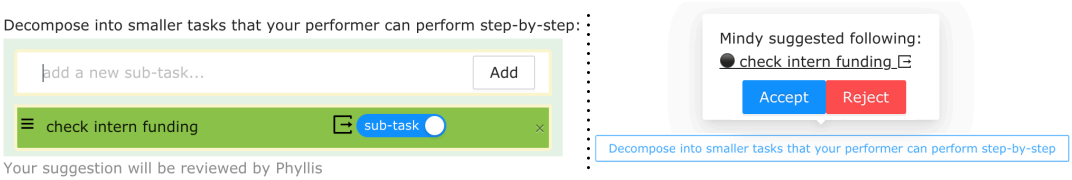


Fig. 8. (Left) A performer offers suggestions to reflect the changing needs of a team. (Right) A requester reviews the suggestion.

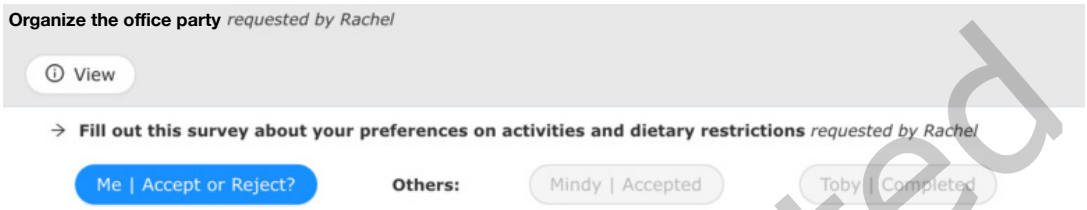


Fig. 9. A performer's view on requests with multiple-performers requests. TaskLight shows other performers' progress and de-highlights the request without required actions from the user at the time being so that the user can manage their attention.

to Phyllis that she has a blocker to approve hiring. Hence, she adds "check intern funding" as an action item for herself to the original request made by Phyllis (shown in Fig. 8). She is told that her new action item will be sent to Phyllis as a suggestion and be reviewed by her. This helps Mindy to break down the request as well as Phyllis to understand that Mindy has a blocker and is working on her request. The two can move forward as they construct the task together.

6.4.3 Soliciting inputs: Disclosing requests motivation and required actions by performers. One often finds themselves needing inputs from their boss and peers. They might feel overwhelmed and easily get lost in what they should do. Especially more so when the coordination is part of a big project. In this example, Rachel is organizing a launching party for the product of her team, which is one of the big projects at the office that needs substantial input and contribution from multiple colleagues. Rachel needs attendees to share their feedback on the party, so she decides to create a request for organizing the party in TaskLight. Then, she adds "fill out survey" as an action item of the main request (Fig. 9). 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 the in-completion of the sub-task. This way, the user can quickly pick up what actions are required by them and why.

7 DEPLOYMENT USER STUDY

We conducted two field studies to understand how the TaskLight interface fares when people utilize it for managing up their peers. In the second study, we improve usability issues of the interface based on feedback from the first study.

7.1 Study 1

For the first study, we explored how users engaged with TaskLight to express and communicate various tasks for managing up. In addition, we also looked for any interface components that may have been missing.

7.1.1 Study Design.

Tasks. We used the same set of study tasks as in our formative study.

Study protocol. Participants took part in the study in pairs, where they alternated roles of being requesters and performers throughout the study. 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, and completed a practice run-through of assigning and performing tasks with their partner. For ease of access, we then assigned TaskLight as their homepage, and installed 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 request from their study partner. When playing the role of the performer, participants were instructed that they would have to periodically check the TaskLight interface for newly assigned tasks. The study was conducted over 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).

At the end of the study, we asked each pair of participants to complete a 30 minute exit interview. In the exit interview, participants were asked a series of questions about their experience using TaskLight.

Participants. We recruited participants by 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 (10 women, 4 men, mean age=21), who are 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 remained. We considered participants to be dropouts if they failed to request three or more of the 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.

7.1.2 Study Results. All exit interviews were transcribed and analyzed using the same method as described at our formative interview study.

Participants felt that TaskLight made requests feel more personalized. Participants mentioned that compared to messaging interfaces (e.g., email, Slack), TaskLight offered similar communication but provided more helpful structure: *"the things that I do use are essentially equivalent to speaking to them in person [..] 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 [features] helps people. Like builds a request, like all of the fields that specify, you know when you want this done by and things like that. And that felt intuitive to use"*. 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* `resources` *just click on* `resources` *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 `resources`, 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 was a field that I could use is like requirements or sort of like an overall description of what to do". 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 sentiments 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"*.

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"*, categorized action, and surfaced it well. Participants especially provided positive feedback on the unique chat channel integrated with each request, stating that it provided *"a distinct chat per project, instead of like you know if you're communicating with somebody 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 integrated chat interface, participants also made good use of the `resources` field, citing how it was a convenient way of linking relevant documents: *"the* `resources` *thing was also really, really useful. That was probably my favorite feature because [..] when I'm [..] 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 use 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 participants said they would use TaskLight again, as they found it adequate for small-scale managing up tasks. The top two reasons why participants stated they would continue to use TaskLight are because it 1) feels centralized and 2) shares relevant task information of on demand: *"It was just like a very convenient and [..] generally relevant to tasks like I [..] haven't had a system provide before"*.

Participants mentioned that they found TaskLight suitable for managing up and would use TaskLight for smaller group projects with multiple tasks. Participants said the interface is not scalable for large projects as it would easily get overwhelming. However, for smaller settings and managing up their few bosses or peers, 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"*.

Participants called for notification to continue using TaskLight. We intentially did not have notification in TaskLight, as we assumed users would check back on the interface to see any updates in their requests. However, as a requester, they wanted to notify their performer about updates, and as a performer, they wanted to at least know they are supposed to check back on TaskLight. During the study, some participants ended up using messaging 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"*. Since discussions about the actual task were contained in the interface, we were still successful in consolidating discussion around the request. When notifications are supported by TaskLight, this limited dependency on chat platforms will likely disappear.

