Plans Work in Mysterious Ways: Evaluating A Plan Mode for Spreadsheet Agents

Aayush Kumar*, Avik Dutta*, Sumit Gulwani†, Gustavo Soares†, Advait Sarkar‡, Emerson Murphy-Hill§
Microsoft
*Bengaluru, India †Redmond, WA, USA ‡Cambridge, UK §Sunnyvale, CA, USA
{t-aaykumar, avikdutta, sumitg, gustavo.soares, advait, emerson.rex}@microsoft.com

***Abstract*—Plan Modes have become standard features in agentic programming tools, allowing users to gain transparency and control by working with the agent to develop a plan before task execution. However, it remains unclear whether the benefits of this feature translate to end-user programming environments such as spreadsheets. Since spreadsheet programmers tend to work iteratively and care less about technical correctness, upfront planning may not fit into their workflows as easily. In this paper, we build a prototype of a Plan Mode for spreadsheet programming and evaluate it against a non-planning baseline through a within-subjects user study (N=24). We found that despite similar task outcomes with both tools, participants prefer Plan Mode across dimensions of creativity support and collaboration, especially for spreadsheet creation tasks. Our results suggest that these preferences stem from a shift from iterative refinement toward responding to clarifying questions when using Plan Mode, and we discuss the implications of these results for the future design of interactive planning modes for spreadsheet programming.**

***Index Terms*—Human-AI Collaboration, End-User Programming, Spreadsheet Programming, Human-AI Planning**

I. Introduction

AI tools are getting dramatically better at performing complex tasks autonomously. *Agentic* tools – tools that interact with their environment and iterate on their own outputs – exemplify this progress. But as these capabilities grow, so do the challenges users face in specifying task goals and communicating preferences to such tools [1]. To address these challenges, agentic coding tools such as Claude Code [2], VSCode Agent Mode [3], Cursor [4], and Cline [5] have adopted a *Plan Mode*: programmers work with the agent to develop a plan before it begins executing the task. This lets programmers control the agent's implementation approach and see the edits the agent will make before it makes them.

While Plan Mode has successfully become a part of the programmer's toolkit, it remains unclear whether this affordance ports over smoothly to end-user programming domains such as spreadsheet programming. Unlike professional programmers, end-user programmers tend to work more iteratively, redefining their task constraints in an emergent manner as they work through their tasks [6]. End-user programmers may also lack the computational thinking skills needed to work with AI tools effectively [7]. Further, while one of the primary benefits of Plan Mode for programmers is gaining transparency into the technical implementation approach taken by the agent, spreadsheet programmers care less about technical correctness and more about the practical utility of the workbook [8].

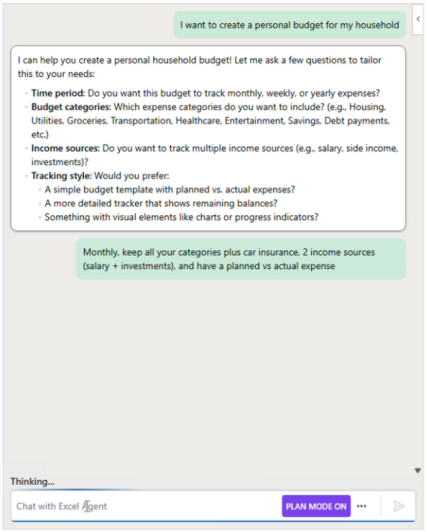
On the other hand, an interactive planning mode may also have benefits specific to spreadsheet programmers. Since end-user programmers struggle in turning ideas into working programs [6], planning can help scaffold complex spreadsheet tasks. Prior work indicates that planning-related affordances such as shared artifacts editable by both agent and user [9], [10] and clarifying questions asked by the agent [11]–[13] can improve user experiences when using AI tools in spreadsheet programming and related domains such as open-ended data analysis. Despite this, the impact of planning affordances on spreadsheet task outputs remains unclear from prior work. Further, to our knowledge, explicit planning modes for agentic spreadsheet programming tools remain unexplored in prior research. In this paper, we begin to address these gaps by 1) articulating a set of design features for plan modes in spreadsheet programming and implementing these features in a prototype, and 2) evaluating the effects of our Plan Mode prototype on process, outcomes, and experience in complex open-ended spreadsheet tasks.

In a controlled mixed-methods study (N=24) comparing Plan Mode with a baseline agentic tool ('Act Mode') for spreadsheet programming, we found that Plan Mode changed how participants exchanged information with the tool, shifting requirements elicitation from iterative refinement toward responding to clarifying questions. We also found that participants preferred Plan Mode over the baseline across dimensions of creativity support and human-machine collaboration. Despite this, we did not find any notable differences in the final workbooks created by participants, indicating that the benefits of Plan Mode primarily emerge from its interaction mechanisms, such as clarifying questions and UI elements.

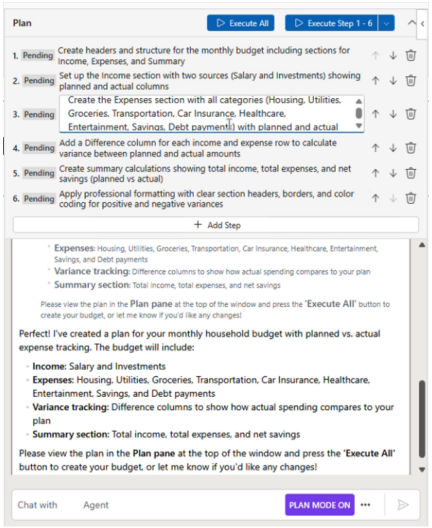
II. Related Work

A. Interactive Human-AI Planning

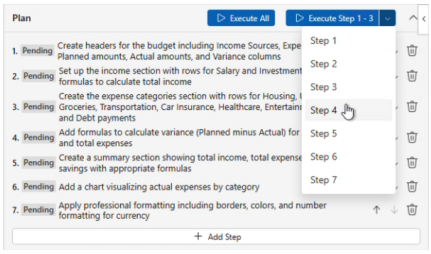
Prior work has explored interactive tools for human-AI collaborations on technically complex tasks across domains such as programming [14]–[17], data analysis [10]–[12], [18], [19], and financial tasks [9], [20]. Many of these tools explicitly or implicitly involve some form of interactive planning to facilitate smoother collaborations .This is especially true when the AI tool is an *agent*, capable of making complex programs autonomously. Prior research suggests that planning can help



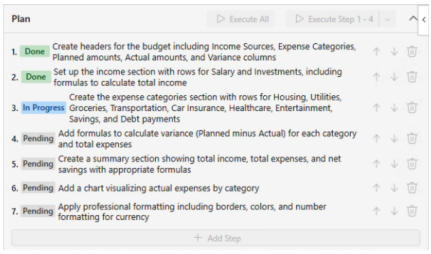
(a) The Plan agent asks clarifying questions to resolve any ambiguities in the user's query



(b) Users can directly manipulate the plan artifact



(c) Partial execution allows users to evaluate the plan at intermediate stages



(d) The Act agent updates the status of the plan during execution

Fig. 1: Interacting With Plan Mode

users communicate their expectations of how a task should be performed and gain transparency into the agent's actions [21]. Such planning affordances are often reified through shared artifacts editable by users and AI agents that represent actions taken/to be taken by the agent. For example, He et al. [20] and Mozannar et al. [21] introduce tools with an iterative loop where the agent proposes a plan that users can update before execution. Cocoa [22] maintains a similar shared plan, but further allows for iteration on the plan at intermediate stages during execution.

Planning affordances have become prominent for coding tools in particular. While planning collaboratively with coding agents enables developers to mitigate risk and ensure that the agent's implementation approach aligns with their expectations [23], even non-collaborative planning by the LLM itself can improve performance on coding tasks [24]. This has led to the adoption of Plan Modes in many popular coding agents, such as Claude Code [2], VSCode Agent Mode [3] and Cursor [4].

Despite this, prior work is mixed on the benefits of explicit planning stages in human-AI collaborations, indicating that while planning can lead to higher task clarity, transparency, and steerability, it can also add cognitive burdens for users [10]. Further, users can mistrust incorrect plans that look plausible [20]. Such errors can arise from a lack of tacit knowledge and failing to account for user assumptions [16], [22], [25]. In this paper, we extend this line of work to end-user spreadsheet programming by comparing a Plan Mode for spreadsheet agents to a non-planning baseline.

Clarifying questions asked by agents to gain information about the user's task have emerged as an important affordance for improved experiences. Such questions represent a more implicit form of planning where the agent identifies and resolves ambiguities in a task before execution, as opposed to proceeding with execution based on assumptions and expecting the user to flag issues post-hoc. For example, in PAIL [15], users must answer questions related to design choices before proceeding with implementation. The authors of Dango [11] and DataSpeck [18] report that asking clarifying questions led to improved intent clarification and efficiency in data tasks. Drosos et al. [12] report that dynamic prompt refinement through UI-based clarifications improves user experiences. Similarly, Cheng et al. [17] found that UI-based interactions during task execution make it easier for participants to explore new ideas in open-ended coding tasks. Yet, it remains unclear whether and how such clarifying questions affect the quality of task outputs [12], [15]. In this work, we begin to fill this gap by measuring participant outputs across a clarifying question-based Plan Mode and a baseline.

