

# UC Irvine

## UC Irvine Electronic Theses and Dissertations

### Title

Design Workspaces: Preserving and Recovering Meaning in Groups of Sketches

### Permalink

<https://escholarship.org/uc/item/0ck094f5>

### Author

Triplett, Sara Melissa

### Publication Date

2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,  
IRVINE

Design Workspaces: Preserving and Recovering Meaning in Groups of Sketches

THESIS

submitted in partial satisfaction of the requirements  
for the degree of

MASTER OF SCIENCE

in Software Engineering

by

Sara Melissa Triplett

Thesis Committee:  
Professor André van der Hoek, Chair  
Professor James Jones  
Professor Marian Petre

2016



# Table of Contents

List of Figures .....	iv
List of Tables .....	vi
Acknowledgements.....	vii
Abstract of the Thesis .....	viii
1. Introduction.....	1
2. Background.....	6
3. Motivating Scenarios .....	9
4. The Current State of Calico .....	12
5. Approach.....	25
Organizing.....	25
Navigating.....	26
Features .....	27
Motivating Scenarios .....	34
Additional Benefits .....	41
6. Evaluation .....	43
Methods.....	43
Findings.....	45
Discussion.....	53
Threats to Validity .....	54
7. Related Work .....	56
8. Conclusions and Future Work.....	58
Bibliography .....	61
Appendix A: Study Tasks .....	63
Appendix B: Cheat Sheet.....	65

Appendix C: Structured Task Rubric..... 68

Appendix D: Interview Questions..... 69

## List of Figures

Figure 1. Calico's original grid view.....	4
Figure 2. A sample cluster. ....	4
Figure 3: Scraps can be added to a palette for future reuse. ....	13
Figure 4: Navigation buttons and a breadcrumb trail help users navigate among recent and related canvases .....	13
Figure 5: Tags define the relationship between a canvas and its parent. ....	13
Figure 6: An example cluster.....	14
Figure 7: Tagging of related canvases. ....	14
Figure 8: The top left canvas has been manually pinned.....	14
Figure 9: Janet's Project X cluster, with designs for a UI with two widgets. It is difficult to discern labels or content at this zoom level. ....	16
Figure 10: A set of related canvases that have been pinned together. ....	17
Figure 11: The spoke on the left has been cloned and pinned in a configuration suitable for a presentation .....	18
Figure 12: After inserting a new canvas as a continuation of a pinned canvas, the pinned configuration is distorted. ....	19
Figure 13: The Widget 2 spoke, showing three alternative designs.....	20
Figure 14: Project Y's "UI" cluster, in which all canvases are pinned. ....	21
Figure 15: Project Y's "Roadmap" cluster, in which all canvases are pinned.....	21
Figure 16: UI group after adding links from UI Overview to other canvases in the group .....	22
Figure 17: Tom's wall, with a cluster for the current working set and another for backlogged items. ....	23
Figure 18. Workspace interface .....	27
Figure 19. In the Tag Picker interface, the user can select one or more tags to apply to the chosen canvas. ....	29

Figure 20. One of these canvases has been marked as a "favorite" (bottom right).....	29
Figure 21. Canvases (a) and workspaces (b) each have metadata that can be edited to provide more information about them.....	30
Figure 22. Selected canvas with visible resizing slider.....	31
Figure 23. Metadata for selected canvases is visible in the details pane. ....	31
Figure 24. The user has searched for “square.” Canvases not matching this term are hidden in the left pane and grayed out on the workspace. Canvases which are already in the workspace are grayed out in the left pane, but still visible.....	32
Figure 25: The navigation pane can be resized to show canvases in a grid-like arrangement.....	33
Figure 26. Navigation pane in sketching mode.....	33
Figure 27. Users can filter canvases by workspace while in sketching mode. ....	34
Figure 28: The UI workspace, showing the progression of the UI design to its final incarnation.....	36
Figure 29: The Widget 1 workspace, showing three different perspectives on Widget 1. ....	36
Figure 30: The Widget 2 workspace, showing three alternative designs with annotations. ....	37
Figure 31: The “Final Design” workspace, showing the final overall UI and how the widgets relate to it. ....	37
Figure 32: On the “Meeting Notes” workspace, Janet has marked up one of the sketches with planned changes during the meeting with her boss. ....	38
Figure 33: Roadmap workspace, showing the next three releases.....	39
Figure 34: Workspace for the next release, showing the planned UI design, architecture, and supported use cases.....	40

## List of Tables

Table 1: Observed behaviors of designers at the whiteboard (content taken from [5]).....	12
Table 2. Design recommendations (from [5]).....	41
Table 3: Participant action coding scheme .....	44
Table 4: Number of participants matching each result code for each subtask.....	46
Table 5: Features Used in Task 5.....	48
Table 6: Features requested by one or more participants.....	52



## **Acknowledgements**

I would like to thank my advisor and committee chair, Professor van der Hoek, for his encouragement, advice, and assistance throughout my graduate career and especially as I completed my thesis. He helped me gain confidence in my decision to attend graduate school, and helped me navigate the new world of academia.

I would also like to thank my committee member Professor Jones, whose work first drew me to UCI and whose classes exposed me to training and ideas that I was able to apply immediately in my professional life.

Finally, I would like to thank my committee member Marian Petre for her work in the area of collaborative design, which I relied on in this thesis, and for her encouragement and feedback on an early version of my prototype.

## **Abstract of the Thesis**

Design Workspaces: Preserving and Recovering Meaning in Groups of Sketches

By

Sara Melissa Triplett

Master of Science in Software Engineering

University of California, Irvine, 2016

Professor André van der Hoek, Chair

This thesis presents a prototype sketching tool designed to address the problem of reusing sketches. Our goal is to help designers preserve, understand, and communicate meaning that is represented in groups of sketches. To this end, we introduce workspaces, which provide an interface for arranging and annotating groups of sketches in flexible configurations. We also provide new sketch metadata, a navigation interface, and filtering capabilities to assist designers in locating sketches of interest.

To evaluate this prototype, we conducted a small user study. Eight participants working in development roles at a local business were recruited for this study. Participants worked individually through a set of tasks using our prototype, while we observed whether and how they used the tool's various features. We interviewed each participant upon completion of the tasks to gather more data about their experience.

Based on our observations of the participants as well as their feedback, we determined several important findings. We believe that workspaces represent an important step toward helping designers visualize, understand, and explain groups of sketches and the design stories they embody, and that the prototype succeeds in making it easier to locate sketches of interest. Additionally, we identified several feature enhancements that would make the tool even more useful.

## 1. Introduction

Sketching is a quintessential activity in any design process, and software design is no exception. Designers are known to spend significant time in front of a whiteboard, working through design problems both alone and in groups [1]. Creating sketches helps them to understand the problem at hand, visualize new designs or design changes, and communicate to stakeholders and each other [1].

With sketching playing such a key role in design, much research in this area focuses on supporting the behaviors and needs of designers during these sketching sessions. However, the utility of sketches does not end when the design session does. It has been observed that designers regularly archive their sketches for later use, whether automatically when using an electronic sketching tool or manually by using a camera to save pictures of whiteboard sketches [1, 2]. Not only do the artifacts generated during a design session provide starting material for future sessions, but they also help designers to understand their work and explain it to others long after the original session ends [2]. While existing work has identified and attempted to support some of these behaviors, the focus has been on activities within design sessions, not between them [3, 4, 5]. Although digital sketching tools such as electronic whiteboard software or even Photoshop provide native support for saving sketches, support does not typically extend to organizing the saved files in any non-trivial way.

Although designers regularly archive sketches, this does not guarantee that they will be able to use the sketches later. Indeed, there are a number of challenges that designers encounter when attempting to make use of existing sketches, which are not explicitly addressed by existing tools. First, for several reasons, archived sketches may not be as useful as they could be. Designers may not be able to discern the meaning of a given sketch simply from its content, weeks or even days after its creation. They tend to use informal and ad-hoc notation which may be inconsistent even from sketch to sketch within a session [1, 6]. Sometimes, they do not annotate at all, and are left with sketches that consist only of unlabeled boxes and arrows [1]. This only adds to the difficulty in making sense of old sketches once a design session is over.

Furthermore, the reasoning activities that accompany the creation of sketches are often lost when the session ends, even if every sketch is saved. Each sketch represents some information, but groups of sketches and the relationships among them tell a wider “design story,” which may not be evident from the sketches alone. Different views of the same sketches may also provide different types of information, which only become clear when the sketches are viewed as a group. For example, a designer might want to view an architecture diagram of a complete software system juxtaposed with detailed diagrams of important components. Then, the designer might want to juxtapose one of those component diagrams with a list of requirements and an alternative design that was discussed. Without some way to preserve these groupings and juxtapositions, the utility of saved sketches is diminished.

Additionally, designers may not be able to easily view past sketches while working on a new one. In the traditional case, limited whiteboard space means that designers must erase part or all of the board in order to start a new sketch. Designers juxtapose sketches in progress, not just completed sketches. For example, they might want to have one sketch of a rough design plan visible while fleshing out details of various parts in new sketches. Without some means of viewing multiple sketches simultaneously at a useful level of detail, designers are forced to switch back and forth or redraw reference material, adding overhead to the process of sketching.

Finally, even in the most organized and well-annotated collection of sketches, finding a particular sketch of interest can become increasingly difficult as the number of sketches grows. This problem is not unique to design work, and is addressed in any modern electronic file system with features like tags and search. However, it is also important not to minimize the value of visual memory. Often, it is faster to scan through a set of images to find one by sight than it would be to try to remember various properties of the image and search by them.

Existing sketching tools, despite automatically saving sketches, do not really address these challenges. Calico [7] is one such tool, which was developed specifically to support software designers at the whiteboard. It targets fourteen observed design behaviors relating to sketching, navigating among

sketches, and collaborating on sketches, aiming to augment the whiteboard experience without interfering with it [5].

Currently, Calico provides a three-level hierarchy for organizing and navigating sketches: canvases (sketches), clusters (groups of visually and logically related sketches), and the wall (a view of all clusters). Additionally, a breadcrumb trail orients the designer within the active cluster, allowing easy switching among canvases in the same cluster. However, these features were developed with the active designer in mind, and are less suited for the types of review activities that we explore in this thesis. Clusters are mutually exclusive, and provide only limited support for juxtaposition. Furthermore, users have found the wall and cluster views to be of limited utility once more than a few canvases are created [5]. Calico also does not support viewing multiple sketches while editing a sketch.

In this thesis, we present a reimagining of Calico which is intended to address these roadblocks in making use of archived sketches. In order to explicitly support behaviors related to revisiting sketches, we propose several new enhancements to Calico. Our goal is to help designers organize, identify, locate, and view sketches in a way that makes them effective vehicles for communication and understanding both during and outside of the design sessions in which they were created. To better facilitate the simultaneous visualization of multiple sketches, we introduce workspaces as a replacement for the wall and cluster views. Unlike clusters, which have mutually exclusive ownership of their child canvases, our workspaces hold nonexclusive groupings of sketches. Additionally, designers can independently resize and position each sketch in a workspace, and can also annotate directly on the workspace by drawing, two features that the cluster view does not provide.

An expandable navigation pane assists with visual searching. This pane displays resizable canvas thumbnails which can be enlarged to make the content more visible. In a previous version of Calico, a grid view supported this method of searching, displaying all canvases in a grid of thumbnails large enough to discern content (Figure 1). The grid also provided a secondary benefit of visually grouping related sketches. The cluster view which replaced the grid view in the latest version does visually group related sketches, but the automatic layout makes the thumbnails too small to be recognizable (Figure 2).

With our proposed navigation interface, we aim to recapture and improve upon the useful properties of the original grid view.

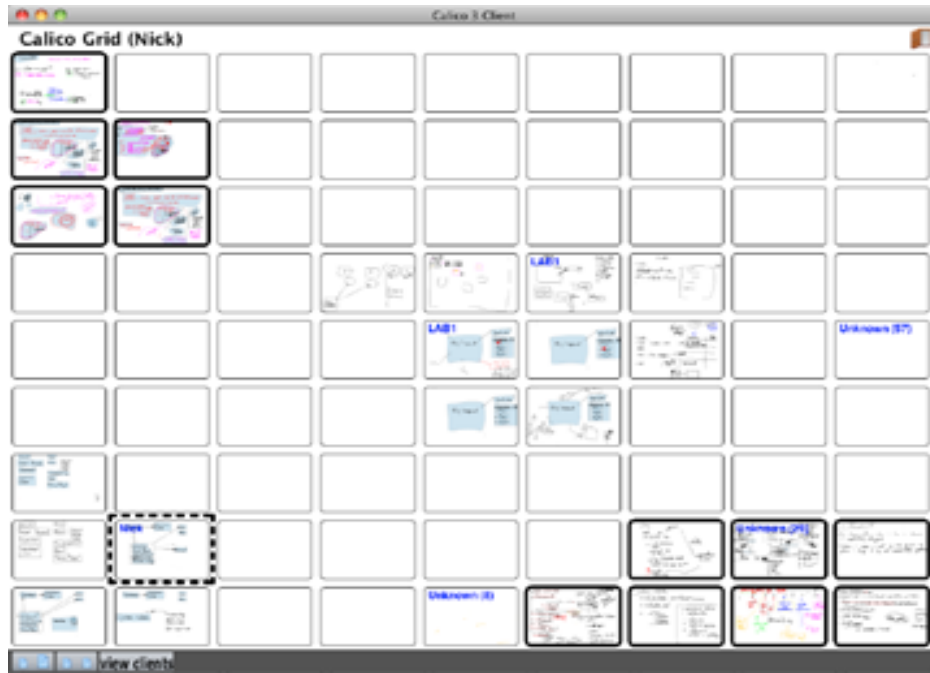


Figure 1. Calico's original grid view.

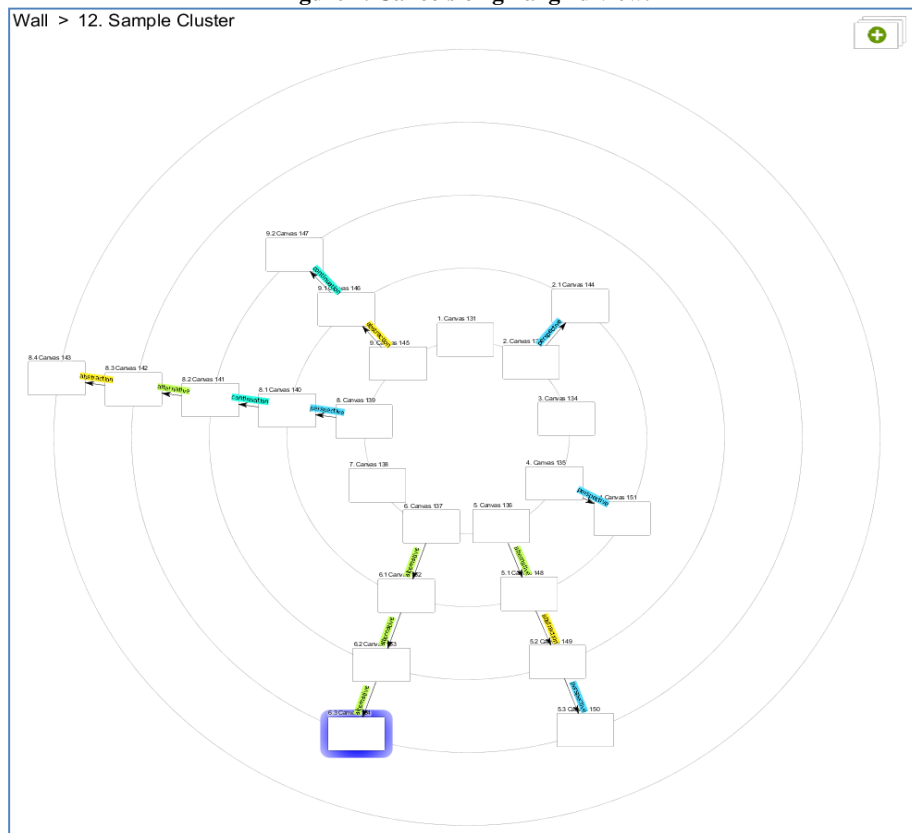


Figure 2. A sample cluster.

The navigation pane also allows juxtaposition and navigation while sketching and provides useful contextual information. This pane is visible both in workspace view and in sketching mode. While sketching, a user can see all canvases, just those canvases in the active workspace, or a subset of canvases based on search terms or metadata filters.

To help designers more effectively identify sketches, we provide lightweight mechanisms for adding identifying metadata, in the form of tags and notes. Users can now add as many tags to a sketch as they want, rather than being limited to one per sketch as in the current version of Calico. These tags are easy to add and remove, and are completely optional. Canvas notes are a brand new feature which optionally allows designers to attach open-ended text, such as a description, to any sketch. Additionally, a new “favorites” feature allows any sketch to be marked as a favorite.

Finally, we incorporate familiar mechanisms of filtering and sorting on sketch metadata, to enable users to quickly identify and locate a target sketch. These features are available through the navigation pane in both workspace view and sketching mode. Users can sort by date created, date modified, canvas name, and creator. Canvases can be filtered on one or more tags or creators, or filtered to show only favorites. Without these features, our new tagging and notes metadata would be of limited use.

In the next section, we review existing research in the field to provide background on why and how designers sketch, how they use these sketches, and what tools exist to help them in these tasks. Section 3 presents three hypothetical scenarios which motivate the design of our new prototype. Section 4 describes how Calico benefits the design teams in the motivating scenarios, as well as identifying areas of weakness or limited support. Section 5 discusses our approach to solving the problem, including a review of our new features and an analysis of how they benefit the teams in our motivating scenarios. Section 6 presents an evaluation of our prototype based on a small user study. Section 7 briefly reviews existing tools for design and sketching support that are most relevant to our motivating scenarios. Finally, section 8 reviews our conclusions and plans for future work based on our findings.

## 2. Background

The use of sketches in software design activities has been well-studied, particularly with respect to the use of whiteboards during design sessions [1, 2, 3, 5, 6, 7, 8, 9, 10]. Sketches serve a variety of functions, both during and after design sessions.

Sketching is ubiquitous in many design fields for several reasons. First, it provides a means of externalizing thoughts so that they can be more easily discussed and understood [9, 11]. It also allows designers to move fluidly from abstract to concrete as they explore and weigh different ideas [12]. Additionally, sketching allows designers to explore ideas without making decisions too early, preventing unnecessary formality from subconsciously undermining future changes [13].

Sketching plays an important role in software design as well. Petre observed that, as in other fields, sketches facilitate design by helping coordinate a design dialog across the team, capturing early ideas in a provisional way, and supporting easy juxtaposition and annotation [9]. Baltes and Diehl found that a majority of developers who created sketches do so for purposes relating to design, explaining, or understanding [2]. Similarly, Cherubini et al. identified nine scenarios in which developers use sketches, which they categorized according to three motivations: to understand, to design, and to communicate [1].

Not only is sketching an important tool during design activities, but the sketches produced during a design session are valuable even after the session has ended. Baltes and Diehl observed that the majority of sketches in their study were archived, often because sketches document important parts of the system and aid in its understanding [2]. Despite the usefulness of sketches, however, they are not always archived for later use. In some cases, this is because sketches serve as a medium for discussion and are no longer needed after the fact, but in other cases, they are replaced by redrawn sketches or simply unable to be saved due to a technical barrier [2]. Cherubini et al. also noted the level of effort required to transform a sufficiently important sketch to a more permanent form [1]. They found that these rendered drawings were particularly prominent in design reviews or when communicating with secondary stakeholders.



Over the last two decades, several tools have been developed which are meant to help designers create and use sketches. Many of these incorporate elements of sketch recognition, aiming to assist designers in translating their freeform sketches into formal models. Early work in this area includes SILK [8] and Knight [14], tools which support both freehand sketching and automatic sketch recognition, as well as transformation to more formal notations. Tahuti also allows users to sketch freehand and automatically captures emergent notations that could be subsequently modified and reused [15]. Flatland [16] explicitly supports spatial context and visual recognition as a means of retrieving sketches, automatically saving and recognizing “segments” within larger sketches. More recently, Bellamy et al. prototyped a tool which provides element recognition, automatic storyboarding, and automatic decomposition through pattern recognition [10].