7.2 Study 2

While the previous study let us explore the usability of TaskLight for given tasks, it is unclear how TaskLight affects users' request management over the long-term. Unlike Study 1, in this study, participants were given only a high-level task, and it was up to them how they compose or manage their requests. We conducted a user study to see how users organically interact with each other through TaskLight. Reflecting on feedback from our previous study result, for the version of TaskLight used in Study 2, we implemented a notification feature which sends an email to users whenever the status of a task transitions to one of "action required" statuses (Section 6.1.3) or when there is a notable update on their tasks. We also modified the `resources` field to allow users to link multiple links or documents per request.

7.2.1 Study Design.

Task. We sought cooperation group work to be a task for this study. As a result, we chose collaborative writing, which requires coordination and creativity to finish the task. Participants were asked to conduct collaborative writing on a range of topics. The topics of the write-ups were the mental, physical, spiritual, psychological, emotional and social well-being of a university. To emulate the situation of managing up their peers, each team member acted as a lead writer of their write-up who needed to coordinate and oversee the rest of the group members to complete their write-up. Hence, each group was engaged with multiple write-ups concurrently each week. We asked participants to discuss which topic they want to be in charge of. This task was an instantiation of a "creativity" task in the McGrath's Task Circumplex Model [58] and as such required teams to work interdependently and engage in discussion and coordination to arrive at a solution.

Study protocol. To explore how TaskLight may be used in the wild, we conducted a two-week-long field study. Participants were asked to lead two writing tasks using two different interfaces out of three options – TaskLight, modified TaskLight to replicate project-management tools (PM), and participants' choice of existing tools e.g., Trello, Notion, Slack – with their collaborators. We introduced the PM condition to measure our system without participants' response bias (i.e. participants adjust their behavior to meet the study investigators' expectations) [18]. Basically, TaskLight and PM were the same style (Fig. 10), however, PM only has features that already exist in project-management tools (Table 2). TaskLight and PM were both introduced as TaskLight in different versions, focusing on different benefit and strength. In this way, comparing participants' perception towards TaskLight and PM allows us to evaluate the system without participants' bias towards tools of the study investigators. The order in which they used the interfaces was randomized. The procedures were as follows:

(1) Tutorial (1 hour): Each group was invited to a tutorial session a day before the start of the study, where they received a tutorial on how to use the assigned interface and tried it out on a sample task. We also introduced them to the tasks they would conduct over the next two weeks. Each group conducted the group task – one per week, each with a different interface – with additional tasks to ensure engagement.
(2) Collaborative writing (1 week, 2x): The group spent one week engaging in a collaborative write-up with one of the interfaces (repeated with the other interface). In addition, each group used a folder on Google Drive which contained a shared doc where they can work on write-ups. At the end of the week, participants were asked to submit the write-up that they were in charge of and fill out an exit survey.

Table 2. Feature Comparisons of PM and TaskLight

	PM (modified TaskLight)	TaskLight
Request field	Deadline, Priority	Plus Estimated completion required, Purpose, Resources, Action items
Chat	Disconnected with request status	Integrated with request status
Narrative Guide	Not included	Included
Request status	Not started, Pending, In progress and Completed	Status described in Section 6.1.3

(3) Exit interview & survey (1 hour): At the end of the study, participants engaged in a 1-hour exit interview focused on understanding their experience and eliciting feedback on our system. The interview was semi-structured and guided by a list of questions. Similar to Study 1, the first author conducted axial coding of the interview transcripts with an open-coding protocol. The authors then discussed and refined the codes.

Participants. We recruited participants using the same method as Study 1. However, we prohibited participants of Study 1 from participating in this study. Each participant was asked to sign up for the study with two peers that they had collaborated with before. Each group was compensated $285 for their time. We recruited eight groups, resulting in a total of 24 participants (17 women, 6 men, one did not specify gender, mean age=21). One group dropped out during the deployment.

7.2.2 Measures.

Group interaction. To compare how different systems shape team interaction, we measured different aspects of groups' conversation. We counted the number of requests and word counts of each request. We also analyzed the conversations using the linguistic dictionary Linguistic Inquiry and Word Count (LIWC) [67].

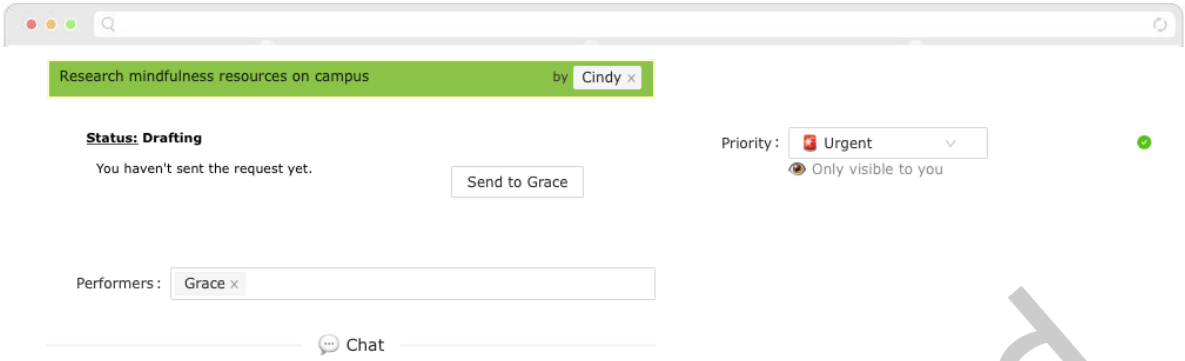
Perceived workload. We also measured subjective workload using a questionnaire of different features and the NASA-TLX survey [33]. We did not include a question about physical demand in our survey because the metric is not relevant to our task and interest.

System preference. During the exit interviews, participants were prompted to express their preference between the systems they had used, or to indicate no preference if they found them comparable. They evaluated the systems based on several criteria: overall system preference, coordination efficiency, quality of write-ups, ease of authoring requests, effectiveness in fulfilling requests, and proficiency in tracking requests.