B. AI tools for Spreadsheet Programming

Prior work reveals a variety of AI-based spreadsheet programming tools with different interaction affordances. SheetCopilot [26] and SheetAgent [27] are autonomous AI tools capable of performing complex spreadsheet tasks. The Invisible Mentor [28] provides spreadsheet users with suggestions for their workflow based on screen recordings, while the Table Illustrator [29] interacts with participants using a puzzle-based metaphor to build plain tables. Other AI tools use some form of interactive intent clarification – Drosos et al. [12] use a UI for prompt refinement, TableTalk [13] gathers user feedback from users at regular stages of spreadsheet development, and Liu et al. [14] use editable task-decomposition based plan representations for limited-scope tasks. However, to the best of our knowledge, as of June 2026, no existing spreadsheet agent contains an explicit Plan Mode that asks clarifying questions to create a persistent plan artifact (we exclude Excel Copilot [30] here since its Plan Mode is based on the findings of this work). For example, while Shortcut AI [31] contains an explicit planning mode, this feature aims to translate user intent into a detailed implementation-heavy prompt rather than a persistent plan, and while Claude for Excel [32] can ask clarifying questions and present a 'plan' within its response

for execution, this plan is not persistent and this feature is triggered only implicitly and at the agent's discretion. In this paper, we begin to fill this gap by building and evaluating one such Plan Mode for a spreadsheet programming agent.

Despite the prominence of Plan Modes for programming, there is reason to believe human-AI planning may have different results when done by end-user programmers rather than by professional programmers. Ko et al. [6] report that the benefits of explicit specifications might be unclear with end-user programmers. Since end-user programmers' primary motivations are their own personal goals, not the software itself, they tend to discover their requirements in an emergent fashion during execution. Similarly, Pandita et al. [8] report that spreadsheet programmers tend to build spreadsheets in a piecemeal manner, prioritizing practical use over technical correctness. Thus, upfront planning may not align with users' workflows, especially for complex tasks. On the other hand, interactive planning may also have its own unique benefits for end-user programmers. Sarkar et al. [7] report that using AI coding tools effectively requires computational thinking skills that end-user programmers might not have, such as decomposing problems into subproblems. Plan Mode can support users in building and enacting such skills. Similarly, Plan Mode can act as an affordance to translate users' ideas into workable programs, something that end-users struggle with [6]. Since end-user programmers find it difficult to comprehend AI outputs and spreadsheets not authored by themselves [33], [34], Plan Mode may also make it easier for end-users to verify the agent's output. In this paper, we begin to address these hypotheses by measuring participants' experiences and outputs when using Plan Mode for complex open-ended spreadsheet tasks.

III. The Plan Mode Prototype

In an agentic tool that can perform technically complex tasks autonomously, 'Plan Mode' refers to a feature that allows users to retain control and transparency over the agent's actions by developing a plan before task execution. This usually involves an interactive planning process which culminates in the agent presenting a plan for users to review based on the user's query, context, and responses to clarifying questions asked by the agent, allowing users to catch any incorrect assumptions early. After agreeing on a plan, the agent leaves Plan Mode and starts executing the task adhering to this plan, providing users with transparency into the edits the agent makes before it makes them. In this subsection, we describe how our prototype for Plan Mode for spreadsheet development in Microsoft Excel implements these goals through an illustrative example (§III-A), the instructions we provided to the agent (§III-B), the design features of our tool (§III-C), and the tool's underlying implementation (§III-D).

A. Illustrative Example

Khushi is creating a spreadsheet to keep track of her household expenses. She decides to use Plan Mode for this task, and submits the query "I want to create a personal budget for my household". The agent in Plan Mode replies with a list of clarifying questions about her budget, such as the time period and income sources. Based on her personal context and taste, Khushi replies to all of these questions, skipping a subquestion on visualizations that she doesn't have an opinion on (Figure 1a). Based on these answers, the Plan Mode agent builds a plan for Khushi to review. Khushi notices that the Expenses section includes a category for debt payments. Since she doesn't have any debts, she decides to remove this category by editing the relevant step in the plan (Figure 1b). After reviewing the plan, she further decides that she wants a chart to track her spending across different categories, and expresses this in the chat to the Plan Mode agent, which then updates the plan to reflect this. Now satisfied with the plan, Khushi decides to execute the plan up to step 4 (Figure 1c) and then assess whether the budget seems to be heading in the right direction. As the agent executes, she follows along the status of the steps as they complete (Figure 1d). Once the agent is done, Khushi reviews the spreadsheet and concludes that it matches what she wants. She then presses the 'Execute Plan' button to continue with the rest of the plan, and updates her actual data in the final spreadsheet to track her expenses.

B. Agent Instructions

We instructed the agent to follow a 3-step workflow across the conversation-

1) **Understand all task context**, primarily the current content in the workbook and any conversation history.
2) If the user's query seems ambiguous or has multiple valid approaches to implementation, **ask clarifying questions** to the user
3) Based on the user's answers and the workbook context, **create a step-wise plan** to execute the task for users to review.

This workflow is flexible and interactive, based around the user's actions. For example, if users prompt the agent to make changes to the plan, and these changes seem to have multiple interpretations, the agent may again ask clarifying questions before updating the plan.

To ensure the clarifying questions stage is interactive without overwhelming users, we took inspiration from prior work. TableTalk [13] scaffolds spreadsheet development by taking feedback from users at regular stages, and based on this feedback, presents previews of its intended edits to users for review. Generalizing this workflow, we designed Plan Mode to ask users for feedback through clarifying questions, and then present the plan for review (which can possibly lead to a second stage of clarifying questions, based on the user's feedback). We operationalized this by instructing the agent *not* to repeat any clarifying questions to the user, even if they are unanswered – if required, the model then makes an assumption about this question, which is surfaced to the user in the plan. This way, users can answer questions which are important to them, without feeling annoyed or overwhelmed by answering questions that may not be relevant to them. Unlike the Plan Modes of tools such as Claude Code and Cursor that require

users to answer all clarifying questions through multiple-choice UI cards, our prototype takes responses to clarifying questions through the chat.

Finally, we instruct the agent to restrict the generated plan to 7 steps and to one sentence per step, keeping the plan easy to read and edit. We also instruct the agent not to include low-level implementation details in the plan, enabling the user and the agent to collaborate on high-level requirements and design decisions for the task in Plan Mode.

C. Design Features

In this subsection, we lay out the design features for our Plan Mode prototype along with the rationale behind them. Some of these features were inspired by those of Plan Modes for commercial coding tools; we lay out the exact relation between our design features and those of such tools in Appendix A.

- **Manipulable persistent plan artifact**: Upon creating a plan, our prototype displays this plan in a separate pane of the chat (Figure 1b, top). This artifact is persistent through planning and execution of the plan, allowing users to refer to it at any point of the process. Further, users can also edit the plan directly – they can add, remove, reorder, and change the content of steps in the plan, allowing for an easy way to update requirements, especially for more minor changes. Prior work has also observed usability benefits associated with such editable plan representations [22]. However, while some tools such as Cursor and VSCode have persistent plans visible to users, most do not allow users to edit these plans in-place through direct manipulation.
- **Partial Execution**: To ensure that participants remain in control of the agent and the scope of its edits, we added a button that allows users to partially execute the plan (Figure 1c, top right). This allows users to iterate at different stages within the task rather than being forced to iterate on a complete task artifact or prematurely pause the agent's execution. Prior work has also found that partial execution within the interactive planning process can improve usability and efficiency of the tool [11], [22]. While complex programming tasks may only produce well-formed outputs once all pieces of the code are in place, participants can use intermediate outputs for complex spreadsheet tasks to interpret whether their output is heading in the right direction. At the time of writing (June 2026), most popular tools with Plan Modes do not have such a feature.
- **Dynamic Plan Status Updates**: When executing the plan, the agent dynamically updates the status of each step in the plan (Pending $\rightarrow$ In Progress $\rightarrow$ Done) as it is executed (Figure 1c), facilitating users in tracking the current state of execution. Tools with persistent plans such as Cursor often have a similar feature, though usually with binary statuses (done/ not done). As the agent often makes multiple spreadsheet edits for a single step, the 'in-progress' status indicates to participants why certain spreadsheet elements might appear incomplete while the agent works through a particular step of the plan.
- **Restrictions to edits**: Plan Mode has read-only access to the user's workbook. This allows the user to collaborate with the agent on a plan without worrying about any expected edits while building the plan.
- **User-controlled explicit mode switching**: To ensure the user is aware of and in full control of the current mode the agent is in, we include a prominent button in the chat bar of the tool (Figure 1a, bottom right). Users can click on this to toggle the mode on and off. Further, the agent only switches from Plan Mode to execution mode when users click one of the 'Execute Plan' buttons (Figure 1b, top right).

D. Tool Implementation

We built our prototype for Plan Mode as an add-on feature to a high-fidelity internal research prototype (hereafter referred to as the 'Act agent') of Excel Copilot [30], an agentic AI tool that can autonomously perform complex tasks within Excel. The Plan Mode prototype served as a way for us to iterate on a viable implementation of a Plan Mode feature for Excel Agent and to gather feedback on the potential benefits and drawbacks of such a feature. Plan Mode has now been released for production within Excel Copilot [30].

The Act agent uses a chat-based interface present in a pane to the right of a Microsoft Excel window; users can enter their queries into a chat box following which the agent performs some internal reasoning (surfaced to users), runs commands to read and make edits to the spreadsheet, and eventually responds to the query with a summary of its findings and changes made to the spreadsheet. The Act agent contains detailed instructions for reading and writing to Excel spreadsheets, and is instructed to execute the user's tasks immediately and autonomously. It does not have the capability to create or store plans for task execution. The Plan Mode feature uses the same interface with the addition of the Plan pane at the top of the chat (Figure 1b, top), and the 'Plan Mode ON/OFF' button at the right side of the chat input box (Figure 1a, bottom right).

In early experiments, we observed that providing a single agent with instructions for both planning and task execution led to the agent sometimes using tools it was not meant to use – for example, trying to edit the spreadsheet in plan mode or edit the plan during execution. We thus decided to build a separate Plan agent for Plan Mode to cleanly separate the instructions for these two phases. When users interact with the tool in Plan Mode, they interact with the Plan agent, and when they switch to execution, the Act agent takes over. This Plan agent does not have access to any instructions or tools for editing the spreadsheet, but has access to an extra tool to render the plan in the Plan pane. Similarly, the Act agent does not have access to the instructions for plan creation or to the tool for rendering the plan, but has read-only access to the plan along with a tool to update the status of the steps of the plan in the UI. Both the Plan agent and the Act agent share the same conversation history so that they have the same context. We used the LLM Claude Sonnet 4.5 for all sessions in our study.