Other tools are focused on capturing design rationale during a design session. The Design Meeting Agent aims to do this through audio, video, and screen capture during design sessions [17]. The “Augmented Interaction Room” proposed by Kleffman et al. is a specially-designed room with custom hardware that records designers at work and displays multiple perspectives of an active design at once [4]. DesignMinders is a proposed companion to Calico which would allow designers to record design knowledge and attach it to sketches [18]. During design activities, this tool would passively supply relevant design knowledge based on recognized keywords.

Several existing sketch support tools are intended to support designers in working with multiple sketches at once and understanding their history and relationships to one another. Ideavis [19] consists of a digital paper interface and large display screen which acts as a bulletin board to display digital versions of sketches. It automatically tracks design history, including forking, and visualizes it in a graph. The Interactive Mural [20] integrates physical sketches with a large wall display, which automatically resizes sheets as they are moved in and out of focus. Logging allows replaying the sequence of actions forward or backward. The Designer’s Outpost [21] provides a large wall interface on which post-its can be arranged and annotated to compose a web of relationships between smaller sketches.

In this thesis, we build upon previous research in the area of tool support for sketching with proposed enhancements to one such tool, Calico [7]. Calico is a software design tool which provides support for several behaviors that designers naturally engage in at the whiteboard. Its goal is to enable and augment these behaviors, without unnecessarily burdening the designer.

Calico's features have been designed to explicitly support behaviors *during* sketching. First, scraps allow designers to use low-detail models while easily capturing emergent, informal notation if and when the designer is ready to do so. Second, a three-level hierarchy, consisting of named canvases, clusters, and the wall, organizes sketches in a way that allows users to view and navigate among related sketches. Third, canvas tags enable the designer to explicitly define relationships between canvases. Finally, a navigation history stack and breadcrumb trail also help users shift focus more effectively.

While many of these features are also useful in returning to existing sketches, the behaviors of interest are viewed from the perspective of active design. As a result, the focus is generally on helping designers create and work with changing content, and there is less explicit support for managing and reviewing existing content.

To summarize, research has consistently identified design, understanding, and communication as the areas in which sketches serve an important role. Most tool support to date has focused on the design component of this trio. Where understanding and communication are explicitly supported, it tends to be in the context of an active design session. In this thesis, we focus on how to support the use of existing sketches in understanding and communicating.

### **3. Motivating Scenarios**

While existing tools provide strong support for sketching during a design session, there has been less focus on how to deal with sketches after the fact. In this section, we present three scenarios which motivate our research into solutions for storing, retrieving, and reasoning about existing sketches.

#### **Scenario 1**

Janet is the senior member of a small design team which has met several times to discuss an overhaul of Product X. Her boss has just scheduled a meeting where he wants her to summarize the team's thoughts and decisions. During each design session, her team spent a lot of time at the whiteboard, hashing out their ideas. The team was diligent about taking pictures of each sketch with their phones, anticipating the need to review them later. Unfortunately, they now have a collection of dozens of sketches, most of which are poorly labeled. Some sketches are several weeks old, and no one remembers exactly what they depict.

Janet successfully prepares for the meeting by creating a set of PowerPoint slides that include several pictures of her team's sketches. She has grouped them in useful ways and added various arrows and boxes to the slides to show how sketches relate. Unfortunately, during the meeting, Janet's boss becomes very interested in one aspect of the design that is not illustrated in her presentation. She has all of the other sketches on her tablet, but it takes time to find the ones that are relevant to her boss's questions.

At the end of the meeting, Janet's boss asks that she make some revisions to her team's final designs. He wants a few changes to the existing designs, and also has a new requirement from the client that needs to be incorporated. He wants to meet again in a few days to see the changes, so Janet and her team get to work. They begin by redrawing the sketches that need to be modified. After starting to change them, the team realizes that the new client requirement means that part of their existing design will not quite work. They spend some time rearchitecting the impacted components, drawing several new sketches in the process.

When the team is finally satisfied, Janet has to redo her PowerPoint slides. The structure is largely the same, but she needs to update all of the included sketches, some of which appear on more than one slide. She also needs to add in a few new sketches.

## **Scenario 2**

Molly is meeting with Andrew, who has recently joined the Project Y team. She is bringing him up to speed on the current design, using printouts of several sketches from earlier design sessions. The design is quite complex, with various interrelated parts and important perspectives. There are several sketches relating to UI design, architecture, implementation, and use cases. Their company does frequent releases, so some of the features they are currently designing will not be implemented right away. During the meeting, they find themselves frequently juxtaposing two or more sketches to discuss some aspect of how they relate. During a conversation about the user interface, they lay out several of the UI sketches to show an overall picture of its design and components. Later, they discuss the project road map by grouping the sketches that illustrate the plans for the next two releases—including a few UI sketches, two architecture sketches, and a handful of others. They want to capture these relationships in some way for future reference, so Molly starts taking pictures of the groupings on her phone. She worries that when they review these pictures later, they will not remember exactly how or why the grouped sketches are related.

## **Scenario 3**

Tom and his team are meeting to continue a design session from two weeks earlier. Fortunately, their company has electronic whiteboards, so they are able to pull up all the drawings from the previous session and pick up where they left off. Unfortunately, in the two weeks that have passed, they were pulled in to help on an unrelated emergency, and are having a hard time remembering the project status. They know that three of the components were finalized, and that they had identified two more components as next on the list. However, there are dozens of sketches and no easy way to find the most important ones.

After a tedious review of the existing sketches, Tom's team finally determines what they are meant to work on next. Their starting point consists of two "final" designs from the last release of their software. They need to modify these designs to add new features and fix several bugs reported by users. The components requiring modifications are closely related to some other components. Tom wants to isolate his changes and avoid propagating unnecessary redesigns to other components. His team has to keep flipping back and forth between their active sketch and the sketches for the other components, to ensure that their new designs will not break anything downstream.

## 4. The Current State of Calico

Before we discuss how we address the scenarios described in Section 3, we first discuss the current incarnation of the Calico tool, examining how and to what extent its features could benefit the design teams in our motivating scenarios.

Calico is designed to support fourteen behaviors that software designers engage in at the whiteboard [5]. These behaviors can be grouped into three categories, shown in Table 1.

**Table 1: Observed behaviors of designers at the whiteboard (content taken from [5])**

<b>Category</b>	<b>Behaviors</b>
<b>How Designers Sketch</b>	<ul style="list-style-type: none"><li>• Designers draw different kinds of diagrams</li><li>• Designers draw what they need, and no more</li><li>• Designers refine and evolve sketches</li><li>• Designers use impromptu notation</li></ul>
<b>How Designers Navigate Sketches</b>	<ul style="list-style-type: none"><li>• Designers work with different perspectives</li><li>• Designers work with alternatives</li><li>• Designers work with sketches at different levels of abstraction</li><li>• Designers perform mental simulations</li><li>• Designers juxtapose sketches</li><li>• Designers review their progress</li><li>• Designers retreat to previous ideas</li></ul>
<b>How Designers Collaborate with Sketches</b>	<ul style="list-style-type: none"><li>• Designers switch between synchronous and asynchronous work</li><li>• Designers bring their work together</li><li>• Designers explain their sketches to others</li></ul>

Each of Calico’s features is intended to support one or more of these behaviors during a design session. Designers can sketch normally with an electronic pen, similar to a traditional whiteboard. Lightweight interaction mechanisms allow them to create and manipulate “scraps” by circumscribing regions of a sketch, converting these regions into grouped objects that can be moved, resized, and copied independently. Scraps can be saved to the Palette for future reuse (Figure 3). These features make it easier for designers to preserve impromptu notations, duplicate portions of a sketch for further elaboration, and make quick modifications to designs without having to redraw parts of them.

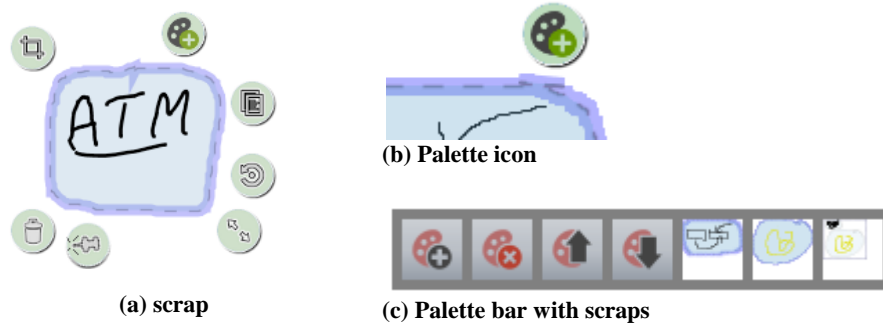
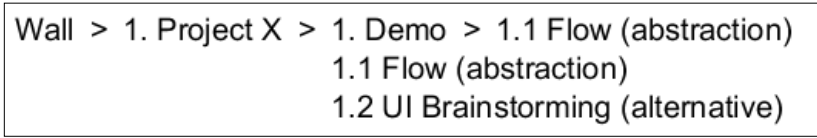


Figure 3: Scraps can be added to a palette for future reuse.

A history stack makes it easy to navigate among recent canvases, while a breadcrumb trail allows users to quickly move to related content anywhere in the hierarchy (Figure 4). Intention tags provide a history of how each new canvas relates to the previous one, so that the evolution of a design can be tracked (Figure 5).



(a) Forward/back buttons for history stack.



(b) Breadcrumb bar.

Figure 4: Navigation buttons and a breadcrumb trail help users navigate among recent and related canvases

In relation to Canvas 53, this canvas is:

- something unrelated [no tag]
- a new alternative [alternative]
- some part in detail [abstraction]
- more of the same [continuation]
- a different view [perspective]

Figure 5: Tags define the relationship between a canvas and its parent.

Canvases are automatically grouped into clusters as they are created (Figure 2), with each canvas linked to its parent, if any (Figure 6). Links are accompanied by tag labels (Figure 7), making it easier to visualize the origins and rationale for any particular canvas. Any canvas can also be moved and pinned in the cluster view, giving the user control over the layout of canvases (Figure 8). The cluster view is also zoomable, which along with pinned canvases provides support for juxtaposition and inspection of two or more canvases.

To help designers think through designs in teams and explain their sketches to others, Calico provides a vanishing highlighter that can be used to temporarily annotate canvases. Users can create new clusters and canvases and work on them asynchronously, and later invite others to collaborate on them. Finally, Calico's distributed architecture supports multiple users working together on the same canvas across multiple devices, making collaboration easier.

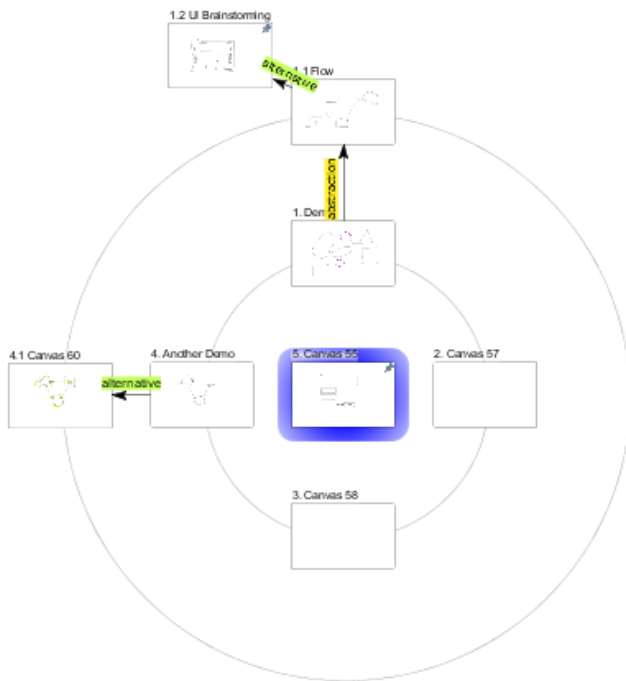


Figure 6: An example cluster.

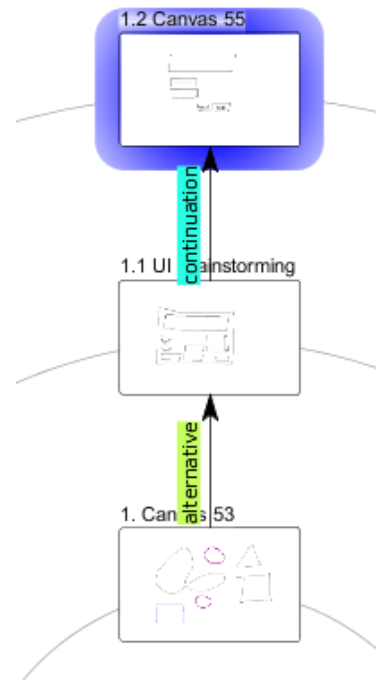


Figure 7: Tagging of related canvases.

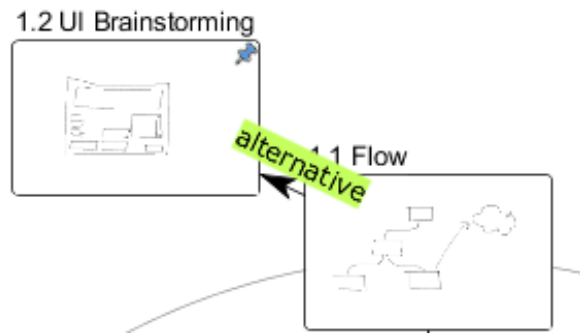


Figure 8: The top left canvas has been manually pinned.

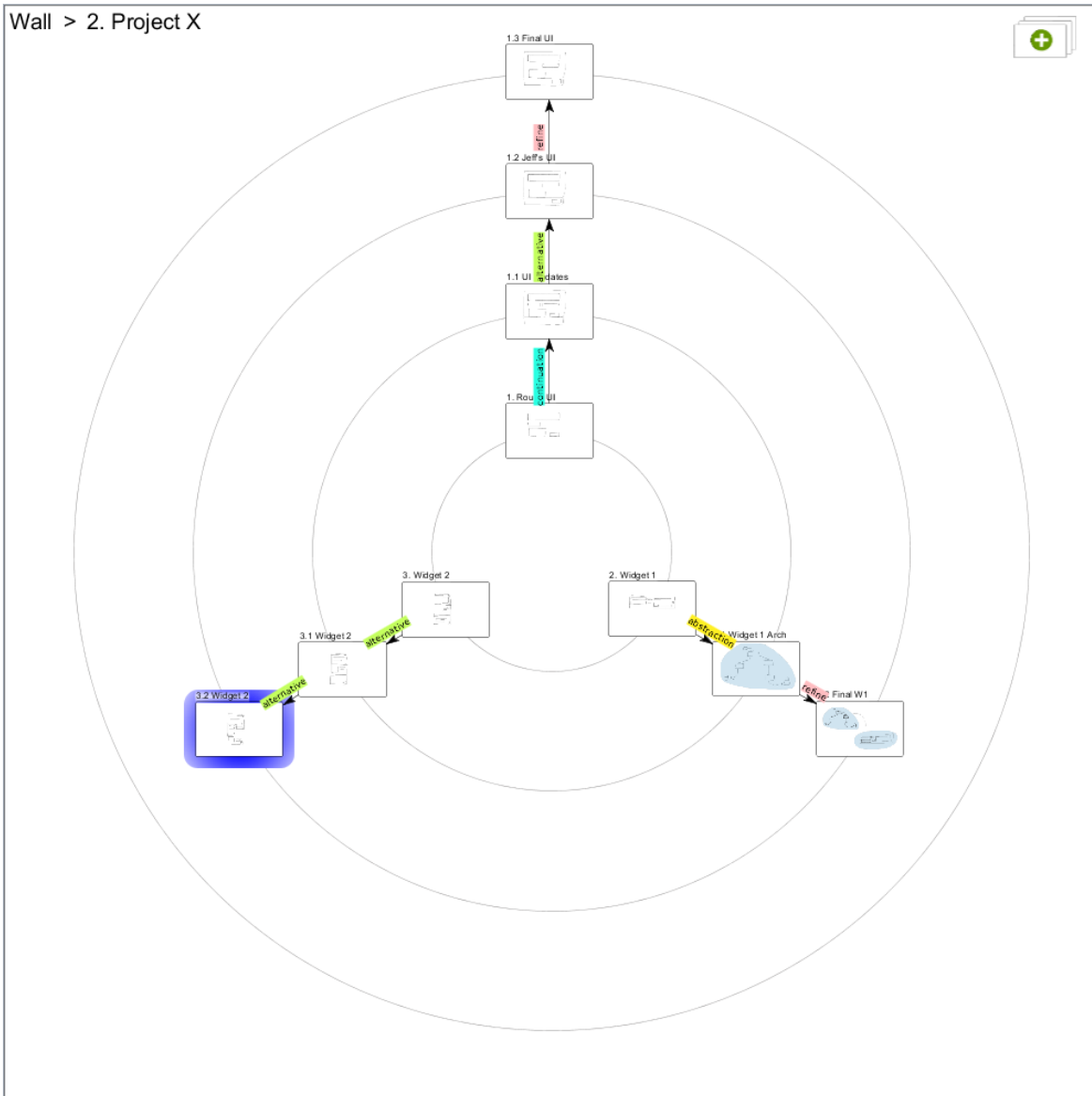


Many of these features are useful in the motivating scenarios we described above. However, some limitations exist that reveal opportunities for new features that might better support our hypothetical users in these scenarios. In the following sections, we show how Calico might be used in each scenario, and explore the factors that might limit its usefulness.

## **Scenario 1**

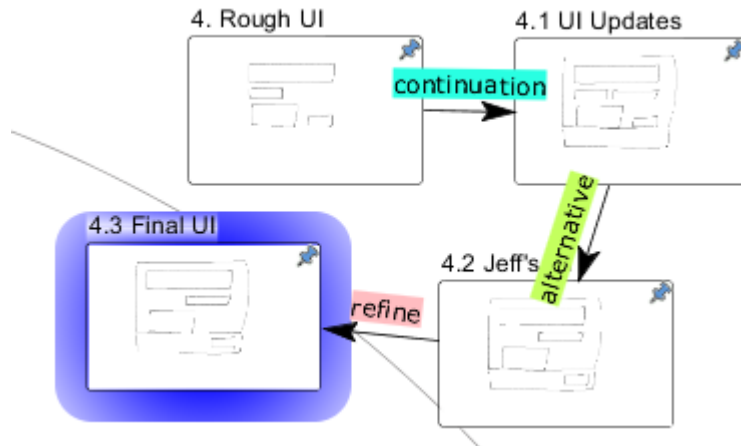
Calico provides an obvious benefit for the team in scenario 1. Rather than using a traditional whiteboard and snapping pictures of their sketches, the team can sketch on an electronic whiteboard which automatically saves their sketches. An emergent organization for these sketches is provided automatically in the form of clusters and the wall. Sketches created during a single design session are automatically part of the same cluster, with links between them preserved and tagged. Clusters and canvases also have explicit names, making it much easier for Janet to identify them after the fact. Some of the design rationale that Janet needs to provide to her boss can be captured in the tagged links between canvases.

Figure 9 shows an example of what one of the clusters from this project might look like. This cluster has three spokes: one for the user interface (top), one for Widget 1 (right), and one for Widget 2 (left). These three spokes have been laid out automatically based on the order in which canvases were created and the relationships between them. In this case, the end of the UI spoke has the final UI design and the end of the Widget 1 spoke has the final design for that widget, as they are the last designs along each chain that the team created. These final designs are also tagged as refinements. Note that, because the layout is an emergent property of the order in which the design is created, the “final” design will not necessarily always appear at the end of a spoke. However, it is not too burdensome to force this to occur by cloning the final design at the end, if necessary.