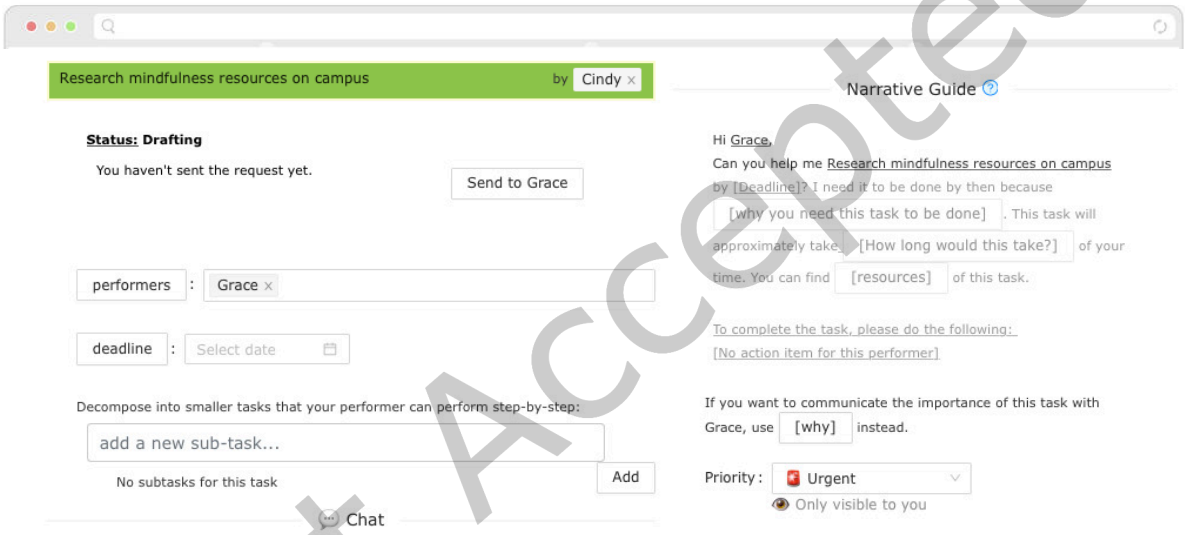
7.2.3 Study Results. To evaluate how different platforms helped request management, we asked participants to compare the two interfaces that they used (Table 4). We computed Cramer's V to get correlation between features. We analyzed responses of the exit survey using a linear mixed-effects model (Table 5).

TaskLight lets requesters author tasks in a clear layout with more helpful information for their performers. Most of the participants (14/15) indicated that they were able to author better requests using TaskLight (Table 4a & 4c). We found that the capability of authoring good requests is strongly correlated with the tool preference, how well they can coordinate, and how well they track requests (Table 4d).

Two factors emerged as contributors to better request authoring. First, TaskLight was able to nudge requesters to provide ample details for their request. Comparing PM and TaskLight, most participants (8/9) thought that they authored requests better and provided details for their *"performers' convenience"* (Table 4c). One said: *"Overall, it is [a] better setup where I can very easily be like: "I have a general idea of what I want you to do here's something that's going to allow me to specify it easily give you the tasks". "* Specifically, we found that when participants



(a) Interface of the PM condition



(b) Interface of TaskLight

Fig. 10. We included the PM condition which was also introduced as TaskLight to the participants.

used TaskLight, there was an increase in the use of the *cause* significantly more than with existing tools ($p < .05$) and marginally more than with PM ($p=.07$). Additionally, there also were more usages of words in the *work* category when using TaskLight than other conditions ($p < .05$). For example, a request made from one group in the study demonstrates the impact of the interface used; when they used their choice of existing tools, they lacked justification as to why they needed the task to be done, or the task tended to be more vague: *"do brief research about journaling's impact on mental health and emotional balance"*. The requests made from the same group when using TaskLight evolved as follows: *"write up your thoughts about food consumption and its benefits/consequences because this should be a reflection of your earlier two tasks and your thoughts while performing them"* and *"take 15-20 minutes to walk around outside or within your dorm because this can be a self-reflecting time or done to benefit physical health"*. Conversely, there was a significantly increased usage of *nonfluencies* (e.g., hmm) by the users of PM ($p < .05$) and existing tools ($p < .01$).

Table 3. Comparison of conversations of each group. While all the groups who used TaskLight have fewer words in the control version, for the groups who were assigned to use PM and existing tools, they both had increased word counts when they used existing tools.

	Group 1		Group 2		Group 3	
	TaskLight	Existing	TaskLight	PM	TaskLight	PM
# of requests	14	4	14	14	15	10
Avg. words per requests	9	13	10	6	13	17
Total word counts	129	54 (58% ↓)	137	93 (40% ↓)	199	177 (7% ↓)

(a) Groups exposed to the control condition first

	Group 4		Group 5	
	TaskLight	Existing	TaskLight	PM
# of requests	20	17	14	12
Avg. words per requests	12	13	15	16
Total word counts	245	229 (7% ↓)	219	200 (9% ↓)

(b) Groups exposed to the TaskLight condition first

	Group 6		Group 7	
	PM	Existing	PM	Existing
# of requests	15	31	31	17
Avg. words per requests	17	16	18	15
Total word counts	255	489 (92% ↑)	557	584 (5% ↑)

(c) Groups not used TaskLight. Group 6 used PM first and Group 7 used later.

Second, participants found that the threading structure of requests was more effective compared to putting a request in a message format. Requests in TaskLight and PM made requesters structure them in a thread, which helped them to write more concrete requests; most participants (4/6, Table 4b), who used existing tools and PM found that PM was more helpful to author requests.

The additional specification in requests also led to higher satisfaction for performers (6/9, Table 4c) because there is more information and the layout of information is easier to digest: *"The layout where from seeing information is very helpful."*

Participants felt less stressed when using TaskLight, and felt the system required less effort and rush. TaskLight led to a lower stress level than both existing tools and PM (Table 5). Echoing the self-evaluation of their own write-ups (Table 4a-4c), there was no significant difference in the self-perceived degree of success between the tools. Furthermore, TaskLight users found it easier to achieve a similar outcome than when they used PM tools, and TaskLight users also felt less hurried than when they used existing tools.