TABLE I: Participant Demographics

Dimension	Details
Age	18–24: 1; 25–34: 9; 35–44: 9; 45–54: 2; 55–64: 3
Gender	Men: 13; Women: 11; Additional genders: 0
Ethnicity	White/Caucasian: 14; Asian or Pacific Islander: 5; Black or African American: 4; Hispanic or Latino: 1
Region	US: 13; UK: 7; Canada: 2; France: 1; Australia: 1
Industry	Technology: 8; Healthcare: 4; Finance: 4; Retail: 2; Marketing/Advertising: 1; Logistics/Supply Chain: 1; Other: 4

IV. Methodology

Plan Mode involves a fundamentally distinct interaction paradigm when collaborating with an agent on a task; rather than entering and resolving queries one-by-one, users now work with the agent in a structured multi-turn workflow. Yet, the technical capabilities of the agent in terms of spreadsheet actions and base LLM remain the same across modes. Our evaluation thus assess how these differences and similarities manifest in practical usage through the following research questions:

RQ1: How does interactive planning affect how end-user programmers **exchange information** with AI tools on open-ended spreadsheet tasks?

RQ2: How does interactive planning affect the **final outputs** created by end users when working with AI tools on open-ended spreadsheet tasks?

RQ3: How does interactive planning affect end users' **perceived experiences** when working with AI tools on open-ended spreadsheet tasks?

To do so, we conducted a controlled mixed-methods within-subjects user study (N = 24). In each session, participants completed one task each with Plan Mode and the Act agent (hereafter referred to as 'Act Mode'). All sessions were conducted in February 2026.

A. Participants

Participant demographics are described in Table I. We recruited participants through an online user study recruitment platform (PlaybookUX, playbookux.com). Through a screener survey (available in supplementary material), we ensured that our participants used spreadsheets at least occasionally (more than once a week) and that they were familiar with creating and manipulating spreadsheets using spreadsheet software. We also ensured that participants had experience with using AI tools previously. Further, we excluded participants who were 'beginners' with spreadsheet software (that is, those who were not familiar with formulas), so that they were capable of interpreting the edits made by the tool.

B. Tasks

For our study, we aimed to use tasks that are open-ended and broadly subject to personal context and preferences – since such tasks have multiple valid approaches and solutions, planning can have an important role to play in shaping the final output. We further used two distinct types of tasks to understand the impact of planning across different tasks: **Creation** tasks in which participants had to create workbooks from scratch, and open-ended **Analysis** tasks in which participants have to perform data analysis on a large dataset based on their subjective preferences to pick final results. These correspond to the "Creation - Artifact" and "Information - analyze" categories of the knowledge worker LLM use taxonomy by Brachman et al. [35] and are the most directly supported activities by the Act agent.

We further sought tasks that could be completed in around 15 minutes, and use typical spreadsheet artifacts (such as formulas and charts) without requiring too much technical complexity. Thus, we used the following two **Creation** tasks: 1) **Budget** : Participants were instructed to create a personal budget, with the objective of creating a useful spreadsheet to use to track their expenses, and 2) **Schedule** : Participants were instructed to create a personal schedule with the objective of using the spreadsheet to manage their time more efficiently.

Similarly, we used the following two **Analysis** tasks: 1) **Vacation** : Participants were provided with a large dataset with details about vacation destinations (such as things to do, temperature, etc.) as well as details about accommodations in these destinations (price, ratings, etc.), and were asked to plan their next vacation based on the details in the data. 2) **Movie** : Participants were provided with a dataset of movies with details such as genres, ratings, popularity, etc. and asked to find a movie for them to watch. We include the task descriptions provided to participants and the datasets for the analysis tasks in the supplementary material.

In each session, participants performed either both **Creation** tasks or both **Analysis** tasks. We paired tasks within the same type so that each participant's comparisons across different modes were made on comparable tasks, isolating the effect of mode from the effect of task type. We balanced the type of tasks as well as the order of the specific tasks and tools participants used each session.

C. Protocol

All user study sessions were conducted by the first author through a video conferencing tool, and all participants worked on their tasks by remotely controlling the study administrator's screen. Each session started with the study administrator giving participants a tutorial of the first tool participants used (Plan Mode/ Act Mode), and answering any questions they had about the tool. While Act Mode consisted primarily of a familiar chat-based interface, Plan Mode consisted of potentially unfamiliar UI elements such as the Plan pane (Figure 1b, top). Thus, our tutorial for Plan Mode also included a demonstration of the different UI elements of the tool. Following this, the study administrator described the first task to participants, and instructed them that they had around 15 minutes to work on their task (in practice, we allowed up to 20 minutes for participants to work on their tasks). We also instructed participants to "think aloud" during their tasks so as

Origin	
Label	**Description**
proactive	User mentions the requirement proactively, not based on any message, plan, or clarifying question presented by the agent
clarification	User mentions the requirement as a reply to a clarification question by the agent
plan	User mentions the requirement after seeing a plan generated by the agent [only for Plan Mode]
refine	User mentions the requirement after seeing edits the agent made to the workbook

Iteration	
Label	**Description**
new	The first time the requirement is mentioned
follow-up	Follow-up on an existing requirement (e.g., deeper or complementary information)
reiteration	The user is describing the same requirement again without any changes
change	The user describes making changes to requirement mentioned earlier

TABLE II: User requirement coding

to understand their process and reactions to the tool. If participants mentioned that they were satisfied with the workbook before the 20-minute time limit (in practice, this happened in most cases), the study administrator asked them if they would like to make any other changes to the workbook. If participants still indicated that they would not make any changes beyond data entry (such as entering their real expense amounts for the **Budget** task), they moved on to the second task, for which the study administrator repeated the same process. After both tasks were complete, participants were instructed to fill out a post-study questionnaire. We asked participants to answer the questionnaire questions based on their experience with the tool rather than their experience working on the task to more accurately measure their preferences for the tool. Further, the study administrator asked participants to walk through their answers to understand the reasons behind their preferences. Each session was scheduled for one hour, and took between 30-60 minutes based on how much time participants took to complete their tasks.

D. Data Collection and Analysis

We collected several sources of data from our study, as described below.

1) Conversation Logs: We collected the conversation logs for participants' interactions with both tools. We used this data to aggregate general information about each task, such as the number of turns, duration, and number of tokens created by the LLM (approximate calculation by dividing the generated reasoning, messages, and tool call arguments by 4).

We also used this data to code the unique requirements participants expressed across their prompts. For **Creation** tasks, we consider a requirement to be a new *type* of information mentioned by the user (such as expense categories, a specific chart, etc.) while for **Analysis** tasks, we consider a requirement to be any filtering criteria or sorting criteria for a particular column in the dataset. We also coded the origin of these requirements and the iteration type of the requirement, as described in Table II.

While the codebook for requirement origin was based on the interaction mechanisms of Plan Mode and Act Mode, the codebook for requirement iteration was initially developed by the first author through open coding of two conversation logs, and refined to form the final version after coding two more such logs. To validate these codebooks, two authors independently coded a stratified sample of 12.5% of the data (6 out of 48 conversations, covering all tasks and treatments). Disagreements after the first round of coding were resolved by updating the codebook based on discussion between the two raters [36], followed by a second round of coding. We used Cohen's Kappa to measure inter-rater reliability, yielding scores of $\kappa = 0.93$ (quadratic-weighted) for extracting requirements from raw conversation transcripts, $\kappa = 0.62$ for the requirements iteration coding and $\kappa = 0.85$ for the requirements origin coding, which indicates substantial to near-perfect agreement [37].

2) Spreadsheets Created: We collected all spreadsheets created by participants in their tasks. We further analyze the *features* present across these spreadsheets, applying the following criteria:

*a) **Creation** tasks:* We applied a hierarchical coding system to analyze the contents of each workbook, noting the number of top-level spreadsheet artifacts (sheets, tables, and charts) present across these workbooks as well as the unique features present in these artifacts. Each feature consists of one primary code (e.g., 'expenses') associated with one or more secondary codes (e.g., 'taxes'). We include the complete feature codebook in the supplementary material. For each task, the codebook was initially developed through an open coding of two workbooks generated by participants. It was then refactored to be hierarchical based on repeating themes across the open coding, and refined across re-coding the same two workbooks along with one additional workbook.

*b) **Analysis** tasks:* We note each information column used by the participant in picking their final results, along with whether these columns were associated with any filtering criteria, sorting criteria, or both. Often, the agent created a score using information from several columns – in these cases, we counted all these columns used as features.

3) Post-study questionnaire: We adapted the Mixed-Initiative Creativity Support Scale [38] for our study. The first 10 questions of this scale focus on creativity support along 5 dimensions ; the next 4 capture Human-Machine Collaboration; and the last 4 use paired statements to elicit perceptions of the final outputs.