**Figure 9: Janet's Project X cluster, with designs for a UI with two widgets. It is difficult to discern labels or content at this zoom level.**

To prepare her presentation, Janet can now take advantage of this cluster both to decide what to present and to arrange content for her slides. She can quickly zero in on the “final” canvases for the UI and Widget 1 spokes. If these components have a long or storied history, she can review the tags to help decide which related canvases are also important to include. Once she has decided which sketches to include, instead of needing to paste images of each one into her slides, she can take screenshots of clusters or sets of pinned canvases (Figure 10), with relationships automatically displayed by Calico.



**Figure 10: A set of related canvases that have been pinned together.**

Calico makes it easier for Janet’s team to make the requested updates, as well. Rather than needing to redraw any sketches, they can simply modify any of the existing canvases. To incorporate the new request, they can clone the original sketches and then modify them, so that the history of changes is preserved. Furthermore, when it comes time for Janet to update her presentation, any pinned groupings she created in her cluster will still be in place, with updated content. She can simply take another screenshot to replace the old one in her slides.

While Calico’s features are a great improvement over a physical whiteboard, some aspects of Janet’s work are still tedious. Calico does not provide search capabilities, so Janet must rely on a visual scan of a cluster, or a tedious search through the breadcrumb trail, to locate a canvas by name. Similarly, once a critical mass of canvases in a cluster is reached, it becomes difficult to discern the content of any canvas, making the cluster view less useful for quickly finding a canvas by sight. This is apparent even in Figure 9, which has only a small number of canvases. While it is still easier for Janet to find a canvas in Calico than among a large set of unorganized images, locating sketches on demand may still be painstaking.

If Janet has cloned any canvases in order to pin them in useful configurations (Figure 11), she will need to update each copy if the content needs to change. Furthermore, adding new sketches causes the cluster layout to change automatically to accommodate them, sometimes in unexpected ways that disrupt

pinned canvases. Figure 12 illustrates what happens to the pinned group of canvases from Figure 11 when a new canvas is added as a continuation of the top right canvas in the group.

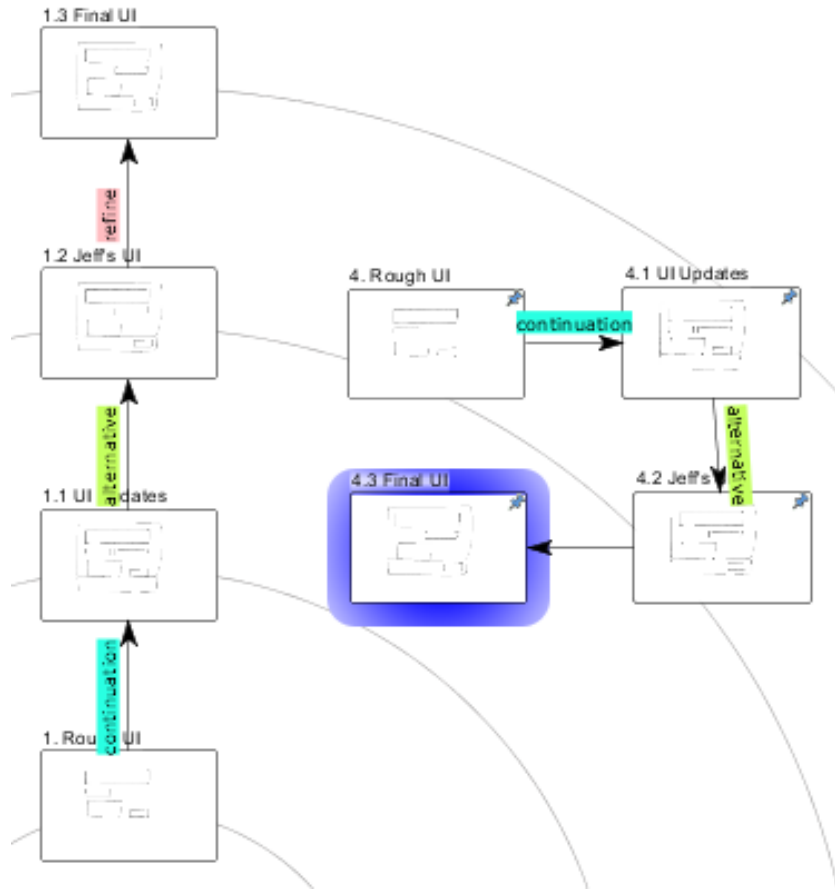


Figure 11: The spoke on the left has been cloned and pinned in a configuration suitable for a presentation

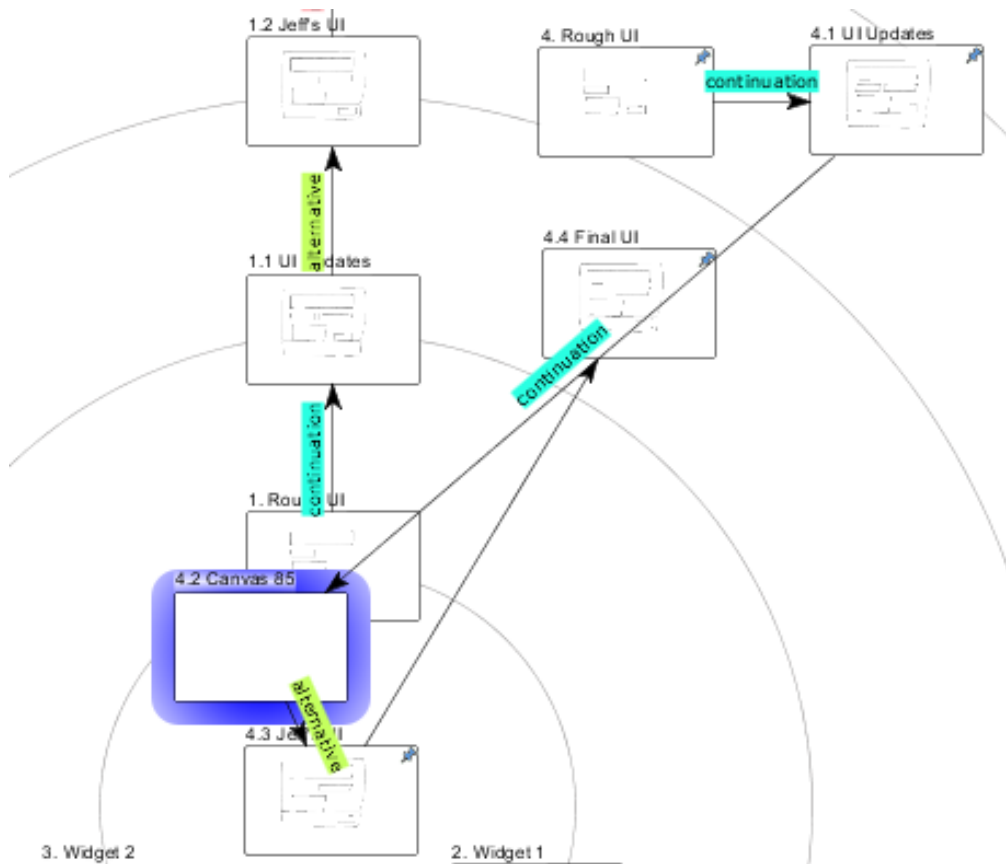


Figure 12: After inserting a new canvas as a continuation of a pinned canvas, the pinned configuration is distorted.

Because tags apply to relationships *between* canvases, they are of limited use in describing the canvases themselves. Furthermore, only one tag can be applied to any given relationship, and each canvas can have at most one incoming and one outgoing visual link, whereas a given canvas may be actually related in multiple ways to multiple canvases. For example, the Widget 2 spoke actually shows three alternative designs for the same widget (Figure 13). These alternatives are all arranged in a line, making it appear as if the outermost canvas represents the end of some train of thought. In fact, all three alternatives may be equally important at this point in the design process.

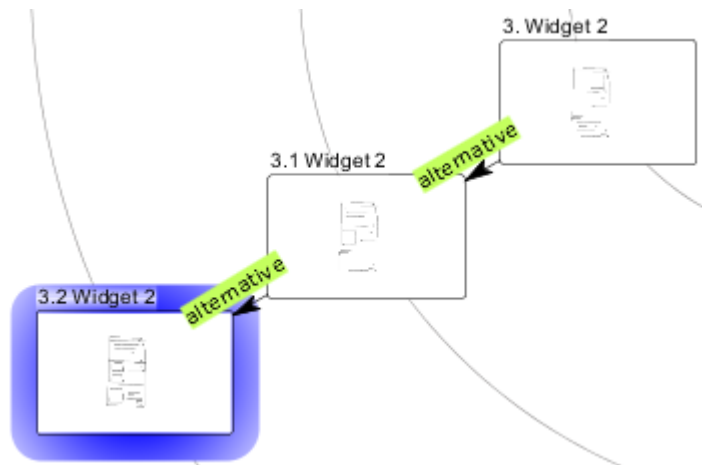


Figure 13: The Widget 2 spoke, showing three alternative designs

Finally, tags and labels are the only identifying metadata available. There is no mechanism for adding notes, descriptions, timestamps, or any other information *about* the sketch that might help Janet discern the significance of each sketch as she creates her presentation.

## Scenario 2

The team in scenario 2 would also be better off had they done their design work in Calico. As described above, related canvases are automatically grouped together in a cluster, with their relationships visualized as lines between canvases. The intentional interface allows users to apply a tag identifying how each new canvas relates to its parent. These tags can be customized by the user, to identify new canvases as abstractions, new perspectives, or detailed views, for example. In this framework, not only would Molly and Peter not have to worry about remembering how the grouped canvases relate, but the relationships could have been explicitly captured during the design session through the use of tags, eliminating the need for paper shuffling and taking pictures.

Figure 14 and Figure 15 show two clusters that Andrew and Molly might create to visualize two important perspectives of their project: the user interface overview and the road map. In both clusters, all canvases have been pinned in a custom arrangement, so the familiar circle visualization does not appear. In the UI cluster, the top central canvas illustrates a master or bird's-eye view of the whole user interface, and the other canvases include abstractions, detailed designs of smaller UI components, and use cases

(Figure 14). The road map cluster consists of several canvases which have been cloned from other canvases, illustrating the design of the system for each of the next three releases (Figure 15).

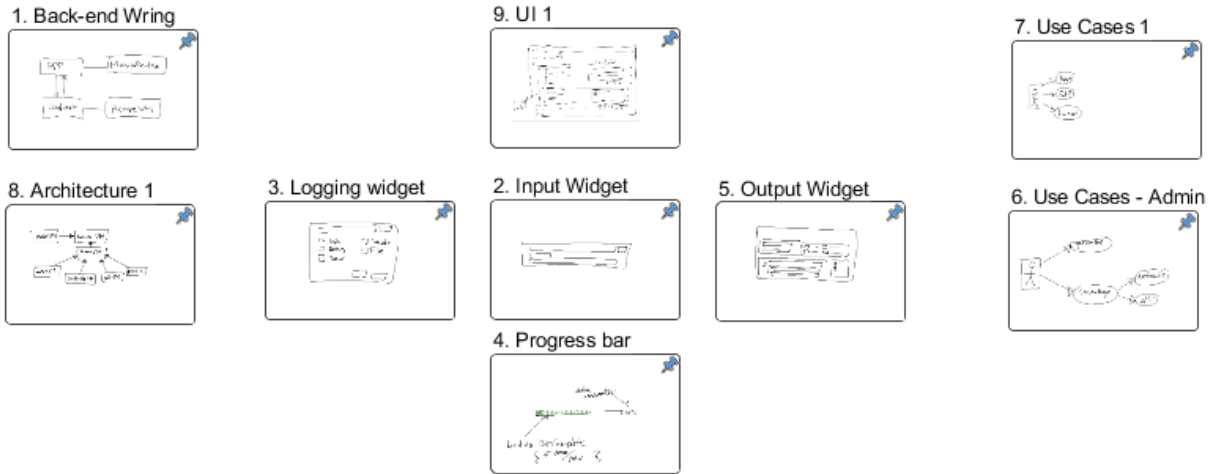


Figure 14: Project Y's “UI” cluster, in which all canvases are pinned.

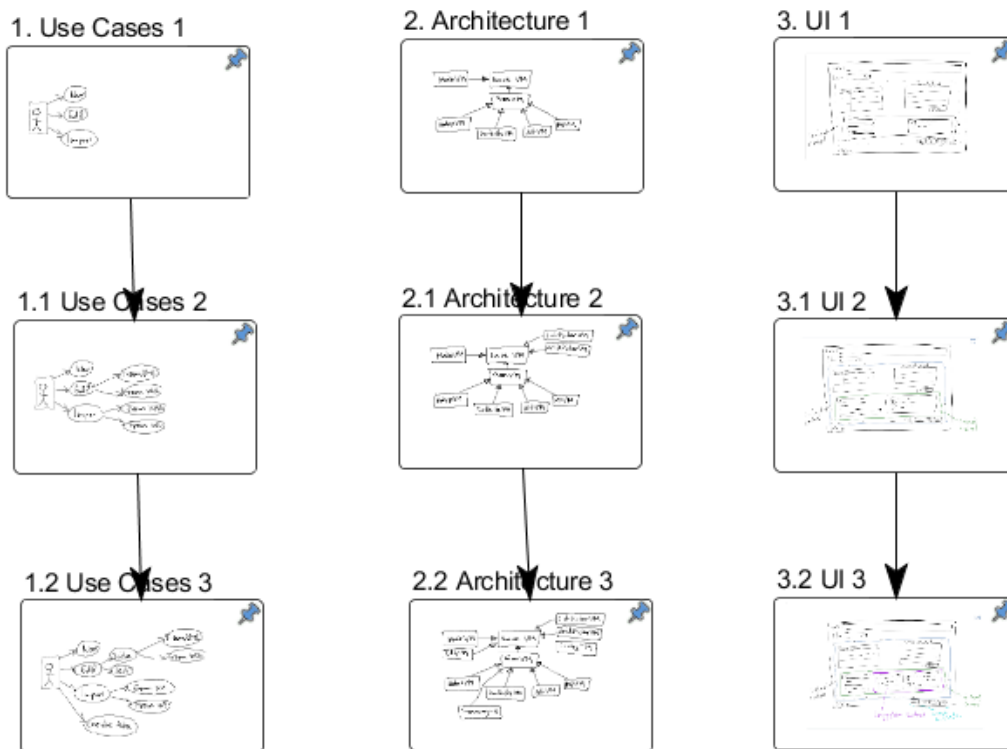
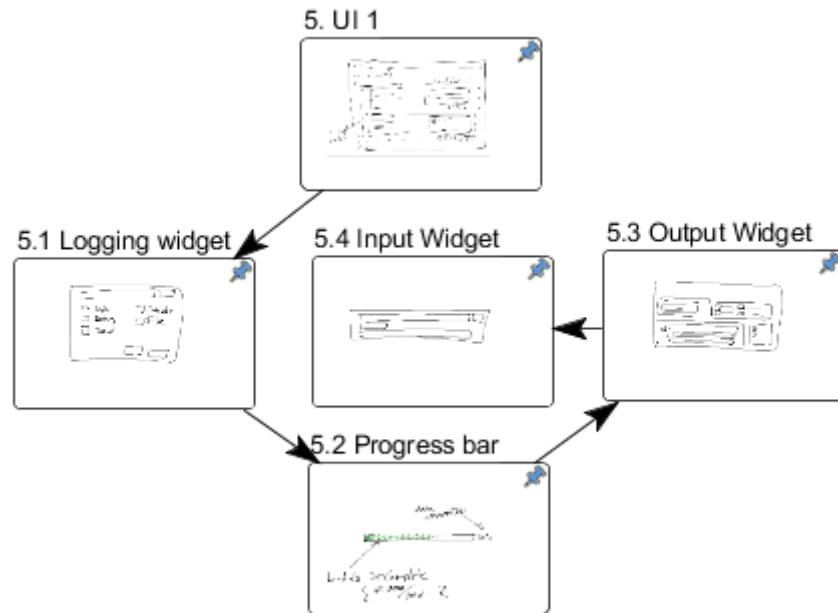


Figure 15: Project Y's “Roadmap” cluster, in which all canvases are pinned.

Unfortunately, as we saw with scenario 1, only one relationship between any two canvases can be captured in Calico's cluster framework. Consequently, tags may actually work *against* understanding if the real relationships are more complex. In the UI cluster, for example, adding explicit relationship arrows between the bird's eye UI view and detailed sketches of each component renders a linear chain of canvases (Figure 16), obscuring the true relationships among them.



**Figure 16: UI group after adding links from UI Overview to other canvases in the group**

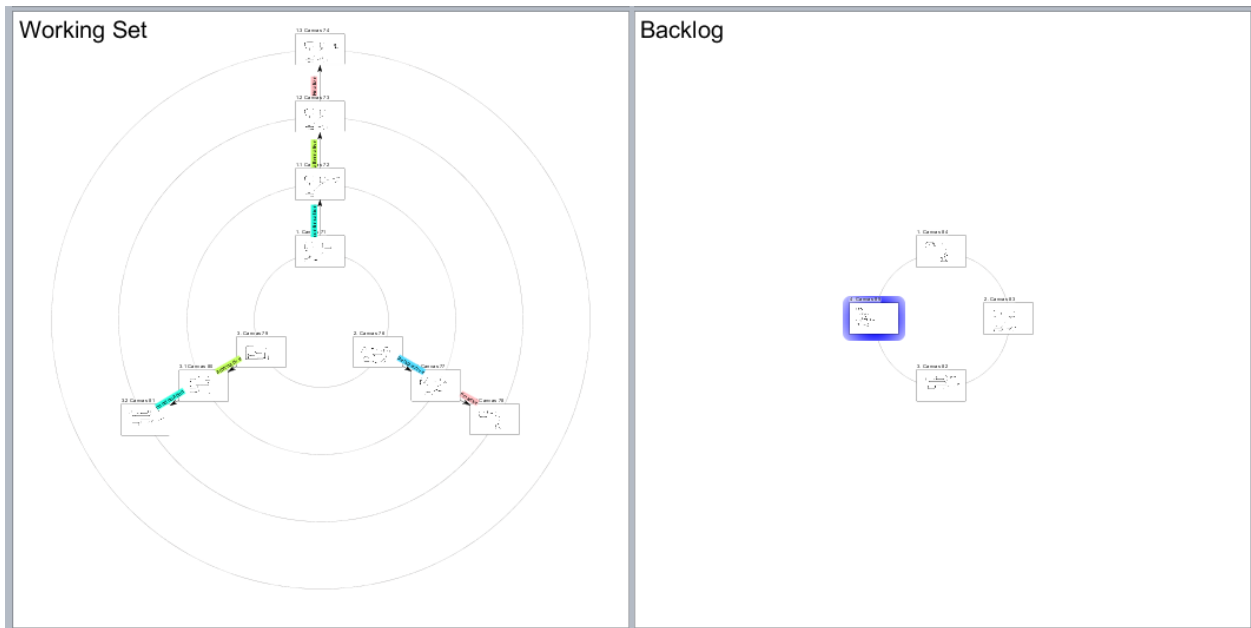
Additionally, clusters are mutually exclusive, so canvases can be grouped together in only one way at a time. In order to reuse a canvas in another group, the user must make a copy of the canvas. The example UI and Road Map clusters actually share three UI sketches (Architecture 1, UI 1, and Use Cases 1), which means that if changes are required to any of these, the changes must be made in both copies.

Furthermore, the mere physical grouping of canvases in a cluster may not be sufficient to remind the team *why* they are grouped together. Two possible interpretations of the groups in the road map cluster are: 1) each group represents a release, or 2) each group represents the road map for one feature. Being able to annotate groups or illustrate more complex relationships visually would help clarify the meaning of these groups.