TaskLight had a different effect on participants' mental load depending on the order of the tool they used. For participants who used TaskLight first, they were trying to mirror their behavior of using TaskLight even when TaskLight was removed in the second week; they were authoring requests in a similar manner and length (Table 3b). Participants who used TaskLight first (6/6) thought using TaskLight was equally or less mentally demanding than the other tool. They appreciated the fact that TaskLight *"prompted to do on what to put in"*, meaning that TaskLight specified exactly what information to provide the interface about the task. Furthermore, it helped the way they authored tasks during their following week with another interface. However, a few users who used TaskLight later (4/9) thought that it was more mentally demanding (Table 3). These groups felt like

Table 4. Results of qualitative comparison. We asked participants to pick either of the tools they used for each question. They were also allowed to pick neither (i.e. Equal) if they found no difference between the tools.

	Preferred system	Coordination	Quality of write-ups	Request authoring	Fulfilling requests	Tracking requests
Existing tool	16.7%	0%	0%	0%	0%	0%
TaskLight	66.7%	100%	50%	100%	50%	100%
Equal	16.7%	0%	50%	0%	50%	0%

(a) Comparison between existing tools and TaskLight (N=6)

	Preferred system	Coordination	Quality of write-ups	Request authoring	Fulfilling requests	Tracking requests
Existing tool	33.3%	50%	0%	16.7%	16.7%	0%
PM	33.3%	0%	0%	66.7%	16.7%	100%
Equal	33.3%	50%	100%	16.7%	66.7%	0%

(b) Comparison between existing tools and PM (N=6)

	Preferred system	Coordination	Quality of write-ups	Request authoring	Fulfilling requests	Tracking requests
PM	22.2%	0%	0%	0%	0%	0%
TaskLight	77.8%	66.7%	0%	88.8%	66.7%	55.6%
Equal	0%	33.3%	100%	11.1%	33.3%	44.4%

(c) Comparison between PM and TaskLight (N=9)

	Coordination	Outcome	Authoring	Performing	Tracking
Preference	.35	-	**.58**	.13	**.68**
Coordination	1	-	**.74**	**.56**	**.61**
Outcome		1	-	-	-
Authoring			1	.42	**.78**
Performing				1	.29
Tracking					1

(d) Correlation between features of interfaces. Each value indicates Cramer's V between two features. Participants' perceived coordination is strongly correlated with the entire life cycle of requests, whereas the quality of outcome is not correlated to any as most of users found that the outcome were at the equal quality regardless of tools they used.

they were forced to transform their way of task authoring to TaskLight's way and input all the fields and come up with reasonable values, even if some fields are obvious. For example, for the task asking for proofreading, participants had to come up with a reason for the `why` field. This suggests that the Narrative Guide lacks flexibility.

Users are able to bring timely attention to requests using TaskLight. Most of the TaskLight users (11/15) said that they were better able to timely manage their attention with TaskLight (Table 4a & 4c). There are multiple contributors to this; first, users said that when they were using other messaging apps or PM, they were overwhelmed by the notifications and lost updates: *"Every message is treated equal, so I would get the same notification for like someone texting the whole group chats as [my requesters] assigning me my tasks."* On the other hand, using TaskLight, they were able to indicate important updates and point their performers to their tasks. One participant said: *"TaskLight was very good and [..] focused on the fact that I was doing a task and then made it very easy to give other people those tasks and [..] they would know [..] this is the task."* Next, better authoring of

Table 5. Results of self-rating by study participants. They are analyzed using a mixed effects model with random effects for groups fit by maximum likelihood. Each value indicates coefficients. As a result, participants found conducting tasks using PM were harder than using TaskLight. They were significantly less stressed when they used TaskLight. Finally, when they used their choice of existing tools, they felt more rushed than when they used TaskLight ($p < .05$ **, $p < .1$ *)

	Conditions	PM	TaskLight
How mentally demanding	Existing tool	0.18 (p=.54)	-0.30 (p=.28)
	PM		-0.19 (p=.43)
	TaskLight		
How successful	Existing tool	0.05 (p=.82)	0.01 (p=.95)
	PM		-0.03 (p=.88)
	TaskLight		
How hard	Existing tool	-0.13 (p=.59)	-0.19 (p=.37)
	PM		-0.34 (p <.1) *
	TaskLight		
How stressed	Existing tool	0.27 (p=.40)	-0.80 (p <.01) **
	PM		-0.69 (p=.01) **
	TaskLight		
How hurried	Existing tool	0.12 (p=.65)	-0.42 (p=.08) *
	PM		-0.23 (p=.25)
	TaskLight		

requests simplified the process of request management and reduced back-and-forth communication between requesters and performers (Table 4d). Three participants mentioned that TaskLight and PM are equally efficient at tracking requests, since they didn't have to clarify any specifications when using TaskLight (Table 4c). Hence, they were only exposed to the same set of status as PM even when they were using TaskLight and felt no difference between TaskLight and PM.

8 DISCUSSION

We presented a tool for managing up one's boss and peers, TaskLight. Our design was grounded in previous literature regarding task management and request management as well as our formative study; however, a longitudinal study remains as future work. Here, we also share the potential of curated information from requesters suggests sociotechnical implications. We also discuss how request management needs a unique design space in between personal information management (PIM) and PM and why neither work perfectly fit for managing requests.

8.1 Design Implications

We set out to reimagine collaborative task-management tools for workers to manage up their boss or peers. Our contribution is that we design an interface that attentive to the social requirements of requesters. The key element of TaskLight is aligning values of requesters (getting things done) and performers (saving their time and effort). We observed that requesters were willing and able to take on the work of coordinating requests when they used TaskLight. They were incentivized to do so since they were motivated to get the task done, and they recognized that providing thorough request-relevant information helps the performer complete the request. Furthermore, using TaskLight, requesters not only provide *more* information but also do so *in a structure* that

performers can use for timely placing their attention. Together, TaskLight aligns incentives of requesters and channels their desire to get the task done in the right place.