To compare experiences between modes rather than capture participants' experiences with agentic spreadsheet tools in general, we changed the phrasing of the first 14 questions of the survey to be comparative across tools (for example, *"I enjoyed using this tool more"* rather than *"I enjoyed using this tool"*), and used a preference-based 5-point scale to collect responses [*'Act Mode', 'Somewhat Act Mode', 'Neutral', 'Somewhat Plan Mode', 'Plan Mode'*]. For the remaining

#	Dimension	Question(s)
		Creativity Support Questions
1	Enjoyment	"I would be happy using this tool more on a regular basis."
2		"I enjoyed using the tool more."
3	Exploration	"It was easier for me to explore many different ideas, options, designs, or outcomes, using this tool."
4		"This tool was more helpful in allowing me to track different ideas, outcomes, or possibilities."
5	Expressiveness	"I was able to be more creative while doing the activity inside this system or tool."
6		"This tool allowed me to be more expressive."
7	Attention*	"My attention was more focused on the task."
8		"My attention was more focused on the tool."
9	Worth	"I was more satisfied with what I got out of this tool."
10		"What I was able to produce was more worth the effort I had to exert to produce it with this tool."
		Human-Machine Collaboration Questions
11	Communication	"I was able to more effectively communicate what I wanted to the system."
12	Alignment	"I was able to steer this tool better towards output that was aligned with my goals."
13	Agency	"At times, I felt that this tool was steering me towards its own goals."
14	Partnership	"At times, it felt like this system and I were collaborating as equals."
		Exploratory Questions
15	Contribution*	"I am creatively responsible for this spreadsheet" vs "The agent is creatively responsible for this spreadsheet"
16	Satisfaction	"I'm very unsatisfied with the spreadsheet" vs "I'm very satisfied with the spreadsheet."
17	Surprise	"The spreadsheet was what I was aiming for" vs "The spreadsheet outcome was unexpected."
18	Novelty	"The spreadsheet is very typical" vs "The spreadsheet is very novel."
19	Capability*	"I could have created the spreadsheet by myself" vs "I could not have created the spreadsheet without using this tool"

TABLE III: Post-study questionnaire used in our study, based on [38] (* indicates dimensions modified for our study)

questions, we asked participants to select the statement that most closely matched their experience for each tool. We also adapted the 'Immersiveness' dimension to a more relevant 'Attention' dimension, modified the phrasing of the 'Contribution' statement, and added a new dimension for 'Capability' (details in supplementary material). The final questionnaire used is described in Table III.

4) Session Recordings: We used video, audio and screen recordings of the study sessions to note whether and how participants interacted with the Plan Mode UI, and to code participant utterances during the think-aloud portions of the study.

5) Limitations: The small sample size (N=24) of our study limits the generalizability of our results; we accept this limitation since it allowed us to conduct in-depth qualitative coding of participant requirements and workbooks, which would be difficult to scale to larger samples. Another limitation of our study is that almost all participants in our study had not used *agentic* tools for spreadsheet programming previously, which may have affected participants' experiences with the tool. We also restricted our tasks to those which could be completed in a short time frame (15 minutes), which may have limited our ability to observe the differences between the two modes. This is evident in the fact that while the comparative measurement instrument (Figure 2a) shows differences, the non-comparative instrument (Table VIII) shows near-identical ratings across tools.

Further, the fact that Plan Mode had unfamiliar UI elements as compared to Act Mode and that our tutorial for Plan Mode covered UI elements while the tutorial for Act Mode did not is a confound that may have affected participants' perceptions of the tool. Additionally, while we report descriptive statistics related to spreadsheet artifacts and total number of requirements, the time-capped nature of each task may have affected these numbers, especially for the participants who did not fully complete the task (though the number of such participants was low). Finally, we acknowledge our positionality as researchers at Microsoft, a technology company that develops AI tools, and any unconscious biases that this positionality might engender in the interpretation of our data.

	Proactive	Clarification	Plan	Refine
Plan Mode	3.1	4.1	0.7	1.3
Act Mode	4.3	0.2	-	3.1

TABLE IV: Mean number of requirements by origin across modes

	New	Follow-up	Reiteration	Change
Plan Mode	6.5	2.0	0.1	0.6
Act Mode	6.4	0.7	0.1	0.4

TABLE V: Mean number of requirements by iteration across modes

V. Results

A. RQ1: Interaction Patterns

RQ1 asks how interactive planning affects how end-user programmers exchange information with AI tools on open-ended spreadsheet tasks. We explore these patterns across three sub-aspects: the differences in how participants expressed their requirements to the agent across modes (§V-A1), whether and how different information exchange patterns changed how participants iterated on their spreadsheets, (§V-A2), and whether participants used the planning UI affordances to exchange information (§V-A3).

1) Requirements Elicitation: Our results suggest that participants using Plan Mode expressed both more requirements and more detailed requirements before the agent made edits to the workbook, a shift largely driven by the presence of clarifying questions.

	Proactive	Clarification	Plan	Refine
Plan Mode	46%	37%	7%	10%
Act Mode	64%	1%	-	35%

TABLE VI: Distribution of origin of **new** requirements across modes.

Participants expressed a similar number of unique requirements across modes (Table IV, "New"), but arrived at them through different interaction patterns. Participants were highly responsive to clarifying questions in Plan Mode, directly answering 88 of the 99 questions asked by the agent. As a result, participants in Plan Mode expressed 44% of their requirements in response to clarifying questions, while only 14% emerged after the agent had already edited the spreadsheet. In contrast, participants in Act Mode expressed over 41% of their requirements as refinements to existing spreadsheet edits (Table IV).

This shift in requirements from refinements to clarifying questions in Plan Mode is also reflected in the origin of new requirements expressed by participants. When using Plan Mode, 10% of participants' new requirements came from refinements, while 37% came from clarifying questions. This was flipped for participants using Act Mode, for whom 35% of new requirements stemmed from refinement requests.

Beyond eliciting new requirements, clarifying questions also helped participants dive into more depth on these requirements. When using Plan Mode, participants provided more 'follow-ups' (updates and complementary information) to their requirements (Table V, 'Follow-up'), and the majority (82%) of follow-ups on requirements in Plan Mode came from responses to clarifying questions. Thus, clarifying questions not only surfaced newer requirements, but also drew out additional detail on existing ones.

2) Refinement Patterns: Our results suggest that spending time on upfront planning enabled participants to build spreadsheets that were more aligned to their goals. Participants using Plan Mode used more turns (5.3 vs 3.6, on average) and took more time (10.4 vs 8.8 minutes, on average) across all tasks in our study. At the same time, we found that participants spent more turns refining the spreadsheet when working with Act Mode (2.2 vs 1.4 on average) and that they often didn't iterate on plans at all– only 10 out of 24 participants made any changes to the first proposed plan through chat messages or the UI.

This reduction in refinement also results in a reduction in cost. Despite the fact that participants expressed a similar number of total requirements for both treatments, the approximate number of tokens used in the LLM's reasoning, messages, and spreadsheet edits was much lower with Plan Mode than with Act Mode across all tasks (11.0k v 17.0k on average).

Our results also suggest that participants found the 'Budget' and 'Vacation' tasks to be lengthier than the 'Schedule' and 'Movie' tasks respectively – on average, they used more conversation turns (4.9 vs 4.0) and spent more time (11.9 vs 7.3 minutes) on these tasks.

Task	Mode	Sheets	Tables	Charts
Budget	Plan Mode	1.5	4.5	2
	Act Mode	2.5	10.5	2
Schedule	Plan Mode	1.5	3	0
	Act Mode	1	4.5	0

TABLE VII: Median number of sheets, tables, and charts created by mode and task.

3) Using the UI: We found that while participants did not tend to use the UI features of Plan Mode, they often still appreciated the control these features offer. 6 out of 24 participants chose to partially execute the plan by using the 'Execute Step 1-x' button (see Figure 1c, top right). Further, only 2 out of 24 participants edited the plan representation in the Plan Mode UI. Yet, 14 out of 24 participants spoke positively about the ability to edit/rearrange the steps of the plan as a way for them to be able to control or steer the agent better. As P8 said, *"I could see, you know, it's like creating a game plan, and if I'm missing something from my game plan, I can add to my game plan."*

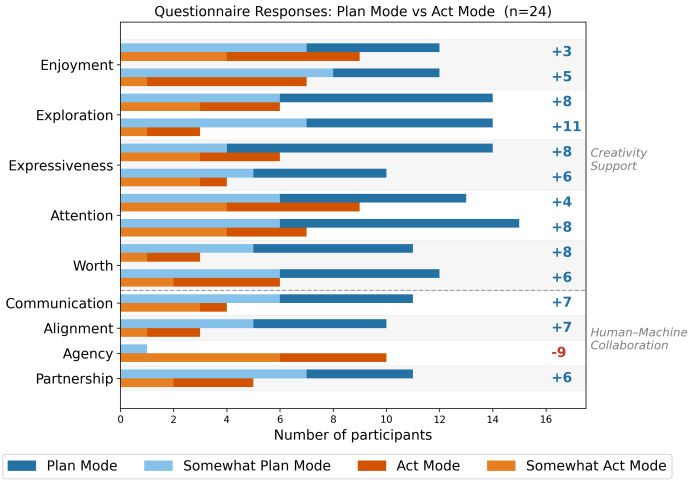
B. RQ2: Differences in generated spreadsheets

Overall, we found little difference in the distribution of features in participants' spreadsheets across modes, indicating that working with Plan Mode does not lead to the creation of more diverse or personalized spreadsheets. However, we did find some task-wise differences, with the lengthier tasks generally using/containing more features.

*1) **Analysis** Tasks:* Participants used data from more columns for the 'Vacation' task (7.3 unique columns for filtering and sorting on average) than for the 'Movie' task (3.8 columns for filtering and sorting on average). We did not find any notable differences in the number of columns used for filtering and sorting across modes (5.3 in Plan Mode vs 5.8 in Act Mode on average). We also found that participants used similar columns for their analysis in each task across both modes.

*2) **Creation** Tasks:* For the creation tasks, we found that participants using Act Mode tended to create more spreadsheet artifacts than those using Plan Mode (Table VII), particularly for the Budget task. However, the spreadsheets created with Act Mode contained only a moderately larger number of total features across both tasks (median values for Plan vs Act: 9 vs 10.5 for the Budget task; 3 vs 4.5 for the Schedule task). Many of the extra features for the Schedule task in Act Mode correspond to meta-information about the spreadsheet and instructions rather than more types of data. Thus, overall, the spreadsheets generated contained similar features across modes despite Act Mode using more spreadsheet artifacts.

We also found that participants created more artifacts when working on the **Budget** task (48 sheets, 138 tables and 30 charts) as compared to when working on the **Schedule** task (23 sheets, 67 tables and 7 charts), further indicating that participants found the **Budget** task to be lengthier.