### Scenario 3

In this scenario, the team is already at an advantage by using an electronic whiteboard. Calico’s intentional interface provides additional support for this scenario beyond simple sketching software. Because Calico automatically maintains canvas groupings, navigational history, and relational definitions, this team should find it easier to trace the history of their design. The most recently created canvases in each set of tagged relationships will appear at the ends of the spokes, making them easier to see. The team can define a tag for “finalize,” making it easy to see when a canvas is finalizing the design in a parent chain. This is similar to the approach we described for Scenario 1.



**Figure 17: Tom’s wall, with a cluster for the current working set and another for backlogged items.**

Alternatively, the team can use Calico’s clusters to manage a pipeline for their sketches. Figure 17 illustrates what Calico’s Wall might look like early in Tom’s project. The first cluster holds the active “working set,” a group of sketches that pertain to the team’s active tasks. The second cluster is the project backlog. This cluster contains sketches pertaining to stages of the project that need to be completed, but are not currently being worked on. As the team starts finalizing designs, they could use a third cluster to hold “final” sketches. Other clusters are also used to hold intermediate stages, project history for finalized sketches, and so on. As Tom’s team works, they simply move sketches to the

appropriate cluster. In our example scenario, the team can easily jump back into the project by checking the “Finalized” cluster to review what has been decided, and beginning work in the “Working Set” cluster.

As Tom and his team get started on their next task, they clone the final designs for the two components that need to be modified, and move these clones to the working set. They also clone the sketches for the dependent components with which they are trying to maintain compatibility, adding them to the working set for reference. This allows them to quickly switch back and forth with the breadcrumb bar (Figure 4a).

Calico’s current feature set goes a long way toward meeting Tom’s needs. However, there is still room for improvement. We have presented two ways in which Tom could have used Calico to manage his workflow. Unfortunately, neither fully supports his needs. If the team organizes clusters to keep all related sketches together, and relies on tags and the automatic cluster to identify final designs, they run into the same problems that Janet encountered in Scenario 1. The position of a canvas in a spoke is not always an accurate visualization of its status, the team cannot quickly isolate all finished designs, and canvases cannot be tagged directly. The second method is problematic as well. Moving canvases between clusters is straightforward, but this approach loses the history of each design. In effect, Tom must choose whether to organize his sketches by topic (keeping the design history together) or by status (keeping final sketches together). A better solution would support both methods of organization simultaneously.

Calico’s support for Tom’s current task is also suboptimal. Cloning existing canvases into a new cluster in order to keep them at hand is easy enough, but creates duplicate data that Tom must now be careful to keep consistent. If another task arises that involves changing one of the referenced components, Tom’s working set will not stay up to date with those changes unless he actively copies them over. Additionally, although it is easy to switch back and forth between the working sketch and reference sketches, or to view them side-by-side in the cluster, it would be better if Tom could actually see the reference sketches while editing his working sketch.

## 5. Approach

In this thesis, we seek to address the limitations of previous versions of Calico and provide broader support for working with groups of sketches through a series of new features. Our primary goal is to support designers in making sense of their sketches later. To this end, we focus on methods for organizing and navigating existing sketches. This includes mechanisms for classifying, identifying, visualizing, and grouping sketches.

### Organizing

When considering how to design a method of organizing sketches, we wanted to find a solution that helps designers use sketches to communicate and understand their ideas. Sketches serve a dual role in design: first, the act of sketching facilitates design by helping designers externalize their thoughts, and second, the sketches themselves preserve various aspects of the design. We wanted to support both of these purposes through a means of organizing sketches.

In the first instance, the reason sketches are useful is that they provide a physical representation of ideas that designers can reference during a discussion or process of thought. The designer can point to parts of a rough sketch while verbally discussing the component it represents. She can gesture to show how data flows, how parts are related, which are the important aspects, and so on. Unfortunately, this reasoning activity is not preserved in the sketches themselves. The process by which sketches were created, how they evolved from one another, why they matter, or why one design was chosen over another, are not explicitly illustrated in any particular sketch. Thus, much of the reasoning that is so important to the design process is not captured, even if all sketches are saved. Because the design process is often critical to understanding and explaining the final design, we wanted to find a way to store and visualize at least some of it in a physical representation as well.

Our second goal is closely related to the first. Sketches evolve and are replaced by new sketches, and often the importance of one sketch is implicit in how it differs from a previous sketch. For example, designers might illustrate two alternatives, or use one sketch to dive into a concrete exploration of an

abstract representation in another sketch. Additionally, while any one sketch may illustrate some important aspect of the design, it is usually many sketches, taken together, that form a complete picture. Again, the information pertaining to how sketches relate to one another is not necessarily represented simply by having all sketches. Furthermore, any particular sketch may represent a different aspect of the design when viewed alongside other sketches. Several sketches of the user interface, taken together, illustrate how that part of the system should be built. A sketch of one widget from the user interface along with an architecture diagram and a data model might illustrate how user input data flows through the system and results in some action taken by the system. Rather than forcing designers to reconstruct these complete pictures from individual sketches as needed, we wanted to find a way to save and visualize these groupings of sketches in meaningful ways.

## **Navigating**

Closely related to organization is sketch navigation. When we consider navigation, we are concerned with how to shift focus among sketches, how to find the right sketch, and how to navigate efficiently.

Designers are known to shift focus as they design, often switching back and forth repeatedly among frames of reference. In the current version of Calico, the breadcrumb trail and navigation buttons are intended to support rapid shifting among recent or related canvases. However, these options are impractical for more complex focus patterns such as shifting among three canvases, or shifting among canvases across clusters. We wanted to help users define their own contexts or frames of reference to work with, in a way that does not constrain sketches to only one frame of reference at a time.

Searching and filtering are also important components of navigation. In order to move to a sketch of interest, a designer must first locate the intended sketch. Sketching tools are capable of automatically capturing a wide variety of data about sketches that may be useful in locating a sketch. We wanted to capture useful data and allow it to be used to quickly locate sketches, either by uniquely identifying a sketch or by visually narrowing large search spaces to more manageable sets of results.

In addition to supporting search on automatically recorded metadata, we wanted to support more organic searching behaviors. Temporal [22, 23] and spatial clues [24] are known to be useful in locating information, and a designer may be more likely to remember what a sketch looks like than to remember its title. Calico’s original grid interface was well-suited to visual search and spatial memory. Both of these benefits have been effectively lost with the cluster interface: the thumbnails in clusters are too small to see, and the cluster layout changes as more canvases are added. We wanted to recapture and improve upon this valuable mechanism for locating sketches by sight.

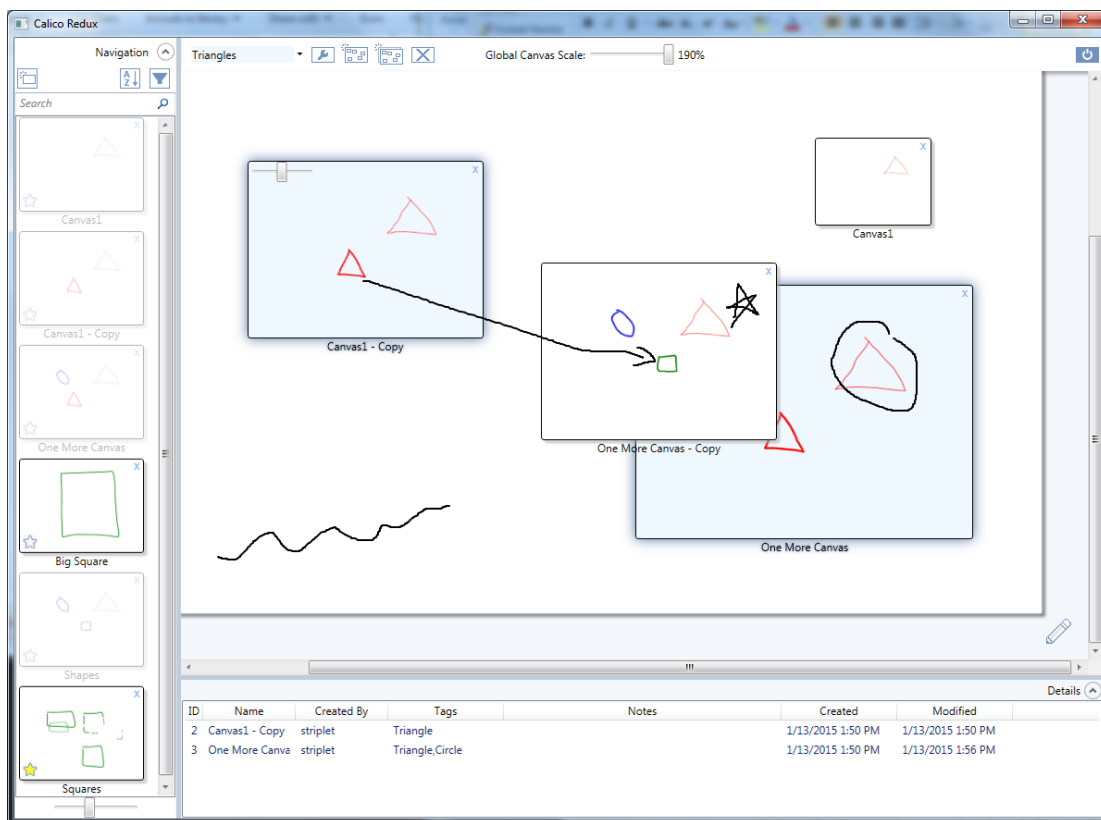


Figure 18. Workspace interface

## Features

### Workspaces

In our prototype tool, the cluster view and wall have been replaced by a new interface (Figure 18) that revolves around a new object, the workspace. The workspace is based on the idea of a bulletin board, where a designer can pin sketches and notes in any configuration. Users can create as many named workspaces as they want, and can shift among them easily with a dropdown menu. Workspaces can also

be cloned. Like clusters, workspaces can hold a number of canvases and show thumbnails of each one. However, they have several key attributes which make them extremely useful in addressing the limitations of prior versions of Calico.

### **1. Resizable canvases**

Canvases which are placed on a workspace can be independently resized using a slider that appears on a selected canvas, as in the top left canvas on the workspace in Figure 18. This allows the user to enlarge important canvases or juxtapose a rough sketch with blown-up sketches of certain components.

### **2. Workspace sketching**

The workspace itself is a simplified canvas. Users can draw directly on the workspace, on a transparent layer above the pinned canvases. They can draw freeform arrows between canvases, circle parts of sketches, or jot notes in the margins, all without affecting the content of the sketches. The thick black markings on the workspace in Figure 18 show examples of such annotations.

### **3. Shared canvases**

Unlike Calico clusters, workspaces are not mutually exclusive. Any canvas can be added to any number of workspaces. Changes made to a canvas will show up in any workspace it belongs to. Essentially, workspaces hold object references to canvases.

## **Metadata**

Our prototype provides several new types of metadata beyond what Calico previously offered. This metadata allows for a richer understanding of the origin, purpose, and content of sketches, which can be useful for anyone reviewing them. It also supports searching, sorting, and filtering.

### **1. Tags and Favorites**

Instead of using tags to describe how a canvas relates to its parent, users now create tags for canvases (Figure 19). A canvas can have any number of tags. For example, a canvas might be tagged as both “UI” and “Preliminary.” Additionally, any canvas can be marked as a “favorite” to call it out as especially important (Figure 20).

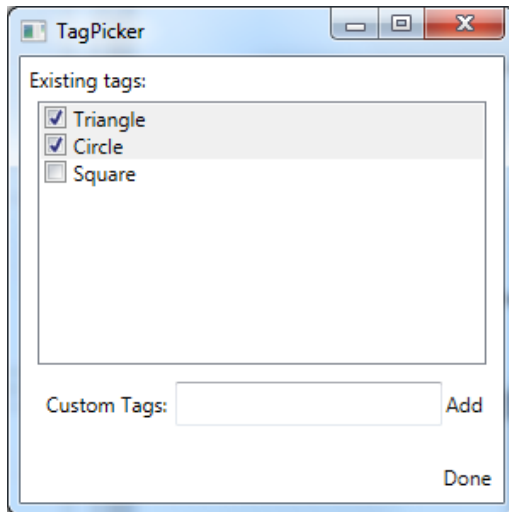


Figure 19. In the Tag Picker interface, the user can select one or more tags to apply to the chosen canvas.

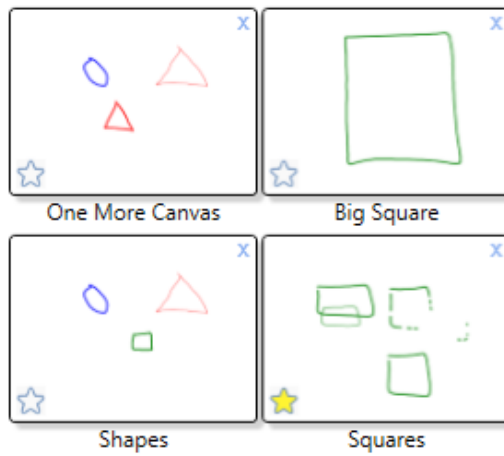
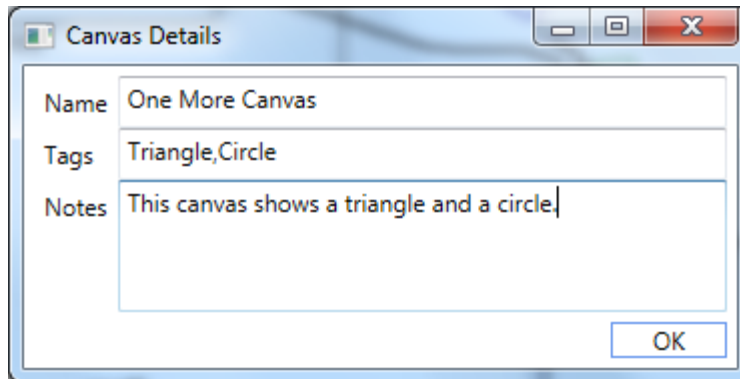
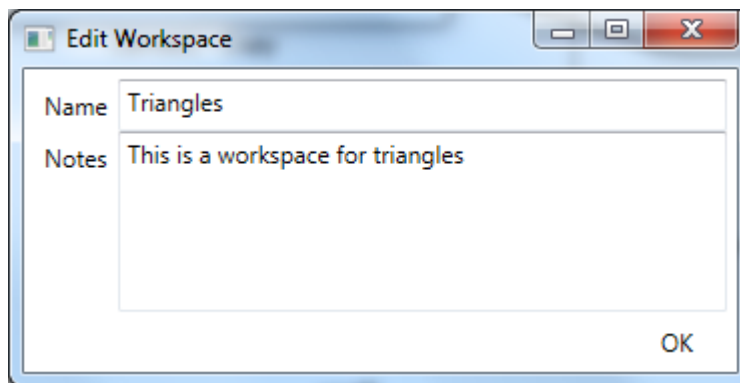


Figure 20. One of these canvases has been marked as a "favorite" (bottom right).



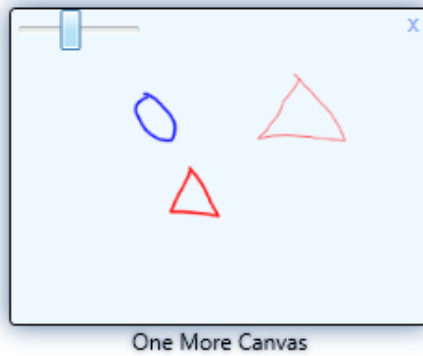
(a) Example canvas details window, with Name, Tags, and Notes fields.



(b) Example workspace detail window, with Name and Notes fields.

**Figure 21.** Canvases (a) and workspaces (b) each have metadata that can be edited to provide more information about them.





**Figure 22. Selected canvas with visible resizing slider.**

ID	Name	Created By	Tags	Notes	Created	Modified
2	Canvas1 - Copy	striplet	Triangle		1/13/2015 1:50 PM	1/13/2015 1:50 PM
3	One More Canva	striplet	Triangle,Circle	This canvas shows a triangle and a circle.	1/13/2015 1:50 PM	1/13/2015 1:56 PM

**Figure 23. Metadata for selected canvases is visible in the details pane.**

## 2. Notes

Both canvases and workspaces now have a “notes” field associated with them (Figure 21). This is a freeform text field in which the user can add whatever information she wants about the canvas or workspace. For example, she might add a description of the contents and purpose of a particular workspace, so its meaning is clear later.

## 3. Other Metadata

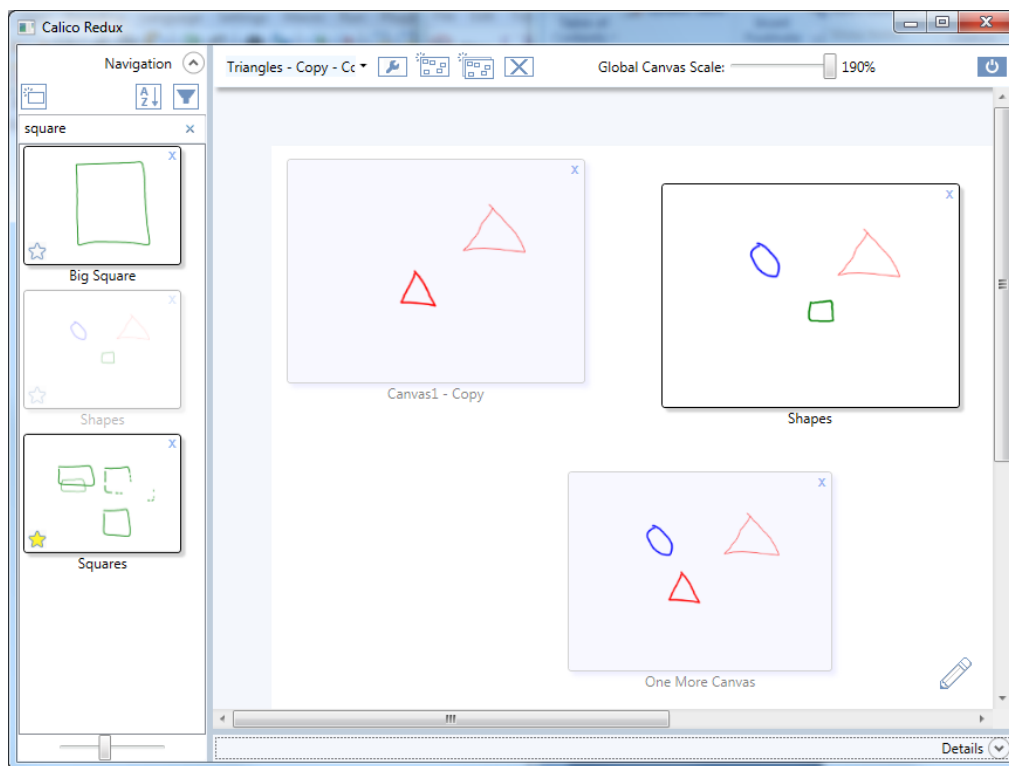
Each canvas has metadata for the date and time of creation, date and time it was last modified, and the user who created it. This data is captured automatically, and cannot be modified by the user.

## 4. Details pane

Canvas metadata is visible in a collapsible details pane at the bottom of the workspaces interface (Figure 23). When a user selects one or more canvases in a workspace or in the navigation pane, their metadata appears in this pane. Double-clicking on a record in this pane allows the user to edit the associated metadata.

## Navigation

A navigation pane on the left side of the workspaces interface allows easy navigation among existing canvases. This pane can be resized, and a slider at the bottom of the pane can resize the canvases. Canvases can be sorted (by name, date, or creator) or filtered (by tags, creator, or favorite) via menu options at the top of this pane. The search field provides searching on canvas names, tags, and notes. Visual cues are provided to further assist in locating specific canvases. Canvases which do not match existing filters or search criteria are hidden in the navigation pane and grayed out in the workspace (Figure 24). Canvases which have already been added to a workspace are grayed out in the navigation pane, but still visible.



**Figure 24.** The user has searched for “square.” Canvases not matching this term are hidden in the left pane and grayed out on the workspace. Canvases which are already in the workspace are grayed out in the left pane, but still visible.