This research suggests sociotechnical implications of *requester-sourcing*, meaning leveraging requesters' motivation as a key element in collaborative technology. First, it helps to address **technical** and information challenge of collaborative task management. The challenge of PIM stems from the messy and overloaded nature of personal information [7, 29, 65]. Because tasks are generally not specified with sufficient information, managing personal information has been not intelligent and is mostly done manually. It is the individual's responsibility to tidy and keep up with their messages and commitments, which constitutes additional workload on top of doing their actual job. This becomes especially burdensome for performers, who are doing tasks for their requesters and need to spend extra time on "tasks of tasks".

Requesters can be helpful in such cases if we allow them to take on the coordination work of request management, which helps performers do the actual work and, in turn, helps requesters reach their goal. Despite the eagerness and motivation of requesters, without guidance, requesters cannot grasp what information will be useful for performers [35]. Through our formative field study, we investigated how to form requests so that they gather and structure information for effective access for performers. Our final study indicates that managing requests through TaskLight was more effective and easy for performers to fulfill requests.

With the increased involvement of requesters, TaskLight can also deliver richer request-relevant information. For example, in our field study, participants remarked how useful several pieces of requester-specified information were for helping them schedule their day around request management. This was previously done through crowdsourcing [45]. In contrast with this, requester-sourcing supplements poor structure of information in PIM and delegates coordination work of request management to requesters. Requester-sourcing is cost-free, since requesters are incentivized to do so, and resourceful, since they have better context and understanding of requests.

Second, requester-sourcing might be an effective method to satisfy **social** requirements [1]. Current task-management tools fail to support the requirements of managing up due to inappropriate connotation and additional obligations imposed on performers. On the other hand, enabling requesters to take on the obligation evidently direct their effort without needing to assert work on performers; during Study 1, both requesters and performers felt that TaskLight made requests more personalized. And in Study 2, they perceived that requests provide more helpful information and felt less cognitive load in general when using TaskLight. Our results suggest that this may be due to reduced social friction between the two groups of users. We believe that such social requirements are ubiquitous in other group work, and requester-sourcing can be applied to other collaborative technology and enrich coordination processes between workers who seek to exchange their resources. For example, in multi-disciplinary teams, members in such teams often need to exchange each other's domain knowledge to be on the common ground. There is always a member who is more motivated to get the knowledge of other members than the rest of the teams (e.g., [64, 68]). Through requester-sourcing, the member can navigate the social requirement and potentially access others' resources more efficiently.

8.2 Positioning of TaskLight

As we described earlier, there was a multi-year effort to apply email technologies to task management [8, 19, 31, 57]. Karger's work suggested that these approaches would have limited power [41, 42], and Whittaker and colleagues provided a series of analyses of why email was insufficient for the task [83, 85]. And yet, email has many attractive attributes for light-weight task management. It imposes few demands during the creation of a task-oriented request [60], and therefore feels light-weight. However, email also imposes subtle burdens *later* if the sender or recipient needs to re-find the interactions [13]. The apparent light-weight-ness of email becomes an illusion in the longer term.

At the opposite extreme is project management systems. These systems typically require entry of a lot of *required* task metadata, making them unwieldy for simple task requests [16]. Project management systems also tend to make each user action surveillable [86], which can further discourage or even inhibit usage [60].

Table 6 outlines the dilemma. Personal (i.e., solo) task management is somewhat well-supported, at least through simple task-lists and paper-lists – but at the cost of a tendency for information to be irretrievable. Project management is reasonably well-supported for large teams and complex task configurations, but at the cost of unwieldy requirements for a lot of metadata. Even if requests *made by colleagues* inevitably have collaborative natures, there is a noticeable gap existing between research discipline and tools [23]. What about the space between those relative extremes? What about request management should position between two regimes? Both somewhat support request management but not perfectly fit – personal task management inherently misses the collaborative aspects and project management does not match with nature of requests.

Winograd and Flores attempted to create a solution to this in-between space through the Coordinator [24, 87, 88]. In summary terms, their proposal was to add more rigor to an email-like system. As we reviewed, their work ultimately led to a user experience that imposed both higher metadata documentation requirements and a transactional style of interpersonal interaction that removed the social bonds that invisibly support collaboration [75].

Geyer and colleagues proposed a different approach with ActivityExplorer, which used minimal metadata to connect people, digital objects, and timelines [26, 27, 62]. The initial prototype generated its own problematic user experiences [72], and eventually became part of a successful but heavier-weight product called IBM Connections Activities, which then led to a secondary set of integration issues with other workplace tools [5, 59, 89], and a further conceptual distance between the activity management software and the users' preferred tools for less formal communication and coordination. Several lighter-weight tools subsequently emerged, such as Trello [38] and Jira [2], and each of these tools also required its own separate application, environment, and user experience.

In TaskLight, we attempted a different strategy to meet the needs of the in-between work domain and support managing up one's boss and peers. We provided minimal and, importantly, *negotiable* metadata, but we kept the core of the communication and coordination activities inside a task management interface. We conducted a first evaluation of TaskLight, with reasonably positive reactions from users. Our next steps should involve a more comparative set of questions and a more formal experimental design to compare TaskLight with existing tools and products in the in-between domain of shared task/request management, using a range of evaluation concepts and criteria to examine each system's impact on task efficiency, performer confidence, and social relations. Further, we hope to develop Table 6 into a stronger set of comparative or even evaluative dimensions for analyzing TaskLight and other applications that may come "in-between."

8.3 Limitation & Future work

Our experiments have several limitations. First, participants of our interviews and field studies are from academia. We attempted to draw our insight from participants of different natures, such as officers from student government, members of diversity initiatives, and students doing course projects. It still remains as future work to see if our findings expand to other populations. We next aim to deploy our system in different backgrounds and examine how our system applies for various teams.