(a) Questionnaire responses (Part 1)
(Neutral responses not shown)

.

TABLE VIII: Questionnaire responses (Part 2).

Category	Statement	Plan	Act
Contribution	The agent is creatively responsible for this spreadsheet	8	10
	I am creatively responsible for this spreadsheet	16	14
Satisfaction	I'm very satisfied with the spreadsheet	22	23
	I'm very unsatisfied with the spreadsheet	2	1
Surprise	The spreadsheet was what I was aiming for	19	20
	The spreadsheet outcome was unexpected	5	4
Novelty	The spreadsheet is very typical	12	14
	The spreadsheet is very novel	12	10
Capability	I could have created this spreadsheet by myself	17	18
	I could not have created this spreadsheet without AI	7	6

Fig. 3: Questionnaire responses by participants.

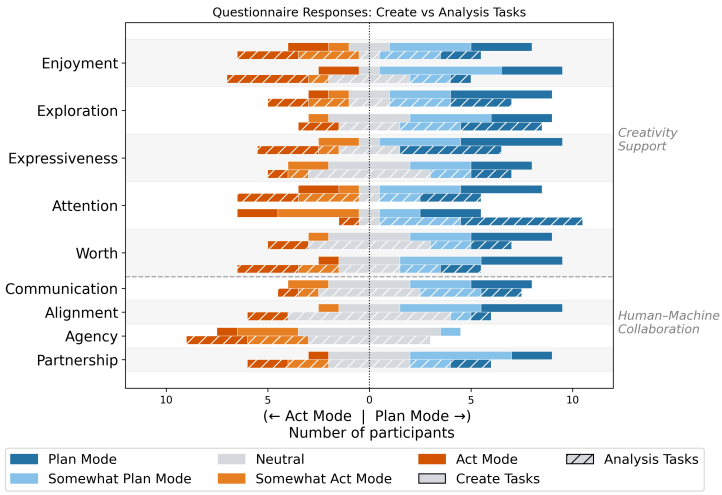


Fig. 4: Participant preferences across task type.

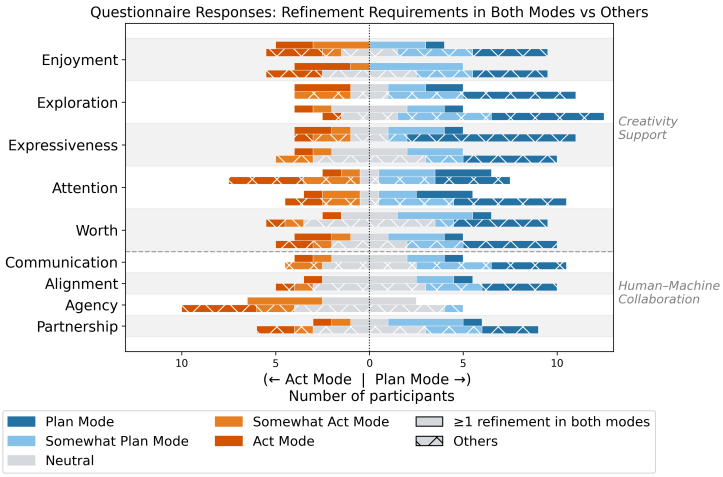


Fig. 5: Participant preferences based on refinement requirements.

C. RQ3: Perceived experiences

We found that participants preferred Plan Mode across almost all dimensions of creativity support and human-machine collaboration and had similar perceptions on the spreadsheets they created (Table VIII), though this relationship was moderated by both the types of tasks participants worked on as well as their own interaction styles. Figure 3 shows a summary of participants' responses on our post-study questionnaire. Participants preferred Plan Mode over Act Mode across all five dimensions on creativity support (each dimension corresponds to two questions) and all four questions on effective collaboration. Participants answered Act Mode more frequently on the 'Agency' question, since this asks about the extent to which the tool steers the user towards its own goals. The only exception is the second question of the Attention scale – participants indicated that working with Plan Mode required more of their attention on the tool, which may be due to the extra UI elements present in Plan Mode as compared to Act Mode. Table VIII also indicates that despite participants preferring Plan Mode for creativity support, they feel a similar amount of creative responsibility and satisfaction with the output when working with Plan and Act Mode.

We found that participants' preference for Plan Mode was moderated by several factors:

- **Task Type:** Participants preferred Plan Mode more for **Creation** tasks across all dimensions, while they were more ambivalent for **Analysis** tasks, as highlighted in Figure 4. The contrasts across the dimensions of enjoyment, worth, alignment, and partnership are particularly stark – those working on **Analysis** tasks were neutral or leaned towards Act Mode for these dimensions, while those working on **Creation** tasks strongly preferred Plan Mode. These specific dimensions indicate that participants working on **Creation** tasks felt more like they were interacting with a collaborative partner when working with Plan Mode. Further, while those working on **Analysis** tasks felt that they had to pay more attention to the tool rather than

the task when working with Plan Mode, those working on **Creation** tasks were neutral, further highlighting that participants working on **Creation** tasks felt the experience to be smoother and more collaborative when working with Plan Mode.

- **Task Length:** Participants preferred Plan Mode more on the tasks they spent more time and turns on (**Budget** and **Vacation** , see V-A2). Participants working with Plan Mode on such lengthy tasks particularly preferred Plan Mode across the dimensions of agency, exploration, worth and partnership, while other participants were more neutral on these dimensions. This indicates that Plan Mode provided participants with more control and a more effective collaboration when working on lengthy tasks.
- **Number of upfront requirements:** Participants who provided more requirements in their first query enjoyed Act Mode more than Plan Mode, and also felt that they could explore ideas and outcomes using Act Mode. We measured this by examining the preferences of those (n = 9) participants who provided more than the mean (3.7) number of requirements upfront in both their tasks. This suggests that Plan Mode was more helpful for participants who were not effective at one-shot prompting.
- **Number of refinements:** While participants who expressed requirements as refinements tended to be more neutral, others strongly preferred Plan Mode across all creativity support dimensions, as well as the communication dimension. Figure 5 shows the questionnaire responses for those participants (n=9) who expressed at least one requirement as a refinement across both modes, as compared to the other participants (n = 15). Thus, participants who followed the pattern of iterative refinement found that Plan Mode did not help them in being creative. 7 participants explicitly mentioned that they liked Act Mode since it facilitated iterative refinement in a way that was easier or quicker than with Plan Mode – it allowed users to view and update outputs in *"real time"* (P2). These preferences seemed to originate from the process of planning itself not matching participants' workflow. For example, P16 mentioned that *"Plan Mode seemed like you really had to plan, and if I'm gonna do that in an AI, I might as well just do that myself... Act Mode is more natural and freestyle. You can just change your mind on the fly."*

VI. Discussion

Our results suggest that using Plan Mode changed how participants collaborated with the agent on their tasks in terms of requirements and refinements and further resulted in a better perception of this collaboration, despite the final outputs created being similar. In this section, we discuss potential design interventions that might allow Plan Mode to engender higher quality outputs for users (§VI-A), why participants may have preferred Plan Mode despite the similarity in outputs (§VI-B), and finally some future research directions for the design and personalization of Plan Mode for end-user programming(§VI-C).

A. Design Implications for Plan Mode

Though participants expressed a similar number of unique requirements across modes, it is interesting to note where these new requirements came from – while all requirements when using Act Mode came from participants, 37% of requirements when using Plan Mode emerged from a shared origin (clarifying questions, asked by the agent and answered by the user) (Table VI). However, this new source of requirements did not manifest in the final outputs, indicating that the agent may not actually be contributing anything particularly new – it might be asking questions that users would answer anyway (perhaps during the refinement stage) or that the tool in Act Mode might implement anyway. This may be exacerbated by automation bias [39] – since participants answered most clarifying questions and usually did not iterate on the plan (Section V-A2), the outputs produced may be strongly influenced by the dimensions the agent thinks are important (expressed through clarifying questions) and follow an execution path the agent thinks is best (expressed through the plan which users did not modify). These homogeneous results might also indicate that agents across both Plan Mode and Act Mode may be reducing, rather than increasing, users' creativity [40].

Thus, the paradigm of clarifying questions should be modified to augment their thinking [41], [42]. Since spreadsheet users can struggle to discover new features in spreadsheet software [43], augmenting their thinking also provides an opportunity for users to do so in an environment scaffolded by an agent. For example, Drosos et al. [12] report that dynamically generated form-based clarifying questions can help participants reflect on their own assumptions, enabling them to explore alternative solutions. Similarly, Danry et al. [44] report that framing AI-generated explanations as questions can increase critical thinking; such a paradigm can fit naturally into a Plan Mode. Future work can explore how other such interfaces, which apply strategies like design friction [45] and cognitive forcing functions [46] to promote critical thinking, can augment users' thinking and creativity.

B. Control Without Action as Design Value

Despite the overall similarities in outputs and requirements between Plan Mode and Act Mode, participants indicated a preference for Plan Mode in the post-study questionnaire (Figure 2a). We observe a similar dichotomy in the usage of the UI features of Plan Mode – despite almost never using these features, participants indicated that the UI provided them with a sense of control over the agent (Section V-A3). Kazemitabaar et al. [10] report a similar finding in the computational notebook domain – though steering mechanisms for AI tools led to no difference in task performance metrics, participants in their study reported greater feelings of control with these mechanisms. This visual tracking and manual control over the agent's actions may serve as a way for users to overcome the "gulf of envisioning" with generative AI, allowing users to better operationalize their goals when collaborating with an agent. Thus, such UI features, even when not used, provide design value in and of themselves – though they may not

increase performance metrics, they may still improve the quality of the human-AI collaboration.