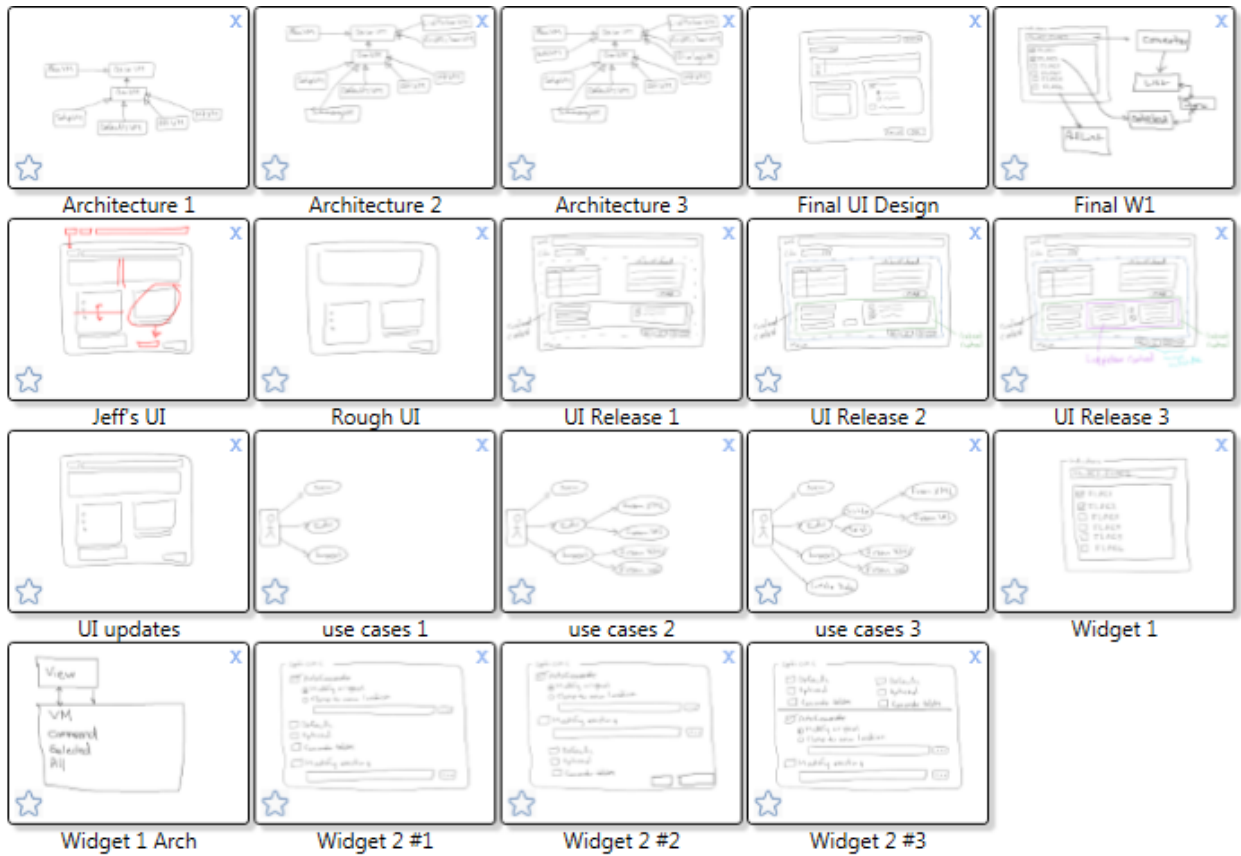


Figure 25: The navigation pane can be resized to show canvases in a grid-like arrangement.

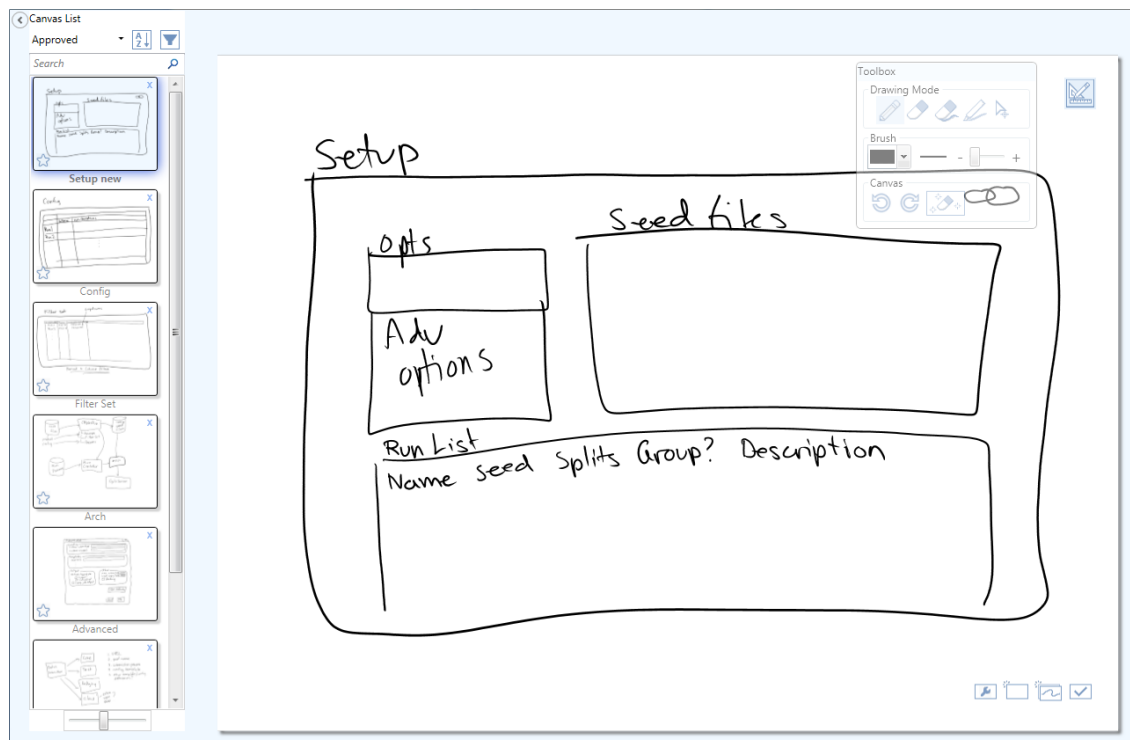


Figure 26. Navigation pane in sketching mode.

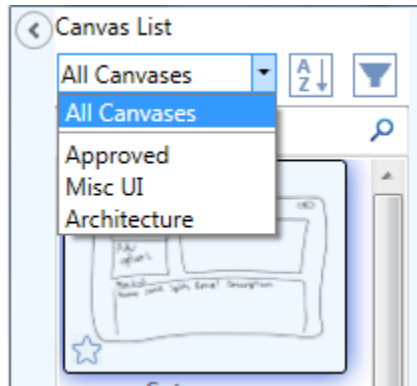


Figure 27. Users can filter canvases by workspace while in sketching mode.

The navigation pane is also available when a user is actually sketching (Figure 26). The user can open a canvas for editing from either the navigation pane or the workspace pane. When working in sketching mode, the user can quickly switch back and forth among canvases via the navigation pane. Additionally, canvases in the navigation pane can be filtered on workspace (Figure 27). This makes the organizational benefits of workspaces available while sketching as well: the user can, for example, create a “current working set” of related canvases that she frequently references together, making it easy to see all of them while working on any one of them. The active sketch is always highlighted in the navigation pane to provide context visually. As in workspace view, canvas thumbnails are resizable, so a user working on one sketch can easily reference the content of others. The combination of workspaces and the navigation pane allows users to create multiple frames of reference for any sketch and switch among them easily.

## Motivating Scenarios

These new features are intended to support designers in situations like those in our three hypothetical scenarios. They are specifically designed to make it easier to find, use, and reason about groups of sketches, better communicating a full picture of the design.

### Scenario 1

With the new tool, Janet has several options to help organize and document her sketches at design time as well as afterward. With tags, she can categorize all of her sketches into meaningful, overlapping groups. She can then search or filter to pull up just one group of interest as needed.

She can take advantage of the workspace feature to provide more structured organization as well, which is particularly useful in preparing for a meeting. Using tags and filtering, she can easily isolate a group of canvases to move onto a new workspace, where she can arrange them in a meaningful way. She can enlarge some or all canvases on the workspace to make their contents more visible. If any sketches are particularly important or require additional background information, she can include it in the canvas note. Similarly, she can include details about the workspace in the workspace note. She can also draw directly on the workspace to illustrate relationships between sketches, draw attention to certain elements, or jot additional notes.

Figure 28 through Figure 31 illustrate one way in which Janet could organize her sketches into workspaces, as an alternative to the sample project cluster we saw in Figure 9. Instead of showing all sketches on the same workspace, Janet has three workspaces, one for each spoke in the original cluster. In addition, she has one more workspace which shows the relationship between the UI widgets and the overall UI. Because there is no automatic layout for sketches in workspaces, Janet can arrange them in ways that make the relationships among them clearer than they would be in a cluster.

This is just one possible way that Janet could organize her project. With the new tool, Janet has complete control over whether and how to visualize her canvases on workspaces, rather than being limited to a single layout of all canvases. Additionally, it provides the flexibility to use manually configured layouts (workspaces) or automatically rendered visualizations (navigation pane), depending on which is most expedient for the purpose at hand.

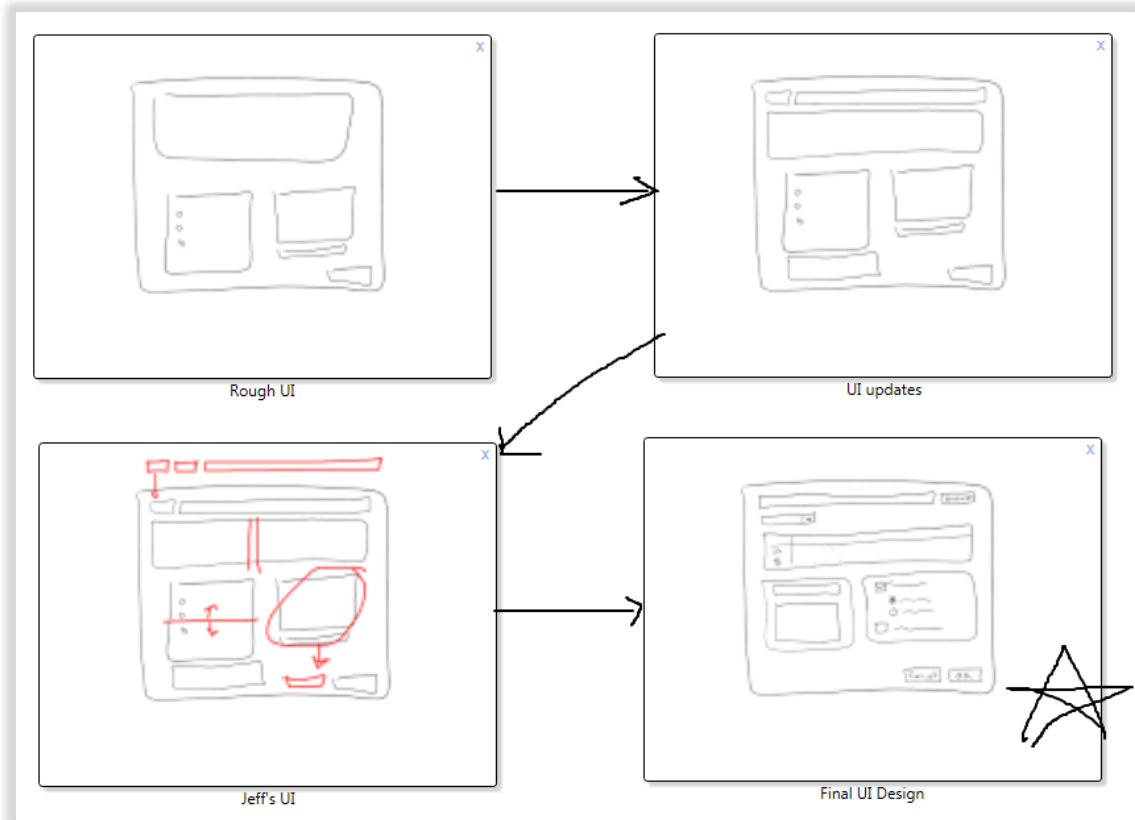


Figure 28: The UI workspace, showing the progression of the UI design to its final incarnation.

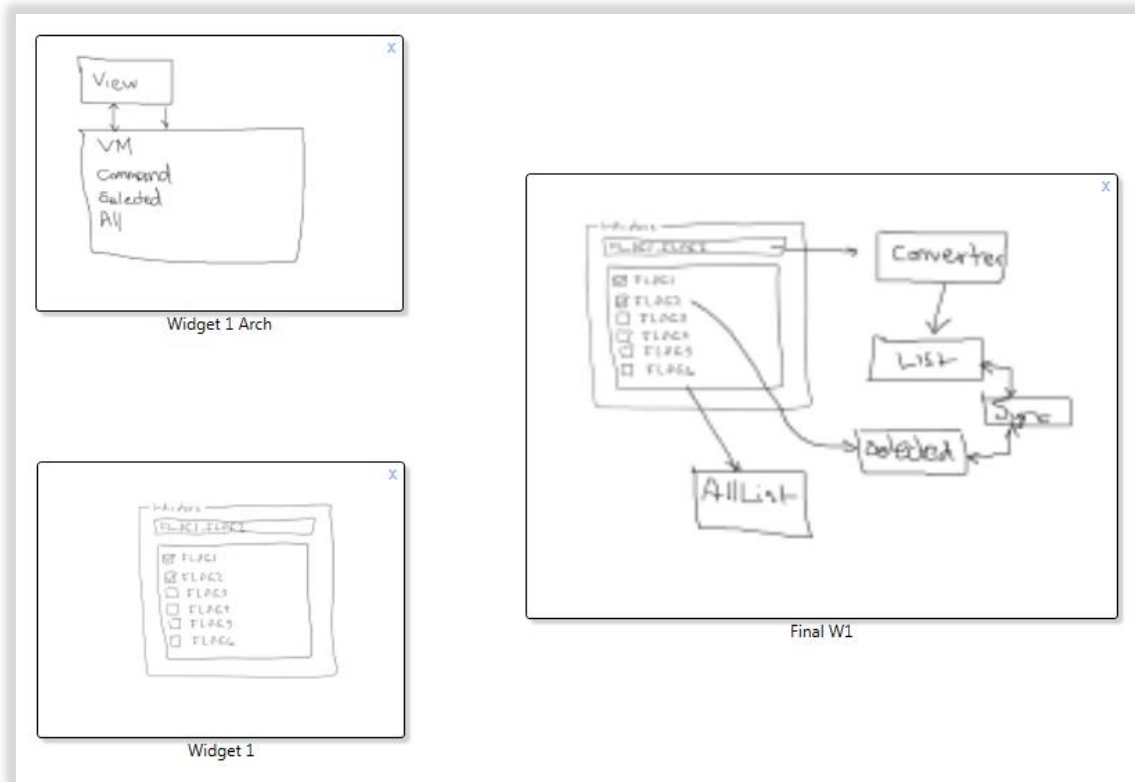


Figure 29: The Widget 1 workspace, showing three different perspectives on Widget 1.

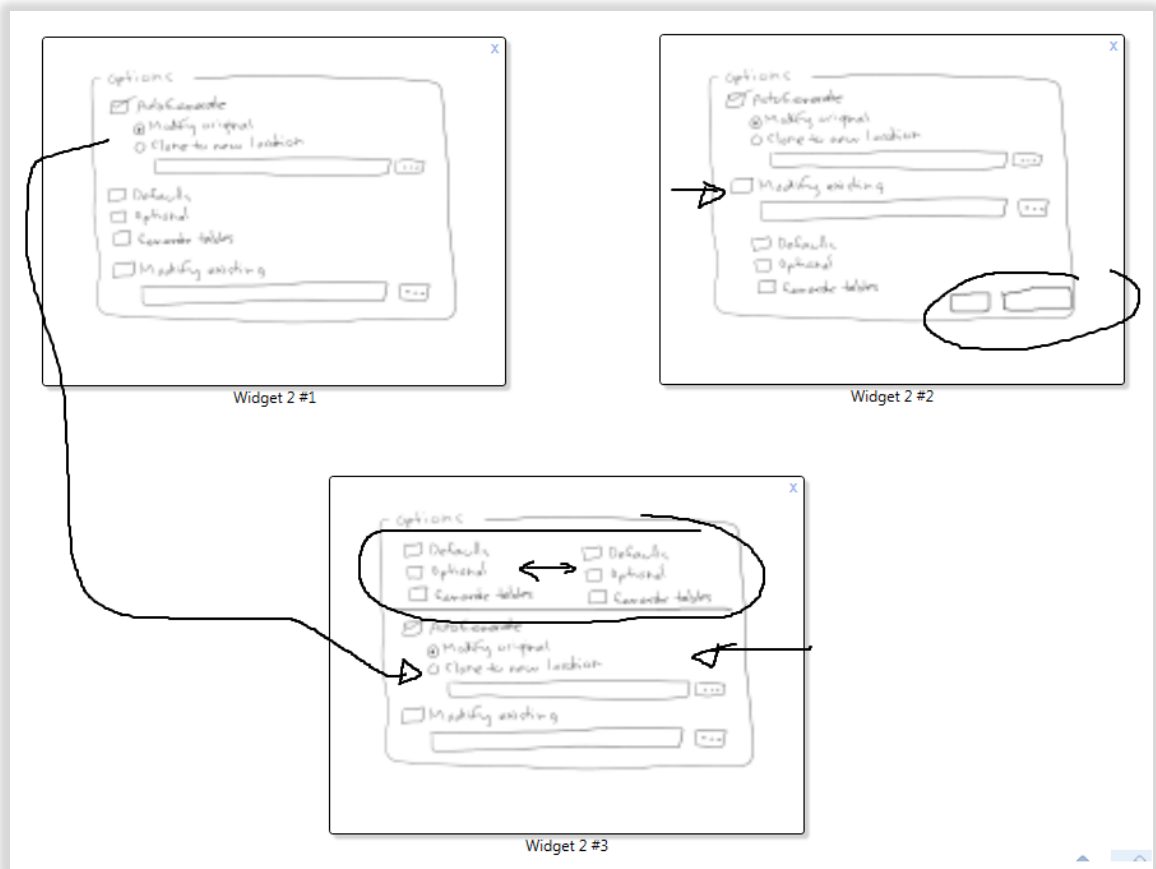


Figure 30: The Widget 2 workspace, showing three alternative designs with annotations.