We only tested our system in a limited range of tasks, namely a collaborative-writing task. Research on CSCW tools typically relies on structured tasks because they permit the evaluation of specific group processes that prior work has established to be critical for group effectiveness. We attempted to mitigate the limitation by varying tasks in a few parameters. For example, participants were required to come up with various sub-tasks of writing in order to complete the write-ups. While we designed the task to be multifaceted, additional sets of tasks and settings could enrich our observation. Next tasks in series could be placed in inter-disciplinary groups and

Table 6. Comparison of Personal Task Management, Shared Task/Request Management, and Project Management

Concept	Personal Task Mgmt	Shared Task/Request Mgmt	Project Mgmt
Formality	Low	Moderate	High
Integration w/ colleagues	1 or 2	Few	Many
Collaboration	Low	High	Moderate
Initial Effort	Low	Moderate	Laborious
Subsequent Effort (i.e., refinding)	Laborious	Moderate	Low
Metadata	None or Implicit	Moderate & Negotiable	System Required
Legibility	Often none	Negotiated	Required
Surveillance	None	Potential	Intentional
Support	Email & IM/Chat	TaskLight	PM Systems

cross-teams where different members have different expertise and resources. Hence, there is a real requester and performer relationship present, unlike what we set up in our study.

We also believe that there is substantial room for improvement for the Narrative Guide. While we collected some empirical evidence that it guides and motivates requesters to provide more contextual information regarding requests, TaskLight users said for some requests, field values were awkwardly fitting in the Narrative guide or it is unnecessary to provide some fields but the interface still reinforces users to do so. As future work, the Narrative Guide can be intelligent and flexible, hence based on given requests, the passage in the guide can change. Or even further, the Narrative Guide could take the form of something other than text, which can be potentially more powerful and motivating than the current form.

9 CONCLUSION

We presented our tool TaskLight as a solution for managing up others who one does not have power over. 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 such as psychological burdens of managing up and inadequately situated tools. We seek a means of re-orienting responsibility and agency in collaborative task management in order for requesters to take on roles and coordination work of maintaining requests. Hence, while requesters and performers can benefit from the structure of request management, requesters can lead the request and have more awareness in the process, which requesters ultimately want in order to make sure the request is completed on time. Accordingly, we design TaskLight as a stepping stone to an interface that can achieve this. Through our field deployment study, we validate that TaskLight improves the process of managing up by guiding requesters to collect helpful information for performers and help them manage attention. Our work identifies the social burden of managing up and proposes design implications for managing-up systems.

ACKNOWLEDGMENTS

We thank Chinmay Kulkarni, Divya Shanmugam, Theia Henderson, Pranav Khadpe, and Amy Zhang for their feedback on our early draft.

REFERENCES

[1] Mark S Ackerman. 2000. The intellectual challenge of CSCW: The gap between social requirements and technical feasibility. *Human–Computer Interaction* 15, 2-3 (2000), 179–203.

[2] Noura Alomar, Nouf Almobarak, Sarah Alkoblan, Sarah Alhozaimy, and Shahad Alharbi. 2016. Usability engineering of agile software project management tools. In *International Conference of Design, User Experience, and Usability*. Springer, 197–208.
[3] Maruthi Rohit Ayyagari and Issa Atoum. 2019. Understanding Customer Voice of Project Portfolio Management Software. *International Journal of Advanced Computer Science and Applications* 10, 5 (2019). https://doi.org/10.14569/IJACSA.2019.0100508
[4] Rosanne Badowski and Roger Gittines. 2004. *Managing up: How to forge an effective relationship with those above you.* Currency.
[5] Aruna D Balakrishnan, Tara Matthews, and Thomas P Moran. 2010. Fitting an activity-centric system into an ecology of workplace tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 787–790.
[6] Peter Bamberger. 2009. Employee help-seeking: Antecedents, consequences and new insights for future research. In *Research in personnel and human resources management*. Emerald Group Publishing Limited.
[7] Victoria Bellotti, Brinda Dalal, Nathaniel Good, Peter Flynn, Daniel G Bobrow, and Nicolas Ducheneaut. 2004. 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*. 735–742.
[8] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. 2003. 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*. 345–352.
[9] Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. 2006. The project fragmentation problem in personal information management. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 271–274.
[10] Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. 2008. Improved search engines and navigation preference in personal information management. *ACM Transactions on Information Systems (TOIS)* 26, 4 (2008), 1–24.
[11] Ofer Bergman, Richard Boardman, Jacek Gwizdka, and William Jones. 2004. Personal information management. In *Chi'04 extended abstracts on human factors in computing systems*. 1598–1599.
[12] Michael S Bernstein, Max Van Kleek, MC Schraefel, and David R Karger. 2007. Management of personal information scraps. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*. 2285–2290.
[13] Richard Boardman and M Angela Sasse. 2004. " 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*. 583–590.
[14] Mark C Bolino, K Michele Kacmar, William H Turnley, and J Bruce Gilstrap. 2008. A multi-level review of impression management motives and behaviors. *Journal of Management* 34, 6 (2008), 1080–1109.
[15] Marcelo Cataldo, Patrick A. Wagstrom, James D. Herbsleb, and Kathleen M. Carley. 2006. 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*. ACM Press, 353. https://doi.org/10.1145/1180875.1180929
[16] Elizabeth F Churchill. 2017. Planning time: HCI's project-management challenges. *Interactions* 24, 5 (2017), 20–21.
[17] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. 2017. 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*. 2382–2393.
[18] Nicola Dell, Vidya Vaidyanathan, Indrani Medhi, Edward Cutrell, and William Thies. 2012. " Yours is better!" participant response bias in HCI. In *Proceedings of the sigchi conference on human factors in computing systems*. 1321–1330.
[19] Nicolas Ducheneaut and Victoria Bellotti. 2001. E-mail as habitat: an exploration of embedded personal information management. *interactions* 8, 5 (2001), 30–38.
[20] Thomas Erickson. 2006. From PIM to GIM: personal information management in group contexts. *Commun. ACM* 49, 1 (2006), 74–75.
[21] Thomas Erickson and Wendy A. Kellogg. 2000. Social Translucence: An Approach to Designing Systems That Support Social Processes. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (mar 2000), 59–83. https://doi.org/10.1145/344949.345004
[22] Oren Etzioni, Alon Y Halevy, Henry M Levy, and Luke K McDowell. 2003. Semantic Email: Adding Lightweight Data Manipulation Capabilities to the Email Habitat.. In *WebDB*. 13–18.
[23] Danyel Fisher. 2004. *Social and temporal structures in everyday collaboration (dissertation).* Ph.D. Dissertation.
[24] Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd. 1988. Computer systems and the design of organizational interaction. *ACM Transactions on Information Systems* 6, 2 (1988), 153–172. https://doi.org/10.1145/45941.45943
[25] freshdesk 2010. Freshdesk. https://freshdesk.com/.
[26] Werner Geyer, Michael J Muller, Martin Thomas Moore, Eric Wilcox, L-T Cheng, Beth Brownholtz, Charles Hill, and David R Millen. 2006. Activity Explorer: Activity-centric collaboration from research to product. *IBM Systems Journal* 45, 4 (2006), 713–738.
[27] Werner Geyer, Jürgen Vogel, Li-Te Cheng, and Michael Muller. 2003. Supporting activity-centric collaboration through peer-to-peer shared objects. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. 115–124.
[28] Victor M González and Gloria Mark. 2004. " Constant, constant, multi-tasking craziness" managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 113–120.
[29] Catherine Grevet, David Choi, Debra Kumar, and Eric Gilbert. 2014. Overload is overloaded: email in the age of Gmail. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. 793–802. https://doi.org/10.1145/2556288.2557013