C. Tasks, Interaction Styles, and Mixed-Initiative Plan Mode Activation

Most implementations of Plan Modes leave entry and exit completely up to the user. Yet, end-user programmers might not be equipped to know *when* to utilize Plan Mode, since they may not have the skills to understand which tasks are complex and may benefit from planning [6], [7]. Thus, an important future direction is to investigate mixed-initiative paradigms for triggering Plan Mode.

Our results suggest that different types of spreadsheet programmers react to Plan Mode differently – participants who specified many requirements upfront and those who iterated on workbooks across both modes tended to enjoy Act Mode more than Plan Mode. These patterns resemble those of a selective information processing style [47], in which end-users tend to apply a more "depth-first" approach to problem-solving. This is in contrast to the comprehensive information processing style, in which end-users gather all information to form a complete understanding of the problem before proceeding – a more plan-like approach. Further, we found that participants preferred Plan Mode more when working on **Creation** tasks, which while similarly open-ended as the **Analysis** tasks, required more comprehensive spreadsheet edits. Future work should thus explore how agents can infer and utilize users' information processing style along with the context of their task to scaffold the integration of Plan Mode into their workflow. Such a paradigm would support end-user programmers in extracting the best out of their agents without imposing the additional metacognitive overhead of adjusting to different modes or features.

VII. Conclusion

In this paper, we presented a prototype of a Plan Mode for spreadsheet programming agents that allows users to plan their task with AI before they execute the task. In a within-subjects user study (N=24), we found that using Plan Mode affected how participants exchanged information with the AI and improveed their perception of the collaboration with the AI. Our findings have since informed the design and deployment of the Plan Mode feature of Excel Agent in Microsoft Excel, where we continue to observe positive feedback on the impact of planning for human-AI collaborations.

VIII. Acknowledgements

We would like to thank the whole feature crew of Plan Mode for Excel Agent for their support through this project. We especially want to thank Avani Reddy for their valuable insights and feedback throughout this project, Arturo Goicochea for their insights on the design of the Plan Mode prototype, as well as San Lee and Saloni Gupta for their help in implementing Plan Mode.

References

[1] G. Bansal, J. W. Vaughan, S. Amershi, E. Horvitz, A. Fourney, H. Mozannar, V. Dibia, and D. S. Weld, "Challenges in Human-Agent Communication," 2024.

[2] Anthropic, "Choose a Permission Mode – Claude Code Docs." https://code.claude.com/docs/en/permission-modes, 2026. Accessed: 2026-05-01.

[3] Microsoft, "Planning with Agents in VS Code." https://code.visualstudio.com/docs/copilot/agents/planning, 2026. Accessed: 2026-05-01.

[4] Cursor, "Plan Mode – Cursor Docs." https://cursor.com/docs/agent/plan-mode, 2026. Accessed: 2026-05-01.

[5] Cline, "Plan & Act Mode – Cline Documentation." https://docs.cline.bot/core-workflows/plan-and-act, 2026. Accessed: 2026-05-01.

[6] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, and S. Wiedenbeck, "The state of the art in end-user software engineering," *ACM Comput. Surv.*, vol. 43, Apr. 2011.

[7] A. Sarkar, A. D. Gordon, C. Negreanu, C. Poelitz, S. S. Ragavan, and B. Zorn, "What is it like to program with artificial intelligence?," 2022.

[8] R. Pandita, C. Parnin, F. Hermans, and E. Murphy-Hill, "No half-measures: A study of manual and tool-assisted end-user programming tasks in Excel," in *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 95–103, 2018.

[9] L. Reicherts, Z. T. Zhang, E. von Oswald, Y. Liu, Y. Rogers, and M. Hassib, "AI, Help Me Think—but for Myself: Assisting People in Complex Decision-Making by Providing Different Kinds of Cognitive Support," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), Association for Computing Machinery, 2025.

[10] M. Kazemitabaar, J. Williams, I. Drosos, T. Grossman, A. Z. Henley, C. Negreanu, and A. Sarkar, "Improving Steering and Verification in AI-Assisted Data Analysis with Interactive Task Decomposition," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, (New York, NY, USA), Association for Computing Machinery, 2024.

[11] W.-H. Chen, W. Tong, A. Case, and T. Zhang, "Dango: A Mixed-Initiative Data Wrangling System using Large Language Model," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), Association for Computing Machinery, 2025.

[12] I. Drosos, J. Williams, A. Sarkar, N. Wilson, S. Rintel, and P. Panda, "Dynamic Prompt Middleware: Contextual Prompt Refinement Controls for Comprehension Tasks," in *Proceedings of the 4th Annual Symposium on Human-Computer Interaction for Work*, CHIWORK '25, (New York, NY, USA), Association for Computing Machinery, 2025.

[13] J. T. Liang, A. Kumar, Y. Bajpai, S. Gulwani, V. Le, C. Parnin, A. Radhakrishna, A. Tiwari, E. Murphy-Hill, and G. Soares, "TableTalk: Scaffolding Spreadsheet Development with a Language Agent," *ACM Trans. Comput.-Hum. Interact.*, vol. 32, Dec. 2025.

[14] M. X. Liu, A. Sarkar, C. Negreanu, B. Zorn, J. Williams, N. Toronto, and A. D. Gordon, ""What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[15] J. Zamfirescu-Pereira, E. Jun, M. Terry, Q. Yang, and B. Hartmann, "Beyond Code Generation: LLM-supported Exploration of the Program Design Space," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), Association for Computing Machinery, 2025.

[16] A. Kumar, Y. Bajpai, S. Gulwani, G. Soares, and E. Murphy-Hill, "Why AI Agents Still Need You: Findings from Developer-Agent Collaborations in the Wild," in *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 432–444, 2025.

[17] R. Cheng, T. Barik, A. Leung, F. Hohman, and J. Nichols, "BISCUIT: Scaffolding LLM-Generated Code with Ephemeral UIs in Computational Notebooks," in *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 13–23, 2024.

[18] A. Rahman, K. Niinuma, and A. Gupta, "DataSpeck: An AI-Driven Human-in-the-Loop System for Automating Transformations in Data Conversion Workflows," in *Proceedings of the 2026 CHI Conference*

on Human Factors in Computing Systems, CHI '26, (New York, NY, USA), Association for Computing Machinery, 2026.
[19] I. Drosos, A. Sarkar, X. Xu, C. Negreanu, S. Rintel, and L. Tankelevitch, ""It's like a rubber duck that talks back": Understanding Generative AI-Assisted Data Analysis Workflows through a Participatory Prompting Study," in *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*, CHIWORK '24, (New York, NY, USA), Association for Computing Machinery, 2024.
[20] G. He, G. Demartini, and U. Gadiraju, "Plan-Then-Execute: An Empirical Study of User Trust and Team Performance When Using LLM Agents As A Daily Assistant," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), Association for Computing Machinery, 2025.
[21] H. Mozannar, G. Bansal, C. Tan, A. Fourney, V. Dibia, J. Chen, J. Gerrits, T. Payne, M. K. Maldaner, M. Grunde-McLaughlin, *et al.*, "Magentic-UI: Towards Human-in-the-loop Agentic Systems," *arXiv preprint arXiv:2507.22358*, 2025.
[22] K. J. K. Feng, K. Pu, M. Latzke, T. August, P. Siangliulue, J. Bragg, D. S. Weld, A. X. Zhang, and J. C. Chang, "Cocoa: Co-Planning and Co-Execution with AI Agents," in *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, (New York, NY, USA), Association for Computing Machinery, 2026.
[23] T. Dong, H. Sampath, J. Y. Lee, S. Y. Shi, and A. Macvean, "From correctness to collaboration: Toward a human-centered framework for evaluating ai agent behavior in software engineering," 2025.
[24] R. Bairi, A. Sonwane, A. Kanade, V. D. C., A. Iyer, S. Parthasarathy, S. Rajamani, B. Ashok, and S. Shet, "Codeplan: Repository-level coding using llms and planning," *Proc. ACM Softw. Eng.*, vol. 1, July 2024.
[25] S. Passi, "Agentic AI has a Human Oversight Problem," *Available at SSRN 5529058*, 2025.
[26] H. Li, J. Su, Y. Chen, Q. Li, and Z. Zhang, "SheetCopilot: Bringing Software Productivity to the Next Level through Large Language Models," 2023.
[27] Y. Chen, Y. Yuan, Z. Zhang, Y. Zheng, J. Liu, F. Ni, J. Hao, H. Mao, and F. Zhang, "SheetAgent: Towards A Generalist Agent for Spreadsheet Reasoning and Manipulation via Large Language Models," 2025.
[28] L. Yan, A. Head, K. Milne, V. Le, S. Gulwani, C. Parnin, and E. Murphy-Hill, "The Invisible Mentor: Inferring User Actions from Screen Recordings to Recommend Better Workflows," in *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, (New York, NY, USA), Association for Computing Machinery, 2026.
[29] K. Ferdowsi, J. Williams, I. Drosos, A. D. Gordon, C. Negreanu, N. Polikarpova, A. Sarkar, and B. Zorn, "COLDECO: An End User Spreadsheet Inspection Tool for AI-Generated Code," in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 82–91, 2023.
[30] Microsoft, "Excel Copilot." https://support.microsoft.com/en-US/excel/copilot/get-started-with-copilot-in-excel, 2026. Accessed: 2026-06-21.
[31] Shortcut, "Shortcut AI." https://shortcut.ai/, 2026. Accessed: 2026-06-21.
[32] Anthropic, "Claude for Excel." https://claude.com/claude-for-excel, 2026. Accessed: 2026-06-21.
[33] S. Srinivasa Ragavan, A. Sarkar, and A. D. Gordon, "Spreadsheet Comprehension: Guesswork, Giving Up and Going Back to the Author," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.
[34] A. Sarkar, "Will Code Remain a Relevant User Interface for End-User Programming with Generative AI Models?," in *Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2023, (New York, NY, USA), p. 153–167, Association for Computing Machinery, 2023.
[35] M. Brachman, A. El-Ashry, C. Dugan, and W. Geyer, "How Knowledge Workers Use and Want to Use LLMs in an Enterprise Context," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, (New York, NY, USA), Association for Computing Machinery, 2024.
[36] N. McDonald, S. Schoenebeck, and A. Forte, "Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, Nov. 2019.
[37] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
[38] T. Lawton, F. J. Ibarrola, D. Ventura, and K. Grace, "Drawing with Reframer: Emergence and Control in Co-Creative AI," in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23, (New York, NY, USA), p. 264–277, Association for Computing Machinery, 2023.
[39] R. Parasuraman and D. Manzey, "Complacency and Bias in Human Use of Automation: An Attentional Integration," *Human factors*, vol. 52, pp. 381–410, 06 2010.
[40] V. Padmakumar and H. He, "Does Writing with Language Models Reduce Content Diversity?," in *The Twelfth International Conference on Learning Representations*, 2024.
[41] A. Sarkar, "AI Should Challenge, Not Obey," *Commun. ACM*, vol. 67, p. 18–21, Sept. 2024.
[42] L. Tankelevitch, E. L. Glassman, J. He, A. Kittur, M. Lee, S. Palani, A. Sarkar, G. Ramos, Y. Rogers, and H. Subramonyam, "Understanding, Protecting, and Augmenting Human Cognition with Generative AI: A Synthesis of the CHI 2025 Tools for Thought Workshop," 2025.
[43] E. Murphy-Hill, D. Y. Lee, G. C. Murphy, and J. Mcgrenere, "How Do Users Discover New Tools in Software Development and Beyond?," *Comput. Supported Coop. Work*, vol. 24, p. 389–422, Oct. 2015.
[44] V. Danry, P. Pataranutaporn, Y. Mao, and P. Maes, "Don't Just Tell Me, Ask Me: AI Systems that Intelligently Frame Explanations as Questions Improve Human Logical Discernment Accuracy over Causal AI explanations," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), Association for Computing Machinery, 2023.
[45] A. L. Cox, S. J. Gould, M. E. Cecchinato, I. Iacovides, and I. Renfree, "Design Frictions for Mindful Interactions: The Case for Microboundaries," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, (New York, NY, USA), p. 1389–1397, Association for Computing Machinery, 2016.
[46] Z. Buçinca, M. B. Malaya, and K. Z. Gajos, "To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, Apr. 2021.
[47] A. Anderson, J. N. Guevara, F. Moussaoui, T. Li, M. Vorvoreanu, and M. Burnett, "Measuring User Experience Inclusivity in Human-AI Interaction via Five User Problem-Solving Styles," *ACM Trans. Interact. Intell. Syst.*, vol. 14, Sept. 2024.