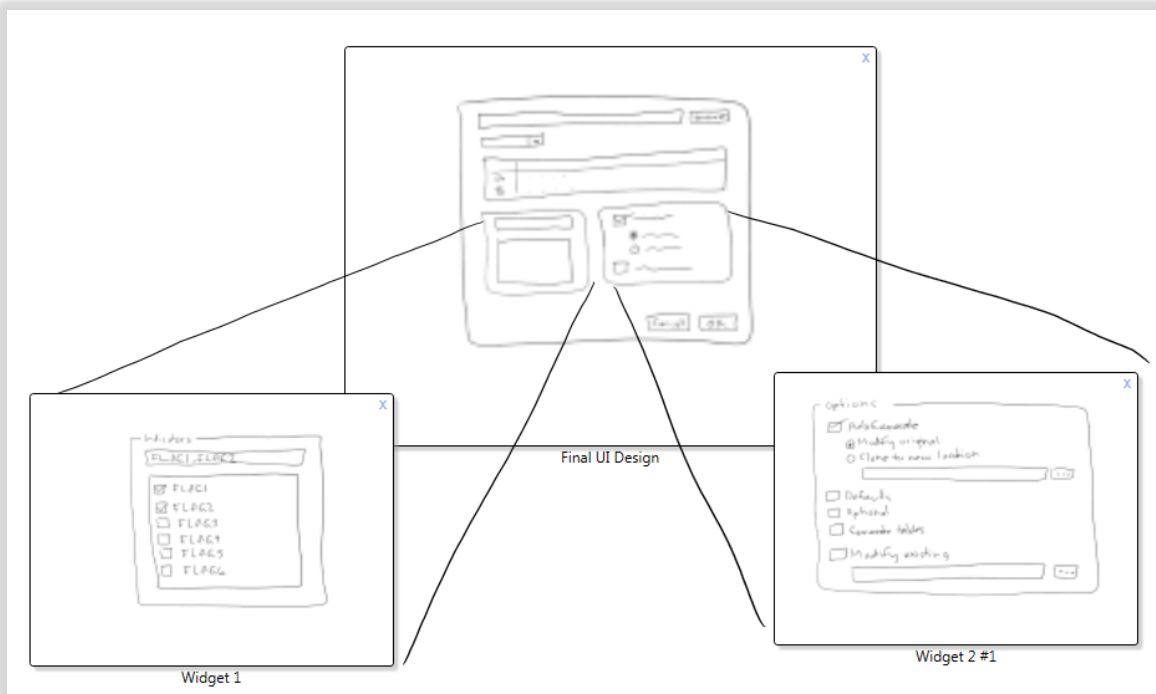
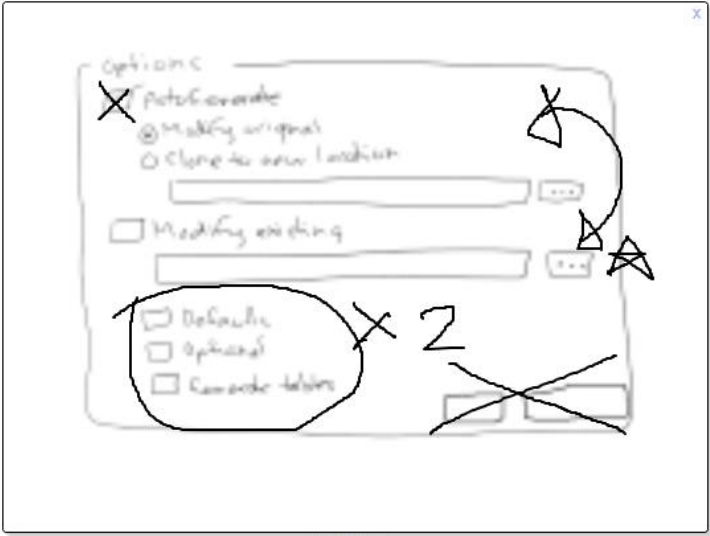


Figure 31: The "Final Design" workspace, showing the final overall UI and how the widgets relate to it.

Now, to prepare for her meeting, Janet may not need to create a separate presentation at all. Instead, she can create workspaces that present the content she would otherwise have added to a slide. Using the new tool in the meeting also aids the discussion, by allowing Janet to quickly find, annotate, and modify sketches of interest. When her boss asks about something Janet has not already prepared to show, she can search or filter to locate relevant sketches. Even if she has not added many notes or tags, she can use the navigation pane to view a grid of all canvases sorted by date and browse the most recently created ones to find the one she is looking for. Additionally, she can take notes directly on a workspace during the meeting to mark changes that need to be made to existing sketches, without modifying them directly (Figure 32). Janet can capture all of the information she needs to bring back to her team within the tool, so they can dive into the required updates more quickly.



**Figure 32: On the “Meeting Notes” workspace, Janet has marked up one of the sketches with planned changes during the meeting with her boss.**

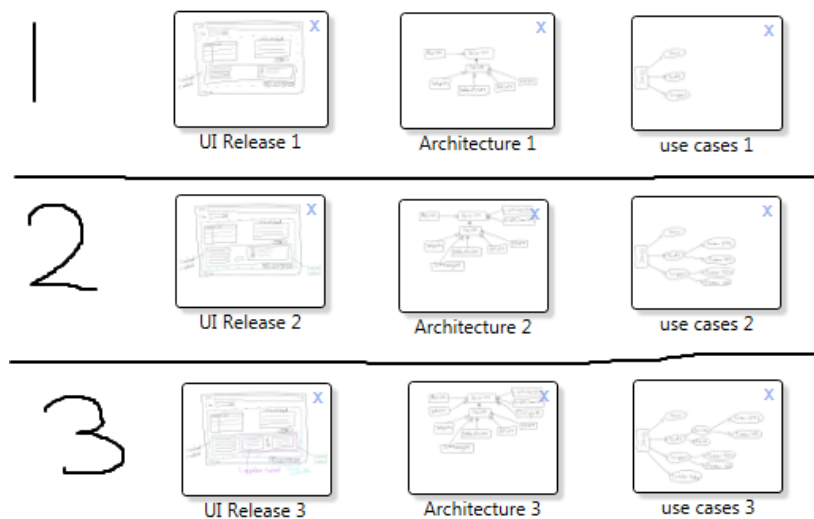
As with the current version of Calico, changes made to sketches will reflect automatically in the sketch thumbnails. However, unlike with clusters, the same sketch can be added to more than one workspace. This means that when Janet’s team modifies a sketch to incorporate the boss’s requests, those changes are visible in every workspace that includes that sketch. Additionally, as they create any required new sketches, this will not cause unexpected layout changes in existing workspaces, like it would with Calico’s clusters.



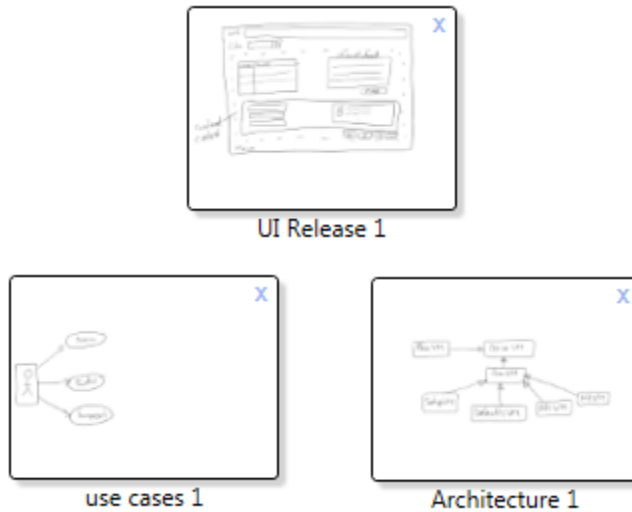
## Scenario 2

Molly and Andrew will have an easier time in their meeting with this version of Calico. They can easily juxtapose sketches in multiple configurations by adding them to new workspaces and enlarging them. Important relationships within each workspace can be captured by drawing or through workspace notes. For example, in one workspace (Figure 33), they could lay out a roadmap for the UI by arranging all UI canvases in order of projected release number, and circling the changes at each stage. In a second workspace (Figure 34), they could put all sketches related to the next release, to get a comprehensive picture of what will be included.

Like Janet, they can take advantage of tags, filters, and search to quickly find pertinent sketches to add to their workspaces during the discussion. Alternatively, they can simply expand the navigation pane and add canvases to appropriate workspaces, with the same ease that paper printouts would provide.



**Figure 33: Roadmap workspace, showing the next three releases.**



**Figure 34: Workspace for the next release, showing the planned UI design, architecture, and supported use cases.**

### Scenario 3

Again, the proposed new features make Tom’s life easier. He has a number of options for identifying the status of each sketch. One possible solution is to use tags to track the status of each sketch. First, Tom can create a “final” tag to use for sketches representing final designs. He can also create a “next priority” tag to apply to sketches of those components that the team should work on next. As the team works, they can add and remove these tags as appropriate to keep their sketches categorized correctly. Alternatively, he could simply use workspaces the same way he would have used clusters in Calico, adding and removing sketches to move them through the pipeline of workspaces.

The tagging approach also helps Tom and his team to view the design history of a component together with its final design, making it easier to recall how the project has evolved. For instance, they could create a workspace for the component, add all sketches pertaining to that component in order of modified date, and make the final design larger (as in Figure 29). Or, since sketches can have multiple tags, Tom could use a combination of “state” tags and component name tags to organize and filter along both dimensions. Another option is to use favorites to indicate final designs, allowing them to stand out visually among other sketches.

Workspaces also make it easier for Tom’s team to maintain consistency among related components in the system they are designing. They can add sketches for each component to the same workspace, providing a composite view that will update automatically when any one component changes. Then, while editing one such sketch, he can filter in the navigation pane to show the other sketches that are part of the same workspace. To make the task even easier, Tom can annotate the workspace with links showing how the components connect, highlighting those areas which require special care when making changes.

### Additional Benefits

It is worth mentioning that our proposed changes, in addition to addressing limitations of Calico with respect to our motivating scenarios, also address findings from past studies on Calico. In a previous study [5], Calico’s authors made several design recommendations motivated by their findings (Table 2). In this section, we describe how our proposed features address three of these findings (bolded).

Table 2. Design recommendations (from [5])

Empirical Result	Design Recommendation
<b>Designers simulate and discuss scenarios very frequently</b>	Enable annotating sketches with multiple scenarios.
Interactive whiteboards diminish handwriting quality.	Enable alternative text input (e.g., speech to text or text recognition)
<b>Designers work simultaneously with several canvases.</b>	Enable multiple canvases to be legibly viewed simultaneously.
<b>While separating sketches across canvases has important benefits, multiple canvases are sometimes parts of a single sketch.</b>	Enable designers to expand canvases when necessary.
Designers use impromptu notations whose meaning is forgotten when sketches are reviewed.	Enable designers to reconstruct meaning by recording and replaying audio from design sessions.
Determining the authorship of content is challenging.	Provide authorship cues as content is created.
Designers work synchronously and asynchronously, moving together between canvases and working on separate canvases.	Enable designers to temporarily subscribe to a group focus.

### Designers simulate and discuss scenarios very frequently

This is a great example of a use case that is well-served by our proposed workspaces. Given a single general-purpose sketch, scenarios can be overlaid using workspace annotations, without modifying the underlying sketch. A user could create a new workspace for each scenario, and add several canvases to each workspace that require different treatment under each scenario. She could then annotate the

canvases on each workspace to illustrate the different scenarios. Workspaces could also be used for storyboarding: canvases could be used as the panels in a storyboard, making it easy to overlay contextual annotations or rearrange and swap out panels to accommodate design changes. Alternatively, panels could be roughly sketched directly on the workspace, and then populated with one or more canvases per panel to create more complex illustrations.

### **Designers work simultaneously with several canvases**

It was an explicit goal of this thesis to address the known issue of canvas legibility in the cluster view. Our resizable canvas thumbnails in both the navigation pane and the workspace certainly satisfy the associated recommendation. However we go further in addressing this finding by also allowing designers to work with multiple canvases at once, not simply view them. By adding canvases to a workspace, designers can view, move, resize, and annotate them simultaneously.

### **While separating sketches across canvases has important benefits, multiple canvases are sometimes parts of a single sketch.**

We take a different approach to addressing this finding than what the authors recommended. Allowing larger sketches allows users to avoid unnecessarily splitting up sketches. However, there may be a middle ground where sketches are different enough to be separated, but similar enough that they are often treated as a unit. Here, our workspaces provide a mechanism for composing larger pictures from individual sketches.

## 6. Evaluation

To evaluate the proposed changes to Calico, we implemented a new prototype, on which we then performed a user study. The goal of our evaluation was to answer three research questions:

1. Do the new features help designers understand and reason about groups of sketches?
2. Do designers use the new features in ways that we did not anticipate?
3. Are there other features that would support designers in revisiting and making sense of sketches?

### Methods

For this user study, we developed a set of design tasks which focused on using existing sketches. Each participant was given a 30 minute hands-on tutorial, followed by a two hour session in which he or she worked alone through the provided list of tasks (Appendix A). The sessions were video recorded, and participants were encouraged to talk through the tasks aloud as they worked. We provided a “cheat sheet” with labeled screen shots as a reference for how to use the tool (Appendix B), but otherwise did not offer any assistance to the participants in how to use the tool or complete the tasks once they began to engage in the tasks.

The tasks themselves were designed to mimic realistic scenarios that designers might encounter at work, in which we felt our tool would be useful. For each task, participants were given a fresh instance of the tool, prepopulated with several sketches. They were asked to manipulate, organize, and review these sketches. The tasks ranged from very specific (e.g. “make this sketch as big as you can”) to very general (e.g. “organize the sketches however you see fit”). Each task instructed the participants to take certain actions using the available sketches and workspaces, but did not specify how the available features should be used to accomplish this. In addition to the list of printed tasks, we provided supplementary material for some of the tasks such as screenshots, descriptions, or lists of sketches.

Participants were recruited from a local company and chosen specifically for their familiarity with the design, use, and source code of an internal tool used at this company. The sketches that we used in the sessions pertained to this tool, to make sure the content of the sketches would be familiar and meaningful

to the participants. This was done because we did not ask participants to do any creative sketching to produce new content during the study, like they would in a real design session. Participants worked through the tasks on a large touchscreen monitor in a conference room on site at this company, in order to more closely mimic an actual design session they might participate in on the job.

One participant is a programmer in an IT infrastructure support role, while the other seven are actuarial programmers who develop financial models and related tools. All participants engage in some type of software modeling in their work.

During the sessions, we observed the participants as they worked through each task. Participants were instructed that there was no “correct” way to complete any given task, and that they were free to skip part or all of any task for any reason. Based on our experience during a pilot of the study, we wanted to prevent participants from becoming fixated on any one task or focusing more on doing the “right” thing than doing what came naturally. Additionally, to help prevent technical issues or inadequate training from interfering with data collection, we also instructed participants to describe their intended actions in cases where they wanted but were unable to do something. Before the study, we prepared a rubric of tasks and the behaviors we expected to observe during each task (Appendix C). We used this rubric to determine whether and how each participant was using the features of the tool to accomplish the given tasks. For each task, we assigned a result code to the participant’s actions. Our coding scheme is shown in Table 3.

**Table 3: Participant action coding scheme**

<b>Code</b>	<b>Meaning</b>
<b>0</b>	Unexpected features were used
<b>1</b>	Both expected and unexpected features were used
<b>2</b>	The expected features were used
<b>3</b>	Some, but not all expected features were used
<b>4</b>	No interesting features were used, or the participant did not follow directions
<b>5</b>	The task was skipped or the participant misunderstood the task.

Once a participant had completed all tasks, we conducted a debriefing interview to learn more about his or her experience. Our questions were intended to discover what motivated each participant’s choices

about what features to use for what purpose. We also sought feedback on usability and recommendations for improvements.

We recruited eight participants for this study along with two participants for a pilot. We identified a few issues with the initial study design during the pilot, which led us to make modifications for the actual study. In most cases, this involved rewording the tasks where they were confusing or misleading, adding more supplementary material, and updating the tutorial to clarify the possible uses of certain features. We also worked through some technical difficulties with the equipment. Below, we discuss the results for the eight actual study participants.

## **Findings**

### **How participants used the tool**

In a majority of cases, participants used the tool exactly as we anticipated. However, there were several interesting deviations from our expectations as well as patterns in which tasks did and did not line up with what we expected. Additionally, a handful of features were generally ignored, with participants taking advantage of other features to accomplish the same goal.

Table 4 lists the subtasks for the first four tasks, the features we expected to be used for each task, and a summary of the resulting actions for each task. The fifth and final task will be discussed below.

Participants used exactly the features we anticipated 60% of the time, and another 14% of the time they used additional features that we did not anticipate. 13% of the time, they did something other than what we expected, but still made use of our tool's important features to successfully complete the task. What is most interesting about these results is the tasks where users differed significantly in their behavior from either our expectations or each other.

In a few cases, participants made use of features in unexpected ways or to do things that the tasks had not specifically asked of them. Several participants took immediately to the features of the workspace, easily arranging and resizing canvases even when the task did not explicitly call for it. Participants often

filtered or sorted sketches in order to more quickly locate those sketches that they were being asked to work with. In a few cases, participants independently chose to add descriptive notes to their workspaces.

**Table 4: Number of participants matching each result code for each subtask**

Task	Expected Features	Result Code					
		0	1	2	3	4	5
1a	New workspace, edit workspace name		1	7			
1b	Navigation pane		1	7			
1bi	Scale slider OR workspace drawing		5	3			
1bii	Workspace drawing		3	3		2	
1biii	Detail pane	7		1			
1c	Navigation pane		1	7			
1ci	Canvas slider AND global slider			5	2		1
1cii	Workspace drawing			4		4	
2a	New workspace, edit workspace name		1	7			
2b	Navigation pane		3	5			
2c	Clone sketch, pen color		2	4		2	
2d	New sketch, pen color	3		4			1
2e	New sketch	1	1	6			
2f	Navigation pane, workspace, workspace drawing		2	5		1	
3a	Canvas detail, tag picker, sorting	3		2	3		
3b	Favorite		1	5	2		
3c	Filter, new workspace, navigation pane			6	1		1
4a	Sort, filter		3	5			
4b	Search	7		1			
4c	Filter OR sort and detail			8			
4d	Workspace name or tags	1		6			1
Total Counts		<b>22</b>	<b>24</b>	<b>101</b>	<b>8</b>	<b>9</b>	<b>4</b>
Percentage		<b>13%</b>	<b>14%</b>	<b>60%</b>	<b>5%</b>	<b>5%</b>	<b>2%</b>

Tagging was another feature with some interesting behavior. Participants were generally comfortable with adding and using tags, but only two participants used the tag picker at all. During the debriefing interviews, most participants mentioned that they forgot the tag picker existed. One of the participants preemptively added all of the tags he would need to the tag picker, so that it would be easier to tick the appropriate tags for each canvas (though this use case was not actually supported by the tool, so his tags were not saved).

About half of the participants drew directly on canvases instead of the workspace, even when explicitly asked to draw on the workspace. When asked about this during the interviews, they usually cited a desire to draw in color, which the workspace drawing feature did not support. One participant had



trouble opening the workspace drawing menu, and another wanted a larger working surface than the workspace provided.

In two instances, almost all participants deviated from our expectations. First, most participants did not use the search bar at all. However, one participant who had early success with search chose to use it for almost every task that involved finding and interacting with a particular sketch. In the interviews, several participants mentioned that they had forgotten about the search bar. One participant said that he actually preferred not to use search: “I like to see all the data at once and select what I need.”

Second, the detail pane was almost universally ignored as an entry point to editing canvas metadata. All but one participant opened the canvas first and edited its details from there, rather than opening the canvas details directly. However, many participants did use this pane to browse canvas metadata after selecting multiple canvases, either to look for information of interest or to check their work when adding tags.

In general, participants made heavy use of the navigation pane and detail pane for browsing information, even when they were not explicitly taking advantage of the features we expected. Many participants would sort or filter just enough so that they could visually browse the remaining sketches easily.

The fifth and final task was deliberately open-ended, designed to allow participants to use whatever features they wanted to organize a set of canvases. Participants were asked to organize a prepared set of sketches and workspaces illustrating a new design for the same tool that was the subject of the previous tasks. They were given a handout listing all sketches with brief descriptions to help inform the organization task, but were not given any instructions as to specifically how the sketches should be organized. They were also asked to delete sketches that were obsolete or not useful, and we identified these sketches in the handout. As before, we asked participants to describe their intended actions in cases of technical difficulty or time constraints. For this task, we noted which features were used and how they were used.

Table 5 shows how many participants used each feature. Unsurprisingly, all participants made use of the canvas list in the navigation pane and the workspace feature. Almost all participants used tags, and at least half used sorting, resizing, and annotations. Notably, at least one person used every feature. More interesting than simply what features were used is how each participant approached the task and made use of the available features.