[30] Jonathan Grudin. 1988. Why CSCW applications fail: problems in the design and evaluationof organizational interfaces. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. 85–93.
[31] Jacek Gwizdka. 2002. Reinventing the inbox: supporting the management of pending tasks in email. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. 550–551.
[32] Jacek Gwizdka. 2002. TaskView: design and evaluation of a task-based email interface. http://eprints.rclis.org/13700/1/GwizdkaJ-2002-CASCON-Conf.pdf Accessed 12 Jan 2022.
[33] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
[34] James Hollan, Edwin Hutchins, and David Kirsh. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 2 (2000), 174–196.
[35] Julie S Hui, Darren Gergle, and Elizabeth M Gerber. 2018. Introassist: A tool to support writing introductory help requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[36] Caryn Ines. 2017. Managing Up and Across. (2017).
[37] Anthony M Johnson. 2019. "I Can Turn It on When I Need To": Pre-college Integration, Culture, and Peer Academic Engagement among Black and Latino/a Engineering Students. *Sociology of Education* 92, 1 (2019), 1–20.
[38] Heather A Johnson. 2017. Trello. *Journal of the Medical Library Association: JMLA* 105, 2 (2017), 209.
[39] William Jones. 2004. Finders, keepers? The present and future perfect in support of personal information management. *First monday* (2004).
[40] William Jones, Abe Wenning, and Harry Bruce. 2014. How do people re-find files, emails and web pages? *iConference 2014 Proceedings* (2014).
[41] David R Karger and William Jones. 2006. Data unification in personal information management. *Commun. ACM* 49, 1 (2006), 77–82.
[42] David R Karger and Dennis Quan. 2004. Collections: flexible, essential tools for information management. In *CHI'04 extended abstracts on Human factors in computing systems*. 1159–1162.
[43] David R Karger and Dennis Quan. 2004. Haystack: a user interface for creating, browsing, and organizing arbitrary semistructured information. In *CHI'04 extended abstracts on Human factors in computing systems*. 777–778.
[44] Harmanpreet Kaur, Alex C Williams, Anne Loomis Thompson, Walter S Lasecki, Shamsi T Iqbal, and Jaime Teevan. 2018. Creating better action plans for writing tasks via vocabulary-based planning. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–22.
[45] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moontae Lee, Florian Schulze, and Scott R Klemmer. 2013. Taskgenies: Automatically providing action plans helps people complete tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)* 20, 5 (2013), 1–25.
[46] Nicolas Kokkalis, Thomas Köhn, Carl Pfeiffer, Dima Chornyi, Michael S Bernstein, and Scott R Klemmer. 2013. EmailValet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1291–1300.
[47] Sanjiv Kumar, Vivek S Adhish, and Abhimanyu Chauhan. 2015. Managing bosses and peers. *Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine* 40, 1 (2015), 14.
[48] Kostadin Kushlev and Elizabeth W. Dunn. 2015. Checking email less frequently reduces stress. *Comput. Hum. Behav.* 43 (2015), 220–228.
[49] LT Leung, TJ Humphreys, and AJ Weakley. 2008. Email as co-habitat in distributed organisations. In *Australian Computer Human Interaction Conference*. ACM Digital Library.
[50] Kurt Luther, Kevin Ziegler, Kelly E Caine, and Amy Bruckman. 2009. Predicting successful completion of online collaborative animation projects. In *Proceedings of the seventh ACM conference on Creativity and cognition*. 391–392.
[51] Wayne G Lutters, Mark S Ackerman, and Xiaomu Zhou. 2007. Group information management. *Personal information management* (2007), 236–248.
[52] Wendy E Mackay. 1988. More than just a communication system: diversity in the use of electronic mail. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. 344–353.
[53] Thomas W Malone, Kenneth R Grant, Kum-Yew Lai, Ramana Rao, and David Rosenblitt. 1987. Semistructured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Information Systems (TOIS)* 5, 2 (1987), 115–131.
[54] Gloria Mark, Shamsi T Iqbal, Mary Czerwinski, Paul Johns, Akane Sano, and Yuliya Lutchyn. 2016. 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*. 1717–1728.
[55] Kate McCready and Claire Stewart. 2017. 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* (Mar 2017), 597–603. https://experts.umn.edu/en/publications/sheriff-irs-auditor-psychotherapist-hostage-negotiator-all-in-one
[56] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–23.