APPENDIX

A. Relation of Design Features to Existing Plan Modes

Table IX summarizes how each of our prototype's design features relates to the Plan Modes of four coding agents – Claude Code [2], VSCode Agent Mode [3], Cursor [4], and Cline [5] – and the planning mode of the spreadsheet agent Shortcut AI [31]. We surveyed these tools in June 2026; as they are updated frequently, some of these features may have since changed.

B. Study Materials

We include the following study materials: the screener all prospective participants filled, along with the criteria for qualification for the study; the protocol for the user study; and the descriptions provided to participants about the tasks they worked on.

1) Screener Survey:

1. Which of the following apps do you use for professional purposes more than once a week? Select all that apply.
   - Notion (*May Select*)
   - Spreadsheet.com (*May Select*)
   - Smartsheet (*May Select*)
   - Adobe Photoshop (*May Select*)
   - Google Docs (*May Select*)
   - Gmail (*May Select*)

Design Feature	Our Prototype	Claude Code	VSCode Agent Mode	Cursor	Cline	Shortcut AI
Read-only access to codebase during planning	✓	✓	✓	✓	✓	✓
User-controlled explicit mode switching	✓	✓	✓	✓	✓	✓
Asks clarifying questions before planning	✓	✓	✓	✓	✓	✓
Clarifying-question modality[b]	NL	MCQ	NL	MCQ	NL	NL
Dynamic plan-status updates during execution	✓	✓	✓	✓	✓	✓
Persistent visible plan artifact	✓	✗	✓	✓	✗	✗
Direct manipulation of plan steps	✓	✗	✗	✗	✗	✗
Partial execution of the plan	✓	✗	✗	✗	✗	✗

TABLE IX: Relation between the design features of our prototype (Section III-C) and the Plan Modes of existing tools, as surveyed in June 2026. ✓ indicates the tool supports the feature; ✗ indicates it does not. [b]Modality of the clarifying-question interaction: NL = natural language in chat; MCQ = multiple-choice questions through UI cards.

- Microsoft Word (*May Select*)
- Microsoft Excel (*Qualify*)
- Google Slides (*May Select*)
- Microsoft Outlook (*May Select*)
- Canva (*May Select*)
- Google Sheets (*May Select*)
- Microsoft PowerPoint (*May Select*)
- None on this list (*Disqualify*)

2. How do you most frequently access productivity applications like Word, Excel, or PowerPoint?
   - Smartphone app (*Disqualify*)
   - Internet browser (*Disqualify*)
   - Desktop or laptop application (*Qualify*)
   - Tablet (Android/iPad) app (*Disqualify*)
3. How frequently do you use spreadsheets?
   - Once a month (*Disqualify*)
   - Once a week (*Qualify*)
   - Multiple times a week (*Qualify*)
   - Never (*Disqualify*)
4. What do you primarily use spreadsheets for? Please select all that apply.
   - Viewing information and charts created by others (*May Select*)
   - Creating new spreadsheets with structured/unstructured data (*Qualify*)
   - Editing existing spreadsheets through data entry/visualization (*Qualify*)
   - Performing in-depth data modelling/analysis (*Qualify*)
   - None of the above (*Disqualify*)
5. What is your expertise level with Excel?
   - Beginner (Basic data wrangling, sheet formatting) (*Disqualify*)
   - Intermediate (Formulas like SUM, COUNT, MAX, charts, PivotTables) (*Qualify*)
   - Advanced (Power Query, Data Model, formulas like VLOOKUP) (*Qualify*)
   - Expert (Data Model, indirect references, Python formulas, formulas like LET and LAMBDA) (*Qualify*)
6. Have you used/heard of VLOOKUP in Excel before?
   - I have used VLOOKUP before (*Qualify*)
   - I have not used VLOOKUP before, but I have heard of it and know what it does (*Qualify*)
   - I have not used VLOOKUP before and I do not know what it does, but I have heard of it (*Disqualify*)
   - I have not used or heard of VLOOKUP before (*Disqualify*)
7. Which of the following would you use to describe yourself? Choose the answer that best fits your scenario.
   - A college, graduate, university, or a post-graduate student (*Disqualify*)
   - Employee or Owner of a business with less than 50 employees (*Qualify*)
   - Employee or Owners of a business with 50 or more employees (*Qualify*)
   - Freelance Worker, Gig Worker or Volunteer Worker (*Qualify*)
   - None of the above (*Disqualify*)
8. What is your experience with using AI?
   - Extensive – I use AI-powered tools frequently for work and/or personal tasks (*Qualify*)
   - Occasional – I have used AI tools but only in limited scenarios (*Qualify*)
   - Rare – I have only tried AI tools a few times (*Qualify*)
   - None – I have never used AI for either personal or work tasks (*Disqualify*)

2) Study Protocol: Sessions followed one of two orderings, *Plan Mode First* or *Act Mode First*, counterbalanced across participants. Both followed the same structure: an introduction, a warm-up/tutorial task with the first tool, two 15-minute study tasks (one per tool), and a post-study questionnaire. Here, we provide the full protocol for the protocol in which participants used Plan Mode first.

Introduction

Hi, good morning/afternoon/evening, how are you doing? Thanks for taking time out to participate in this user study.

Today, you will be using two versions of an AI tool in Excel to work on 2 different spreadsheet tasks by remotely controlling my screen. For each task, try to make as much progress as possible within 15 minutes, after which we will

move on to the next task. Before we start, I'll give a short tutorial on how to use the AI tool.

The session will be around 45 minutes long, and you will fill out a post-session survey at the end.

Do you have any questions?

Great, let me share my screen/set up quick assist. (Make sure to share screen either way!)

Warm-up Task

Let's go through a quick demo task so you are familiar with Plan Mode, the first tool you'll be using today.

This AI tool is a chatbot, similar to something like ChatGPT, except that it has the capability to make complex edits directly to your Excel workbook. Plan mode allows you and the AI to create a plan without making any changes to the workbook before it actually makes any changes. For example, let us ask to create a spreadsheet for reporting grades of students in a class.

"I want to create a gradebook for my class of grade 10 math students"

The agent first asks some questions, let's say these are our preferences.

"numerical grading, 30 students, only 2 exams, rest up to you"

Based on this, it will build the plan. Now, you can see the plan here. You can directly edit the plan, or revise the plan with the AI through the chat. Once you are satisfied, you can execute all the steps or execute up to any step you want. Pressing execute will leave plan mode and go to act mode, where the AI can make edits to the spreadsheet. You can see the internal reasoning of the process followed by the agent here. You can of course continue the conversation to make any edits.

You can switch back and forth from plan mode whenever you wish, but please try to use plan mode at the beginning, it will be on by default.

Do you have any questions?

For the first task, TASK DESCRIPTION 1 (Section B3).

Also, please try to think aloud as you work on the task. You can go ahead and start!

Participant completes Task 1 (15 minutes)

Great! Now, let's move on the second part of the study. For this part, you will be using the Act Mode. In this mode, the AI can start making changes immediately, and will not try to create a plan. So, for example, if we say

"Create a workbook to report the sales of my furniture company"

The AI will start to do so immediately.

Do you have any questions?

So for this task, TASK DESCRIPTION 2 (Section B3).

Again, please try to think aloud as you work through the task.