**Table 5: Features Used in Task 5.**

<b>Feature</b>	<b>Number of Participants Using</b>
<b>Annotations</b>	5
<b>Canvas Notes</b>	3
<b>Canvas List</b>	8
<b>Detail Pane</b>	1
<b>Favorites</b>	2
<b>Filtering</b>	3
<b>Navigation Size Slider</b>	1
<b>Canvas Resizing</b>	4
<b>Search Bar</b>	1
<b>Sorting</b>	6
<b>Tags</b>	7
<b>Workspace Notes</b>	1
<b>Workspaces</b>	8

All but one participant spent time at the beginning of the task scrolling through the canvas list to review all available canvases. Most also checked the list of tags early in the task to see if any existing tags could be used to identify sketches. Several participants remarked with disappointment on the lack of predefined tags. All of these behaviors suggest that these features are useful in helping newcomers to a project get a sense of what is going on in that project.

In addition to this “preliminary review” behavior, participants regularly inspected the sketches when determining what to do with them. This included browsing the sketch descriptions on the handout and inspecting the content of the sketches themselves. Only one participant used the navigation size slider to enlarge the canvas thumbnails. Several others opened them for editing. In both cases, the goal was to better inspect the content in order to determine whether and how to incorporate the sketch into a particular workspace.

One important aspect of this task was locating sketches of interest, whether to delete them, add them to a workspace, or simply determine if they had already been organized in some way. Participants

generally sorted all canvases by name in order to quickly find a given sketch. Of the two participants who did not use sorting, one used search for this purpose instead, while the other ignored the handout, and therefore did not need to locate sketches by name. Participants also used tags and favorites as a means of identifying or flagging sketches to make them easy to find later. One participant was interested in identifying all sketches already belonging to a workspace, and investigated using tags or favorites to flag such sketches so that he could move all non-flagged sketches to a new workspace.

While all but one participant used or wanted to use tags, their approaches and goals were quite different. One participant spent time tagging every sketch, and then filtered on those tags to select overlapping categories of sketches to add to new workspaces. Another created tags for component names and categories, but did not subsequently use them for identification. Other participants were hoping that tags had already been added, so that they could use them to help organize the sketches.

Similarly, each participant used the workspaces slightly differently. One participant did not create any new workspaces or add any sketches to the existing workspaces, instead using only tags to categorize them. Others created several new workspaces and spent time arranging canvases on them, including resizing and annotating sketches. Most ended up creating mutually exclusive workspaces, and two participants actually made this an explicit goal. Those that did create workspaces typically created at least an “architecture” and a “UI” workspace, with one or more additional workspaces that varied between participants.

Those participants who annotated their workspaces did so in similar ways. The three uses of annotations were: 1) to show relationships between sketches, 2) to label groups of sketches, and 3) to highlight important sketches or parts of sketches. In one interesting case, a participant actually sketched out the high level UI on the workspace itself, and drew arrows showing how the components in the sketches connected to the main UI.

In summary, participants used the features of our tool in very different and sometimes unexpected and creative ways. Additionally, they each used different approaches to organizing, identifying different grouping characteristics in the sketches. While there were some commonalities, such as similarly named

workspaces, participants still differed in how each sketch mapped to the dimensions along which they wanted to organize the collection.

### **What participants thought about the tool**

At the end of each session, we interviewed the participants about their experience. We wanted to gain an understanding of what the participants liked and did not like, what influenced their use of features in the tool, and how applicable they thought it might be to real work. A list of questions we used for this interview can be found in Appendix D.

Participants responded positively to the tool, saying that it was “intuitive,” “easy to use,” “straightforward,” and well-suited to its purpose. All participants noted issues with the touchscreen interface that had interfered with usability, but correctly attributed these to hardware malfunctions. Most participants did not reference the cheat sheet at all, and those who did used it only once or twice. When asked about this in the interviews, participants said that they felt the training was sufficient and the tool was intuitive enough that they did not need it. One participant remarked, the tool had “just enough features to remember – not so many that I got lost.” On the other hand, most of the participants did report forgetting that one or more features existed. This suggests that the number of features was not so large as to be overwhelming, but large enough that thirty minutes of training is not sufficient to cement all of them in every participant’s memory.

When asked to describe which features they found most useful, participants generally referred to our tool’s most important features, though a few talked about the sketching capabilities. They liked being able to create and annotate a view of one or more canvases “without messing them up.” They also noted the ease with which relationships could be explained using workspace annotations. Several participants also mentioned tags, filters, and searching as being easy to use and very useful in locating sketches of interest.

We also asked participants whether they thought any of the features were difficult to use, and what made them so. Participants were less forthcoming in this area, but generally gave feedback around usability. They felt many of the interactions were cumbersome, such as double-clicking to edit canvas

details, using the sliders, or switching input modes. It is worth noting that calibration issues with the touchscreen greatly exacerbated these issues for some participants. One participant thought it was too easy to accidentally delete canvases, and wanted a confirmation popup for that action.

Finally, we wanted to get a sense for how well the tool's features supported design tasks, both those we had assigned and tasks the participants might encounter in their jobs. With respect to the assigned tasks, unsurprisingly, participants identified the tasks which could be completed with a simple sort or search as the easiest. However, there were a few instances where participants did not use the expected features for these tasks, and either gave up, guessed, or got the wrong answer. The fact that the same task was trivial when a feature was used but impossible when it was not suggests that these features could be hugely impactful under the right circumstances.

Participants also agreed that the assigned tasks would have been difficult to perform without this tool, such as when using paper sketches or a whiteboard. They remarked that it would be difficult if not impossible to illustrate relationships between sketches without such a tool, and that the tool made it very easy to organize sketches in a meaningful way. They also noted the usefulness of the canvas metadata and the ability to search, sort, and filter on this data.

We also asked participants whether and when the tool would have been useful in their own work. Two participants pointed to situations where they have had to present information to stakeholders. They highlighted its support for creating multiple perspectives for different audiences, while allowing the presentation to be interactive. A few of the participants referred to recent design sessions that would have benefited from this tool, particularly noting that it would help them keep the big picture in mind and document relationships between parts of the design. One participant felt that the tool was marginally helpful, but probably not necessary for his work.

### **Recommendations for enhancements**

Based on participant statements during the tasks and interviews as well as our observations, we have identified several potential enhancements that would improve our tool. The most common types of requests, mentioned by over half the participants, were:

**Multi-canvas operations:** Enhancements in this category relate to performing operations on several canvases at once. Participants wanted to be able to tag and delete multiple canvases at once as well as group, move, and resize multiple canvases in the workspace. A Select All feature was also requested.

**Contextual cues:** On several occasions, participants forgot that a filter was turned on or that they had entered workspace sketching mode. Some kind of visualization to alert the user of these states would have been helpful.

**Workspace annotations:** Several participants wanted more features related to workspace annotations – more colors, text entry, highlighting, and the ability to select and move annotations.

**Usability:** Participants ran into various issues involving actions they tried or wanted to take, but which were not supported. These include drag-and-drop support for adding canvases to a workspace, an option to clear all filters, deleting canvases from sketching mode, and allowing larger canvas sizes in the workspace. There were a handful of other issues related to bugs.

In addition to these common themes, there were several other suggested enhancements recommended by one or more participants that we felt were particularly interesting and relevant to the intent of this tool. These are shown in Table 6.

**Table 6: Features requested by one or more participants**

<b>Feature</b>	<b>Description</b>
<b>Detail sorting</b>	Allow the detail pane to be sorted on any field by clicking the header
<b>Meta-workspace</b>	Provide one more level of abstraction, such as a folder structure or meta-workspaces where users can work with thumbnails of workspaces.
<b>Import images</b>	Allow users to add their own images to a workspace
<b>Juxtaposition while sketching</b>	Provide better support for juxtaposition while in sketching mode, particularly to support viewing one canvas while editing another. <sup>1</sup>
<b>Tag creation</b>	Allow tags to be created without adding them to a canvas
<b>Detail editing button</b>	Add an obvious in-line button to open canvas details for editing. Double-clicking is not an obvious or easy interaction.
<b>Summary statistics</b>	Provide a summary of number of canvases by creator, tag, etc.

<sup>1</sup> The navigation pane does allow users to view all sketches while a single sketch is open for editing, but participants were interested in something more like a picture-in-picture view of a second sketch.

## Discussion

We approached this study with certain expectations of how our tool's features would be used. In many cases, participant behavior matched our expectations exactly, suggesting that our feature set was well-designed to support its intended purpose. Additionally, we observed deviations from our expectations in terms of which features were used and how they were used. Both types of findings present several important lessons that can inform future work on Calico and similar tools.

First, while some features were ignored by most participants, every feature was used by at least one participant. Often, participants used these less popular features for a purpose that other participants simply did not think of. Even for those features that were used by all or almost all participants, participants often used them in very different ways, and for purposes that we had not anticipated. These facts suggest that different features appeal to different types of users, and that we provided a good breadth of flexible features that could be adapted to serve these differing preferences.

Second, the tool's usefulness was negatively impacted by both technical problems with the hardware and usability issues in the tool (which themselves were often exacerbated by the technical issues). These issues contributed to the infrequent use of certain features compared to those that were easier to use. However, participant feedback suggests that these issues are not insurmountable. Indeed, participants gave several suggestions to improve usability, and suggested that they would have used these problematic features more if not for these issues. Therefore, we believe that our features do seem valuable and likely would be even more valuable with improvements to usability. In other words, the feature set is "right," but the implementation needs improvement.

Third, one major area of interest for participants was the ability to sort, filter, and search on metadata, as well as browse all available sketches. The navigation and detail panes provided a set of features which are not particularly groundbreaking in and of themselves, but combined, they provide a powerful way of addressing the problem of efficiently locating a particular sketch among many. They also address a

demonstrated need for easily browsing and reviewing many sketches at once. One of our goals was to make it easy for designers to identify sketches of interest, so we believe our feature set met this goal.

Finally, of all the features we provide, participants expressed the most satisfaction with workspaces in general, and annotations in particular. They liked being able to arrange sketches on a workspace to tell a story, and anticipated benefits to being able to share these stories with stakeholders by using our tool during presentations. Even more than this, though, they liked being able to use annotations to record information *about* the sketches, “without messing them up.” They also noted the ease with which relationships between sketches could be explained using workspace annotations. Our overall aim with this tool was to help designers work with and reason about groups of sketches, with the primary mechanism being the workspace. Given that our participants believed that workspaces would indeed help them visualize, understand, and explain groups of sketches and the design stories they embody, we believe that our tool represents an important step forward in supporting such activities.

## **Threats to Validity**

**Internal Validity:** The participants in this study were current coworkers of the primary author of this thesis at the time of the study, which took place on site at work. They may have been favorably biased toward the tool because of this relationship and setting, and more interested in finding ways to use it than an average user might be when faced with a new tool. Participant reactions to the tool were overwhelmingly positive, with all criticism focused on usability or implementation problems rather than the features *per se*. While participants did suggest enhancements to existing features to improve them, as well as new features, no participant expressed a strong negative opinion of any existing feature as such. This feedback could be explained by a favorable bias. Nevertheless, the positive feedback tended to be consistent across participants, and they did use our tool to complete the assigned tasks successfully. This suggests that, while there may be weaknesses in our tool which were not revealed by this study, we can have confidence in its strengths. Furthermore, the fact that participants did find unexpected ways to use the tool, even if they might have been biased toward doing so, suggests that it has promising applications.



**External Validity:** Although we made an effort to create realistic tasks using real-world content that was familiar to the participants, working through artificial tasks is not the same as using this tool for real work. The tasks we assigned were precisely the kind of tasks that our proposed features are intended to support, so it is no surprise that these features were so heavily used. It is possible that these tasks are not adequately representative of the problems that designers would commonly encounter. However, the tasks we asked them to perform seemed to resonate with most participants, and no participant suggested that any of the tasks were unrealistic or would never happen in real life. Thus, we feel that our tasks did represent at least a subset of those issues that designers typically face, and that our tool therefore does provide significant benefits in such situations.

Another potential threat is that all but one of the participants are in nontraditional developer roles: they are actuaries working on actuarial modeling tools, not members of traditional software development teams. The experience of traditional developers who work on larger-scale design problems may be quite different. Furthermore, the number of participants was relatively small, so we are unable to draw any statistically significant conclusions from this study. Combined with the specialization in participant work experience, this means that our findings may not generalize well. However, since our goal was simply to assess the usefulness of our prototype and seek feedback on potential enhancements, we feel that this small study still provided valuable information.

## 7. Related Work

In section 2, we reviewed several tools created to help designers work with sketches. Most of the tools focus on the act of sketching: automatic sketch recognition, capturing emergent notation, assisting with translation to formal models, and so on. Such tools, while undeniably useful, do not have direct applications to the particular problems in our motivating scenarios. Here, we further examine only those tools which provide the most relevant benefits for the teams in our motivating scenarios.

The Design Meeting Agent aims to capture design rationale through audio, video, and screen capture during design sessions [17]. This tool would be most useful in scenario 3, as Tom's team attempts to pick up where they left off. Since the previous design sessions would have been recorded, the team can review relevant portions of the recordings to remind themselves of the project status and key pieces of information. For example, they may be able to easily identify the finalized designs and the ending project state by reviewing just the most recent design session or a portion thereof. If their saved sketches are timestamped, they can quickly jump to the corresponding points in the recordings to review the surrounding context. These benefits, while useful in general, do not directly address the issues faced by the teams in scenarios 1 and 2, and only partially address Tom's needs.

The Augmented Interaction Room (AugIR) records designers at work and displays multiple perspectives of an active design at once, allowing users to annotate diagrams with data that can be used to filter [4]. Additionally, it captures context history and provides mechanisms for navigating this history in order to retrieve relevant contexts. All three hypothetical teams would benefit from working in such an environment, which matches well to their needs. Each wall in the AugIR can display one sketch and any number or related artifacts, which can be rearranged and edited on the fly. As with our tool's workspaces, the three teams could use this feature to examine overlapping groups of the same sketches (scenario 2), view different perspectives on their projects (scenarios 2 and 3), and prepare content to present to others (scenario 1). Additionally, the context history navigation would help Tom's team recover the state of their project in scenario 3. However, this tool is an extreme solution that may not be feasible to

implement in most design teams, requiring a dedicated room and special hardware. Additionally, the walls do not allow users to display any content at any time: they work together to show perspectives on a single chosen component. This makes it ill-suited for the kind of ad-hoc review in which our hypothetical teams engage.

Ideavis [19], the Interactive Mural [20], and the Designer's Outpost [21] provide features most similar to our workspace concept. Each provides a large interface on which sketches or other artifacts can be pinned, arranged, and resized. Many of the benefits of workspaces that we discussed in Section 5 would also apply with these tools. However, as with most tools in this area, these are focused more on the active design session. They do not elevate the pinned layouts to the status of design artifacts, as we do with workspaces. This means that while context could be recovered by replaying history, there is no mechanism specifically intended to capture arrangements of sketches as meaningful data for later review. Thus, for example, Molly and Andrew could pin and rearrange sketches as they look at different groups together, but there is no concept of "saving" such groups and then switching among them. While snapshots can be taken of the pinned groups, the content of those snapshots is fixed and does not update as the component sketches are changed.

## 8. Conclusions and Future Work

In this thesis, we wanted to provide a better way for designers to organize their sketches during and after design sessions, to help address some of the roadblocks that currently prevent designers from using archived sketches effectively. Our goal was to help designers organize, identify, locate, and view sketches in a way that makes those sketches effective vehicles for communication and understanding both during and outside of the design sessions in which they were created.

Our motivating scenarios laid the groundwork for thinking about how our prototype compares to existing sketching and design tools in situations that designers might encounter. Based on the resulting analysis, we believe our prototype has clear benefits in helping designers use collections of sketches to communicate with stakeholders, bring team members up to speed, examine multiple perspectives on a design problem, maintain consistency between different aspects of a project, and engage in project planning. While the current version of Calico provides many important benefits to our hypothetical design teams, our prototype better supports those scenarios. Additionally, no other tool treats spatial arrangements of sketches as first-class design artifacts or provides a means of saving multiple overlapping configurations of content that will update automatically, as we do with workspaces. The features we have implemented are both powerful and flexible, providing lightweight mechanisms for capturing vital metadata without overly burdening the designer with any formal requirements or processes.

With these hypothetical benefits in mind, our user study demonstrated the utility and flexibility of these features in tackling real design tasks. Based on our observations of the participants in action, as well as their own statements in debriefing interviews, our prototype was extremely useful in accomplishing the assigned tasks. Furthermore, participants largely believed that the tool would be useful in their own day-to-day work, with several naming specific projects for which they would like to use it.

Additionally, the variety of solutions that the participants were able to achieve by using this tool speaks to its versatility in supporting different approaches to design, without forcing designers into any particular paradigm. While they often used features exactly in the ways that we anticipated they would,

they also came up with new and creative ways to use them, particularly in the final, open-ended task. Therefore, we believe that our tool's features are both well-designed in service of their intended purposes, and flexible enough to be used for purposes that we did not and cannot anticipate.

While our findings are quite promising, they also reveal several opportunities for future work. First, as we discussed, our user study was quite small and represented a specialized subset of designers. A larger study, involving participants with traditional software design backgrounds, would provide valuable evidence as to the benefits and weaknesses of this tool. Additionally, it would be extremely helpful to observe designers using the tool over multiple design sessions, to evaluate its benefits in reusing one's own sketches.

Second, we identified several enhancements that would improve our tool's power and usability. Most of these enhancements fall into the category of "quality of life" improvements. These are changes that make the tool easier to use, and therefore help designers work more efficiently. The enhancements of this type that we believe are the most important are as follows:

- a) Multi-canvas operations, such as adding tags, deleting, resizing, moving, etc.
- b) Displays and contextual cues that indicate what filters are active, whether the user is sketching or erasing on the workspace, etc.
- c) Drag-and-drop interactions for adding and removing canvases.
- d) Improved workspace annotations, ideally providing the same features when sketching on a workspace as when sketching on a canvas.

In addition to these more incremental changes, there are two major enhancements we have identified that represent significant modifications to the tool's functionality. These changes fundamentally change how the tool works, in a way that we believe will make it yet more powerful and better-suited to our goals. The first enhancement is to add one more level of abstraction to the workspace-canvas hierarchy, a sort of "meta-workspace" level. At this level, users would be able to work with thumbnails of workspaces—organizing them, naming them, visualizing them, and annotating them. This feature becomes particularly important when multiple users are working on multiple projects. The second major

enhancement is to provide a picture-in-picture type interface for editing canvases. Rather than (or in addition to) seeing all canvases in a side panel, the user could choose one canvas to display at a relatively large size while editing another canvas. This could be implemented as a dockable floating window that the user can optionally choose to show. It would also support easy switching between two canvases of interest.

While the “quality of life” enhancements are necessary for the ultimate goal of having a tool that will actually be adopted by users, these two major enhancements are much more fundamental to the purpose of the tool. These can be approached in parallel, particularly since the former will make future user studies easier and more informative, as users will not be distracted or encumbered by usability issues.

## Bibliography

- [1] M. Cherubini, G. Venolia, R. DeLine and A. J. Ko, "Let's Go to the Whiteboard: How and Why Software Developers Use Drawings," in *CHI 2007 Proceedings*, San Jose, CA, 2007.
- [2] S. Baltes and S. Diehl, "Sketches and diagrams in practice.," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, Hong Kong, China, 2014.
- [3] L. Lichtschlag, L. Spsychalski and J. Bochers, "Using hand-drawn sketches connected to code bases in navigation tasks.," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2014.
- [4] M. Kleffmann, M. Book, E. Hebisch and V. Gruhn, "Automated Versioning and Temporal Navigation for Model Sketches on Large Interactive Displays," in *SAC'14*, Gyeongju, Korea, 2014.
- [5] N. Mangano, T. D. LaToza, M. Petre and A. van der Hoek, "Supporting Informal Design with Interactive Whiteboards," in *CHI 2014*, Toronto, ON, Canada, 2014.
- [6] N. Mangano, T. D. LaToza, M. Petre and A. van der Hoek, "How Software Designers Interact with Sketches at the Whiteboard," *IEEE Transactions on Software Engineering*, vol. 41, no. 2, pp. 435-156, 2014.
- [7] N. Mangano, A. Baker, E. N. Dempsey and A. van der Hoek, "Software Design Sketching with Calico," in *ASE '10*, Antwerp, Belgium, 2010.
- [8] J. A. Landay and B. A. Myers, "Interactive sketching for the early stages of user interface design.," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995.
- [9] M. Petre, "Team coordination through externalized mental imagery," *International Journal of Human-Computing Studies*, vol. 61, no. 2, pp. 205-218, 2004.
- [10] R. Bellamy, M. Desmond, J. Martino and P. Matchen, "Sketching tools for ideation," in *ICSE*, 2011.
- [11] B. Tversky, M. Suwa, M. Agrawala, J. Heiser, C. Stolte, P. Hanrahan, D. Phan and e. al., "Sketches for design and design of sketches," in *Human Behaviour in Design*, Springer Berlin Heidelberg, 2003, pp. 79-86.
- [12] J. C. Jones, *Design methods*, New York, NY: John Wiley & Sons, Inc., 2002.
- [13] Y. Y. Wong, "Rough and ready prototypes: Lessons from graphic design," in *SIGCHI conference on human factors in computing systems*, 2002.