[57] Luke McDowell, Oren Etzioni, Alon Halevy, and Henry Levy. 2004. Semantic email. In *Proceedings of the 13th international conference on World Wide Web*. 244–254.
[58] Joseph Edward McGrath. 1984. *Groups: Interaction and performance*. Vol. 14. Prentice-Hall Englewood Cliffs, NJ.
[59] Thomas P Moran, Alex Cozzi, and Stephen P Farrell. 2005. Unified activity management: supporting people in e-business. *Commun. ACM* 48, 12 (2005), 67–70.
[60] Michael Muller, Casey Dugan, Michael Brenndoerfer, Megan Monroe, and Werner Geyer. 2017. 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*. 1009–1023.
[61] Michael Muller, Casey Dugan, Aabhas Sharma, Werner Geyer, and Thomas Erickson. 2017. 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).
[62] Michael J Muller, Werner Geyer, Beth Brownholtz, Eric Wilcox, and David R Millen. 2004. 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*. 375–382.
[63] Michael J Muller and Daniel M Gruen. 2005. Working together inside an emailbox. In *ECSCW 2005*. Springer, 103–122.
[64] Soya Park, April Yi Wang, Ban Kawas, Q Vera Liao, David Piorkowski, and Marina Danilevsky. 2021. Facilitating knowledge sharing from domain experts to data scientists for building nlp models. In *26th International Conference on Intelligent User Interfaces*. 585–596.
[65] Soya Park, Amy X Zhang, Luke S Murray, and David R Karger. 2019. Opportunities for automating email processing: A need-finding study. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[66] Ei Pa Pa Pe-Than, Laura Dabbish, and James D. Herbsleb. 2018. 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*. ACM, 305–308. https://doi.org/10.1145/3272973.3274083
[67] James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates* 71, 2001 (2001), 2001.
[68] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 131 (apr 2021), 25 pages. https://doi.org/10.1145/3449205
[69] Daniela Retelny, Michael S. Bernstein, and Melissa A. Valentine. 2017. No Workflow Can Ever Be Enough: How Crowdsourcing Workflows Constrain Complex Work. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (Dec 2017), 1–23. https://doi.org/10.1145/3134724
[70] Bahareh Sarrafzadeh, Ahmed Hassan Awadallah, Christopher H. Lin, Chia-Jung Lee, Milad Shokouhi, and Susan T. Dumais. 2019. Characterizing and Predicting Email Deferral Behavior. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 627–635. https://doi.org/10.1145/3289600.3291028
[71] Peter Scupelli, Sara Kiesler, Susan R Fussell, and Congrui Chen. 2005. Project view IM: a tool for juggling multiple projects and teams. In *CHI'05 extended abstracts on Human factors in computing systems*. 1773–1776.
[72] Shilad Sen, Werner Geyer, Michael Muller, Marty Moore, Beth Brownholtz, Eric Wilcox, and David R Millen. 2006. FeedMe: a collaborative alert filtering system. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 89–98.
[73] O. Seppälä, T. Auvinen, V. Karavirta, A. Vihavainen, and P. Ihantola. 2016. 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)*. 432–435.
[74] Preethi Srinivas. 2015. 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*. 105–108.
[75] Lucy Suchman. 1993. 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*. Springer, 1–14.
[76] Lucy Suchman. 1994. Speech acts and voices: Response to Winogradet al. *Computer supported cooperative work (CSCW)* 3, 1 (1994), 85–95.
[77] Lucy Suchman. 2007. *Human-machine reconfigurations: Plans and situated actions*. Cambridge university press.
[78] Sunny Tian, Amy X Zhang, and David Karger. 2020. A System for Interleaving Discussion and Summarization in Collaborative Document Writing. In *Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing*. 59–63.
[79] Carlos Toxtli, Andrés Monroy-Hernández, and Justin Cranshaw. 2018. Understanding Chatbot-mediated Task Management. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 1–6. https://doi.org/10.1145/3173574.3173632
[80] Marcie J Tyre and Wanda J Orlikowski. 1994. Windows of opportunity: Temporal patterns of technological adaptation in organizations. *Organization science* 5, 1 (1994), 98–118.
[81] Gina Venoglia, Laura Dabbish, J.J. Cadiz, and Anoop Gupta. 2001. Supporting email workflow. https://ininet.org/supporting-email-workflow-gina-danielle-venolia-laura-dabbish.html Last accessed 16 September 2017.

[82] Diana White and Joyce Fortune. 2002. Current practice in project management—An empirical study. *International journal of project management* 20, 1 (2002), 1–11.
[83] Steve Whittaker, Victoria Bellotti, and Jacek Gwizdka. 2006. Email in personal information management. *Commun. ACM* 49, 1 (2006), 68–73.
[84] Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. 2011. Am I wasting my time organizing email? A study of email refinding. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 3449–3458.
[85] Steve Whittaker and Candace Sidner. 1996. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada). 793–802. https://doi.org/10.1145/238386.238530
[86] Terry Williams. 2005. Assessing and moving on from the dominant project management discourse in the light of project overruns. *IEEE Transactions on engineering management* 52, 4 (2005), 497–508.
[87] Terry Winograd. 1987. A Language/Action Perspective on the Design of Cooperative Work. *Human–Computer Interaction* 3, 1 (1987), 3–30. https://doi.org/10.1207/s15327051hci0301_2
[88] Terry Winograd, Fernando Flores, and Fernando F Flores. 1986. *Understanding computers and cognition: A new foundation for design*. Intellect Books.
[89] Svetlana Yarosh, Tara Matthews, Thomas P Moran, and Barton Smith. 2009. What is an activity? Appropriating an activity-centric system. In *IFIP Conference on Human-Computer Interaction*. Springer, 582–595.
[90] zendesk 2007. Zendesk. https://www.zendesk.com/.
[91] Amy X Zhang and Justin Cranshaw. 2018. Making sense of group chat through collaborative tagging and summarization. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–27.
[92] Qianqia (Queenie) Zhang, Soya Park, Michael Muller, and David R. Karger. 2024. "How fancy you are to make us use your fancy tool": Coordinating Individuals' Tool Preference over Group Boundaries. *Proc. ACM Hum.-Comput. Interact.* 8, GROUP, Article 4 (feb 2024), 31 pages. https://doi.org/10.1145/3633069

A 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. [44].

A.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.2 The PM tool group

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

A.3 The TaskLight group

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

- Invite [authors] to a new Google drive Folder / - Create a new folder at Google drive - Invite [authors] as editors / We will use this folder over the 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 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