Participant completes Task 2 (15 minutes)

Post-Task

Thanks for participating! The session was very insightful. Now I'm going to open the post-study questionnaire on my screen, please go ahead and fill that out. Make sure to answer the questions based on your experiences with the tool in general, not about the specific task you worked on with the tool, and please talk me through your answers.

Participant fills the post-study questionnaire.

Great! Before we end, do you have any questions?

Okay, then once again, thanks for participating! Have a nice day!

3) Task Descriptions: **Creation Tasks**

Personal Budget: Try and create your own personal budget, based on your actual needs and preferences. Your objective is to create a useful spreadsheet that you could use practically to track your finances. We will send you the spreadsheet after the session for you to use if you want to.

Personal Schedule: Create a personal schedule for yourself; your objective being to manage your time more efficiently. Again, we will send you the spreadsheet after the session for you to use.

Analysis Tasks

Movie: I'm providing you with a movie dataset, and your task is to find the next movie for you to watch based on your own preferences. Let me provide you with a brief summary of the dataset – it contains information about the genres, languages, runtimes and release dates of the movies along with keywords, taglines, and an overview of the movie. It also has information about the production company, country of origin, budget, revenue, and lifetime popularity of the movie. It also has some audience ratings, with an average rating and number of ratings.

Vacation: I'm providing you with a dataset of vacation destinations and hotels, and your objective is to plan your next vacation. Again, I'll give a quick summary of the dataset – it contains information about the specific hotel, such as the type of accommodation, the stars, the rating, number of ratings, distance from city centre, and price. It also contains information about the place itself, such as the location, a short description of the attractions, the average monthly temperature, general budget level, and ratings out of 5 for different aspects of the city such as culture, adventure, nightlife, beaches, etc.

C. Feature Codebook

Figure X describes the codebook we used to code features across the Creation tasks in our study.

D. Modifications to Post-Study Questionnaire

Questions related to 'Immersiveness' in the original scale (e.g., *"My attention was fully tuned to the activity, and I forgot about the system or tool that I was using."*) were less applicable to agentic tools that make complex edits than to purely co-creative tools. We therefore adapted these items to more directly measure participant attention across the tool and the task (e.g., "My attention was more focused on the tool" and "My attention was more focused on the task").

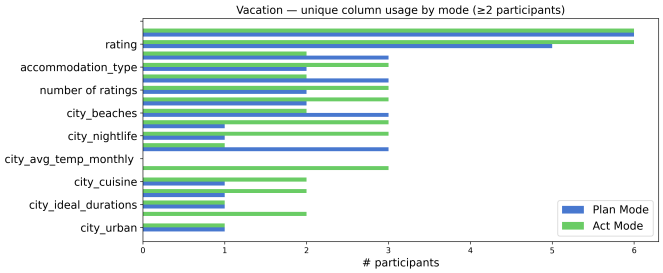
We also modified the phrasing of the 'Contribution' statement (*"I/ the system made the sketch"*) to capture creative responsibility (*"I/ the agent is creatively responsible for the spreadsheet"*) rather than literal contribution since the agent

Budget Task Codebook	
Primary Codes	
Code	**Description**
incomes	Artifacts dedicated to describing income from any source
expenses	Artifacts dedicated to describing expenses of any kind
savings	Artifacts dedicated to summarizing savings or comparing incomes and expenses
Secondary Codes	
Code	**Description**
expected vs actual	Comparing expected (budgeted) amounts vs. actual amounts
individual	Noting down individual expenses [only for expenses code]
financial goals	Describing savings goals and/or progress [only for savings code]
time-wise comparison	Comparing amounts across any quantity of time (e.g., monthly, weekly, yearly, daily)
categories	Comparing amounts across categories
specific category	Deep-dive into amounts for one specific category
taxes	Artifacts dedicated to tax calculations
balances	Artifacts dedicated to account balance calculations [only for savings code]
tips	AI-generated tips for budget management/interpretations of spreadsheet data

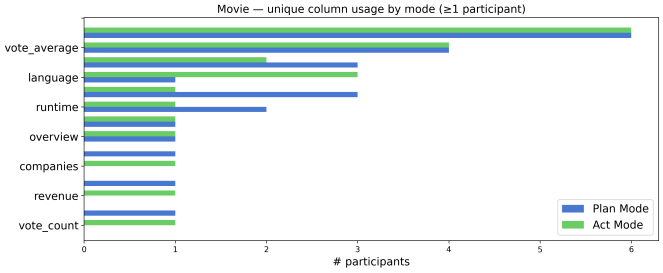
Schedule Task Codebook	
Primary Codes	
Code	**Description**
calendar	Calendar-like table describing tasks
metainfo	Information about the workbook itself such as legends
summary	Analysis of task distribution over time
task list	Table tracking all tasks, not formatted as a calendar
tips	AI-generated tips for time management
Secondary Codes	
Code	**Description**
priority	Contains information about task priorities
status	Contains information about task statuses (pending, in progress, etc.)
color coding	Color coding indicating task type/status [only for calendar code]
specific	Tracking individual, non-repeating tasks
task categories	Contains information about the categories of different tasks
instructions	Instructions on how to use the spreadsheet [only for metainfo code]

TABLE X: Hierarchical feature codebooks used in the analysis.

performed all of the spreadsheet actions in the task. To measure whether working with the agent helped participants push beyond their capabilities to produce tables they could not create by themselves, we added an extra question to capture participant perceptions on their *Capability* with respect to the final output; this dimension was not originally captured by the questionnaire as it focuses on purely creative tasks. These edits are not meant to support new statistical claims, but



(a) Columns used by participants in the Vacation task



(b) Columns used by participants in the Movie task

Fig. 6: Columns used by participants in the Analysis tasks across modes

	Analysis	Create
Plan Mode	8.6	9.8
Act Mode	7.4	7.8

TABLE XI: Mean number of requirements by mode and task type.

rather to contextualise our qualitative findings on participants' interaction patterns across modes.

E. Additional Figures

In this section, we include some figures and tables we did not have space for in the main draft.

Figure 6 shows, for each of the two **Analysis** tasks, how many participants referenced each information column of the provided dataset while working in Plan Mode versus Act Mode, and table XIV shows the average number of these columns participants used for filtering, sorting, and both across modes and tasks. Figure 7 and Figure 8 show participants' post-study questionnaire preferences for Plan Mode versus Act Mode compared across the complexity of the tasks they worked on and the upfront requirements the provided for their tasks, respectively.

Table XI shows the mean number of requirements participants expressed per task, broken down by mode (Plan Mode vs. Act Mode) and task type (**Analysis** vs. **Creation**), while Table XII provides the same breakdown at the level of each of the four individual tasks (**Budget** , **Schedule** , **Vacation** , **Movie**).

Table XIII shows the origin of participants' follow-up requirements originated across modes.

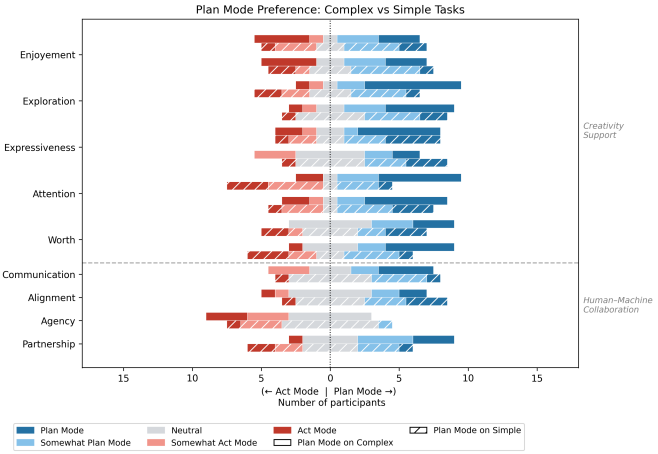


Fig. 7: Participant preferences across task complexity.

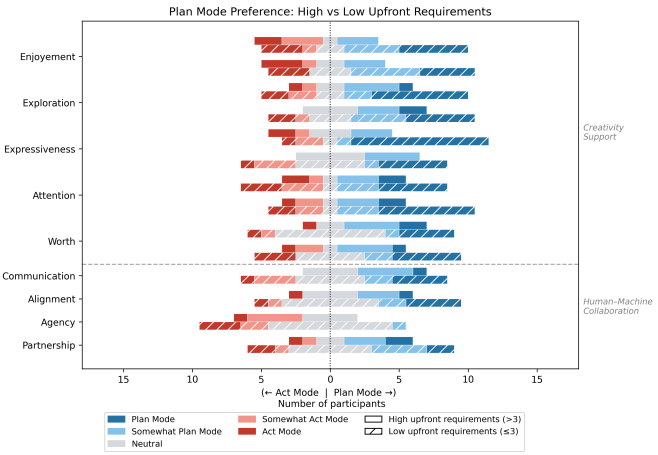


Fig. 8: Participant preferences based on upfront requirements.

	Budget	Schedule	Vacation	Movie
Plan Mode	10.7	9.0	10.2	7.0
Act Mode	9.3	6.2	9.2	5.7

TABLE XII: Mean number of requirements by mode and task.

	Proactive	Clarification	Plan	Fix
Plan Mode	6% (3)	82% (40)	6% (3)	6% (3)
Act Mode	12% (2)	12% (2)	0% (0)	75% (12)

TABLE XIII: Distribution of follow-up requirements by origin within each mode. Percentages are shown with counts in parentheses.

	Both Tasks	Vacation	Movie
Plan Mode	4.3	5.2	3.3
Act Mode	4.1	5.3	2.8

(a) Filtering

	Both Tasks	Vacation	Movie
Plan Mode	2.3	3.5	1.0
Act Mode	2.6	3.8	1.3

(b) Sorting

	Both Tasks	Vacation	Movie
Plan Mode	5.3	6.5	4.0
Act Mode	5.8	8.0	3.7

(c) Filtering/Sorting

TABLE XIV: Average number unique information columns used per analysis task across modes