- [14] C. H. Damm, K. M. Hansen and M. Thomsen, "Tool Support for Cooperative Object-Oriented Design: Gesture Based Modeling on an Electronic Whiteboard," in *CHI '2000*, The Hague, Amsterdam, 2000.
- [15] T. Hammond and R. Davis, "Tahuti: A Geometrical Sketh Recognition System for UML Class Diagrams," in *AAAI Spring Symposium: Sketch Recognition*, 2002.
- [16] E. D. Mynatt, T. Igarashi, W. K. Edwards and A. La-Marca, "Flatland: new dimensions in office whiteboards.," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999.
- [17] T. Hammond, K. Gajos and R. Davis, "An Agent-Based System for Capturing and Indexing Software Design Meetings," in *WAID*, 2002.
- [18] G. Bortis and A. van der Hoek, *DesignMinders: A Design Knowledge Collaboration Approach*.
- [19] G. F., J. Budzinski and H. Reiterer, "IdeaVis: a hybrid workspace and interactive visualization for paper-based collaborative sketching sessions," in *Nordic CHI*, 2012.
- [20] F. Guimbretière, M. Stone and T. Winograd, "Fluid interaction with high-resolution wall-size displays," in *UIST 2001*.
- [21] S. R. Klemmer, M. W. Newman, R. Farrell, M. Bilezikjian and J. A. Landay, "The designers' outpost: a tangible interface for collaborative web site.," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001.
- [22] K. Wideroos and S. Pekkola, "Presenting the Past: A Framework for Facilitating the Externalization and Articulation of User Activities in Desktop Environment," in *Proceedings of the 39th Hawaii International Convergence on System Sciences*, 2006.
- [23] K. Gyllstrom, "Passages through time: chronicling users' information interaction history by recording when and what they read.," in *Proceedings of the 14th international conference on Intelligent user interfaces*, Sanibel Island, Florida, 2009.
- [24] L. Kelly, M. Fuller, Y. Chen and G. J. Jones, "A Study of Remembered Context for Information Access from Personal Digital Archives," in *IiX'08, Information Interaction in Context*, London, UK, 2008.



## Appendix A: Study Tasks

Below is a reproduction of the list of tasks that participants received.

### Tasks

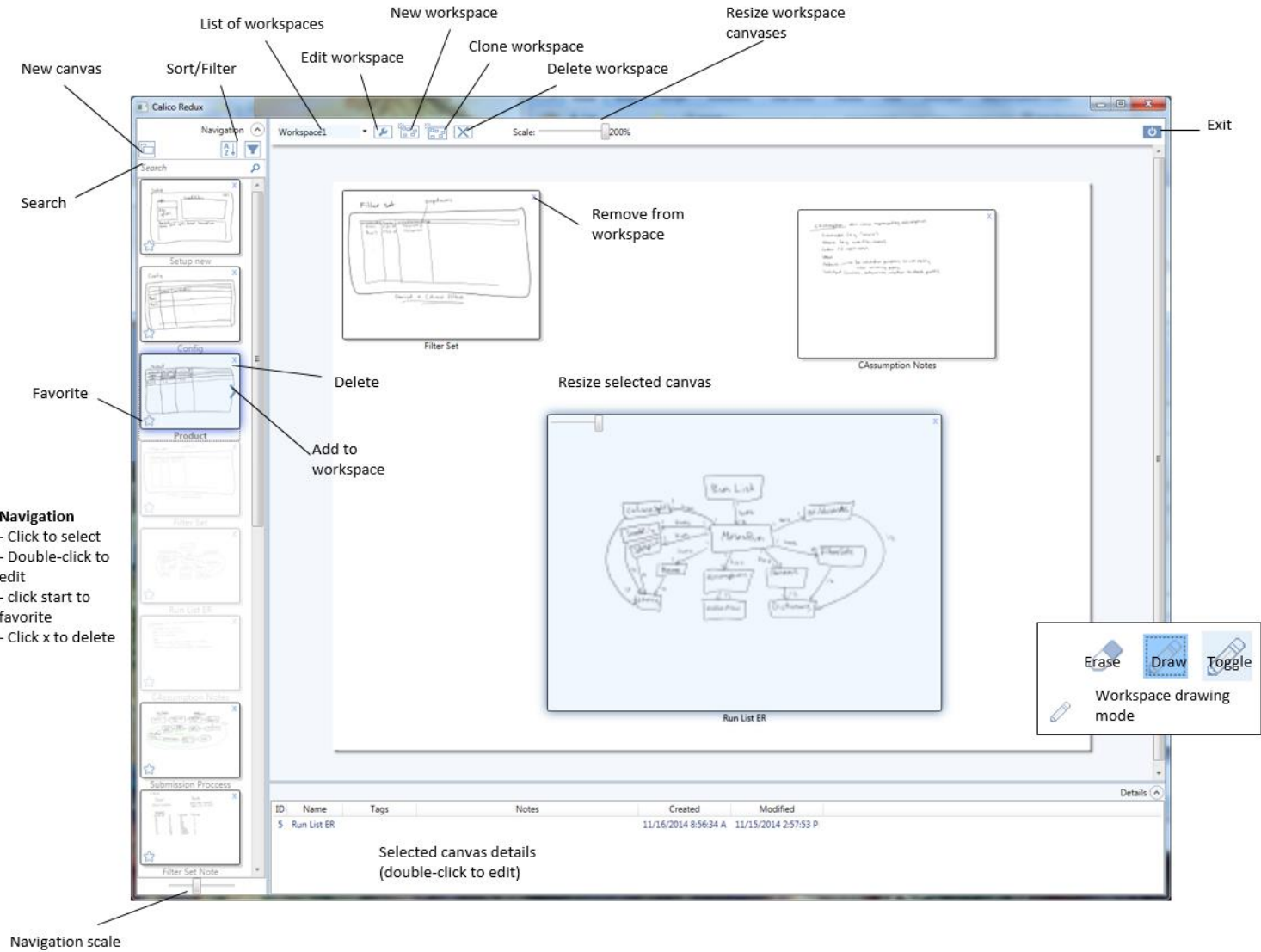
1. **Prepare for a meeting:** You are preparing for a meeting with your boss, where you will be giving a status update on the new Batch Submitter based on the last design session. Your boss is most interested in the UI, but would also like to see what you have decided on for the architecture.
  - a. Create two workspaces: one for UI sketches and one for architecture sketches
  - b. On the UI workspace, place the four indicated sketches out of the UI sketches
    - i. “Setup” is the most important, so you want your boss to pay the most attention to it; indicate this in some way
    - ii. Highlight the differences between “Product [current]” and “Product [proposed]” by drawing on the workspace
    - iii. Make a note on the “Setup” canvas explaining why it is important [wording provided]
  - c. On the architecture workspace, place the indicated canvas from the architecture sketches only, as this is the final design
    - i. Make this canvas as big as you can since it is the only one
    - ii. Draw attention to the three data structures in the canvas (highlighted on the sample)
2. **Debrief:** You have just finished the meeting with your boss, and it did not go the way you expected at all. Instead of approving all of your designs and recommending a raise for all your hard work, he asked you to change several things in the UI. You have a meeting with your team later to debrief, but you need to make some notes first.
  - a. Create a new workspace to hold all of your notes and changes.
  - b. Move the key UI sketch to the workspace for reference [see notes]
  - c. Create a marked-up version of this sketch. Mark in red all the things that your boss wants you to change, but do not modify the original sketch as it will be needed for future meetings [list and screenshots provided]
  - d. Sketch the new UI that your boss wants, based on these changes [screenshot provided].
  - e. Create one more sketch which provides a detailed view of the advanced options dialog, in which your boss showed particular interest. [screenshot provided]
  - f. In the workspace, show the relationship between the new advanced options dialog and the whole UI.
3. **Organize sketches after a design session:** You have just finished a lengthy design session with your team to incorporate feedback from your meeting with your boss. This has produced 20 new sketches. You want to organize them in a meaningful way, to make sure everyone can understand them later.
  - a. The sketches are of several different types, and some are in more than one category. Identify each listed sketch according to its type(s) (some sketches will not be categorized).
  - b. Identify the most important sketches in some way and move them all to a workspace representing the project’s current status [list provided]

- c. Create a workspace for each member of the design team (each user) and move the sketches they created to these workspaces.
4. **Pick up where you left off:** You are returning to work after a lovely vacation backpacking through Peru. You didn't check your email the whole time, so you have no idea what has been going on with the Batch Submitter project. Unfortunately, the rest of your team bought a lottery ticket in your absence, won the jackpot, and quit, so you are alone on the project. Fortunately, your team has been using Calico for their design sessions. Try to use their sketches to determine the status of your project.
  - a. Find the most recently modified sketch in each category (identified by tags).
  - b. Find the sketch that shows the current UI design approved by the steering committee.  
Note: this is not necessarily the one that was most recently changed.
  - c. Find out which user created the most sketches while you were away
  - d. Find out which designs were approved by the boss
5. **Be the boss:** A week later, you discover that your boss has coincidentally won a completely different lottery and quit. His boss has been so impressed by your work on the Batch Submitter that she decides to put you in charge of his whole department. As your first order of business, you need to hire a whole new team and train them. You decide to go back and organize the team's sketches from the new RAP-based Batch Submitter, which the new team will be taking over.
  - a. Organize the sketches however you see fit
  - b. Delete sketches that are obsolete or not useful

## Appendix B: Cheat Sheet

The following three pages were provided to participants to remind them of the available features and how to access them.

65



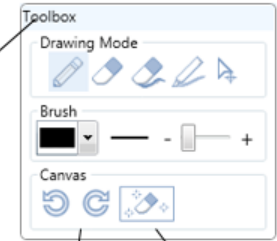
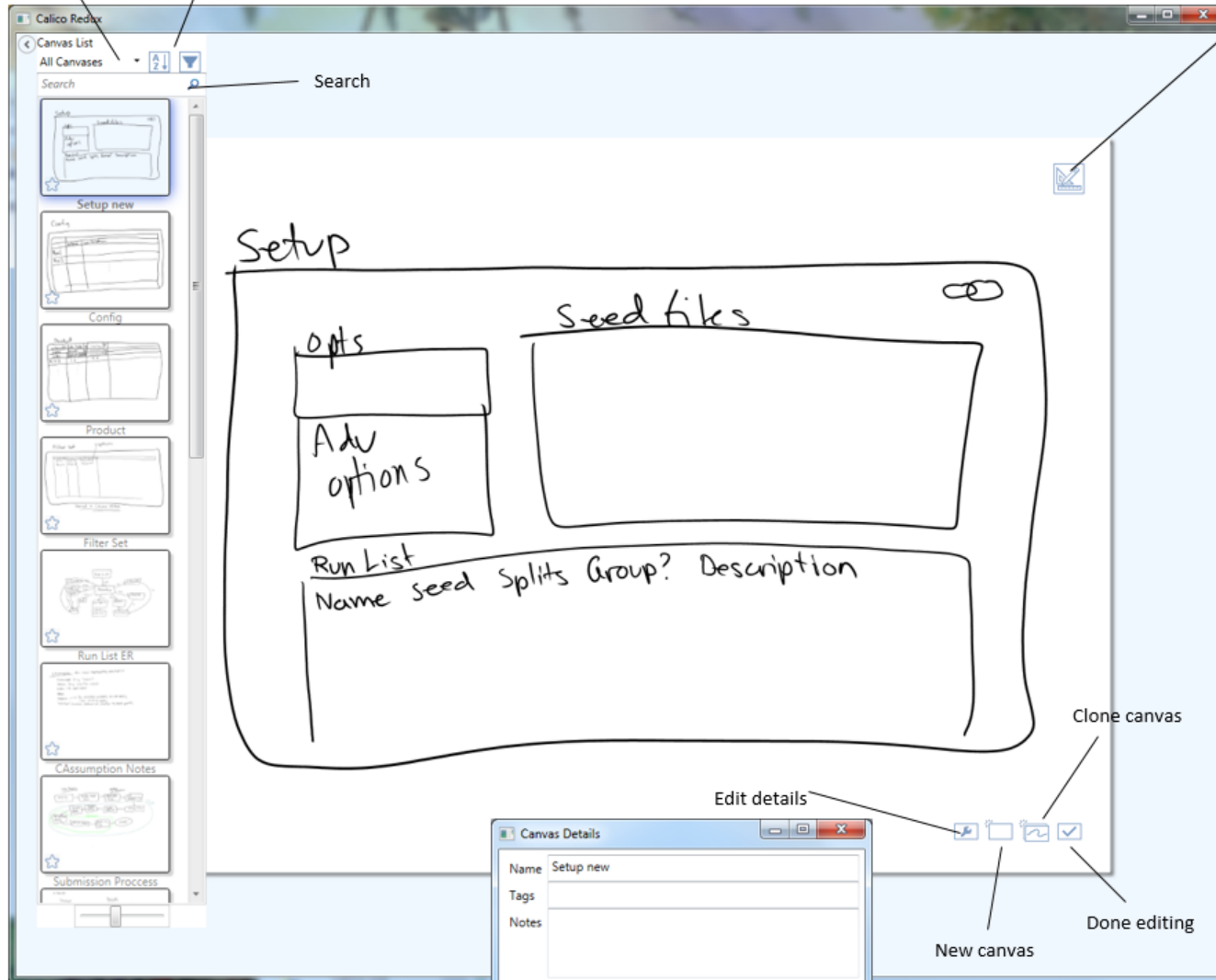
- Navigation**
- Click to select
  - Double-click to edit
  - click start to favorite
  - Click x to delete

Selected canvas details  
(double-click to edit)

List of workspaces

Sort/Filter

Search



Undo/redo

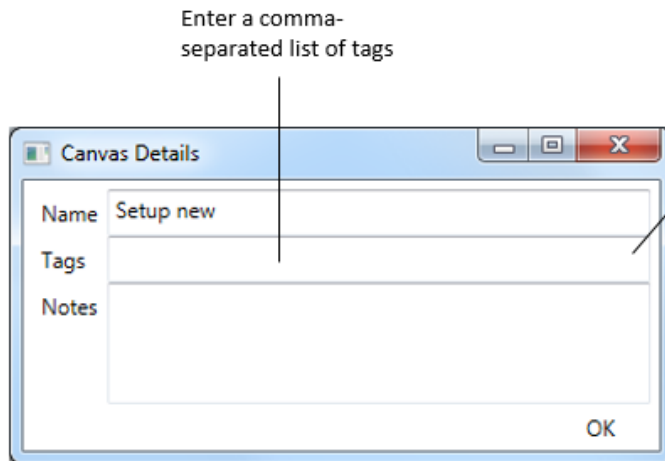
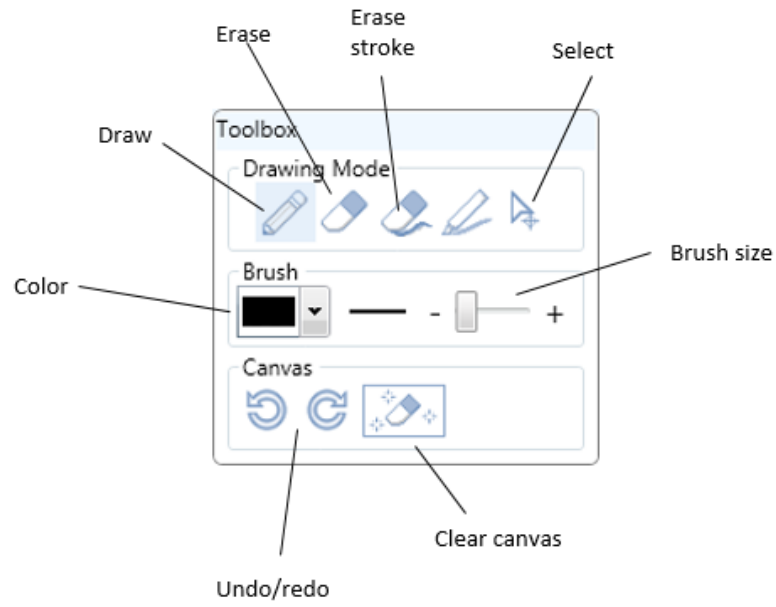
Clear canvas

Clone canvas

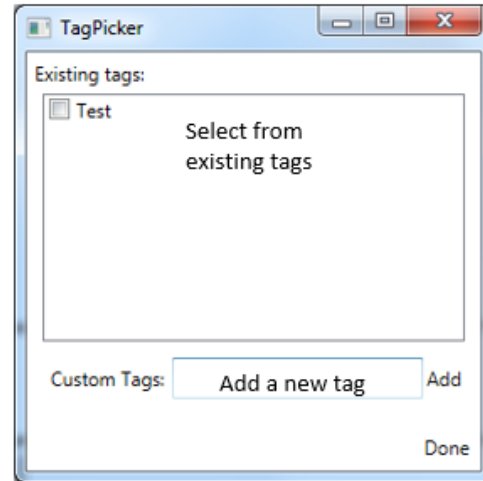
Edit details

New canvas

Done editing



Double-click



## Appendix C: Structured Task Rubric

Task	Expected action(s)
1a	New workspace + rename
1b	Select UI sketches and move
1bi	Make sketch bigger and/or draw on it
1bii	Use pen to note differences
1biii	Double-click detail record
1c	Select indicated canvas and move to arch WS
1ci	Use both canvas and global slider
1cii	Circle on workspace
2a	New workspace + rename
2b	Select and move
2c	Clone sketch, change pen color to red, note changes in red
2d	New sketch, change pen color to black, sketch new picture
2e	New sketch, sketch new picture
2f	Add new UI and advanced sketch to UI workspace, draw relationship on workspace
3a	Add tags to each sketch, use tag picker after adding custom tags. Sort by name to more easily find sketches.
3b	Favorite sketches
3c	Use filters to identify users and make a workspace for each. Select all canvases matching each filter and move then to the appropriate WS.
4a	Sort by modified date, filter on tags
4b	Use search box and check canvas note
4c	Filter on user and count % of sketches, or sort by user and select all, look in details
4d	Workspace called approved, tag = approved

## Appendix D: Interview Questions

For everyone:

1. Tell me about your overall experience with this tool?
2. Which feature did you find the most useful? What was useful about it?
3. Were there any features that you had trouble using? What made them hard to use?
4. Was there anything you thought was missing, or that you think would have made the tasks easier?
5. Which task did you find most difficult and why? Easiest?
6. Do you feel like the features of this tool made it easier to perform the tasks? Howso?
7. Were there any tasks that you think would have been particularly difficult to accomplish without this tool?
8. Can you think of any situations where this tool would have been useful in your work?

Situational:

- I noticed that you did not use [feature] at all. Why not?
  - Tell me about your approach to [task].
  - You seemed to have trouble with [task]. What made it difficult?
  - I noticed you referencing the cheat sheet a lot during [task]. What were you looking for?
- OR
- I noticed you did not reference the cheat sheet at all. Why not?