



# Internal research

Moderated usability tests conducted with internal Comcast employees to help gauge impressions and ease-of-use within the Ruby desktop app.

FEBRUARY - MARCH 2019

Please note: This version is not interactive. Below are instructions for the InVision version.

Jump to...

## Executive summary

An overview of the scope and methods of research

Executive summary ▶

## Finding categories

All findings have been categorized into themes for wholistic analysis

Iconography ▶

Status (availability) ▶

Workflow optimization ▶

User expectations ▶

## Tips on using this document

- **Navigate via the main navigation** (and sub navigations) at the top of the page
- **Find interactive points** by clicking anywhere on the page or pressing shift and looking for the blue boxes
- Research findings are **tagged by one of the icons** below for scanability



**Frustration** — a pain point observed



**User Impact** — an observation that impacted the user’s experience or expectation



**Success** — a highlight of success



## FINDINGS

# Users have learned behaviors and expectations

There is a flood of communications apps in the market that workers use on a daily basis. The way people currently use various communications apps, for work and personal life, greatly impacts the way they **want to use** new tools and their **expectations** of what those tools **should be capable of**.

The research findings are broken into four main categories:

### Iconography

Some visual languages need to be more clearly identified

### Workflow optimization

The dashboard should be a place for a user to optimize their workflow

### Status (availability)

Availability is a way for people to set expectations for their coworkers

### User expectations

Workers use a variety of communications tools and have expectations around what they should and shouldn't do



## PROJECT RECAP

### Project scope

This project utilized internal resources to test InVision proof-of-concept demos of the Ruby desktop application. All research was conducted within 1-2 weeks time.

DISCUSSIONS WITH

7

Comcast employees

### Project purpose

**We want to...** understand users' expectations and usages of communications apps

**So that...** we are designing the right features, with the right purposes, in a sensible order

**Because...** users that can easily engage with and utilize a communications app will become lifetime customers and potentially spread word to new customers overtime.

### Success means...

- More direction for the prioritization and roadmapping of features
- A better understanding of user expectations for features to influence design approach
- Positive customer feedback upon initial external app deployment

### Research method

Proof-of-concept prototype tests (InVision)

Our team created InVision demo prototypes that represented the look and feel of the app and mimicked the app's functionality. Users were provided task prompts and scenarios to walkthrough and discuss utilizing the demo prototypes. Detailed notes were taken during the sessions and then organized and synthesized to be leveraged for app design improvements and next steps.

## ICONOGRAPHY

Users don’t immediately recognize certain icons without text associations.



### Feedback

**6 out of 7** users confused the feedback icon for something else:

- chat / messages
- support
- FAQ

#### NEW IMPLEMENTATION

##### We aligned the icon to match the ActiveCore ‘feedback’ icon

This gave us the opportunity to better align the designs of ActiveCore and Ruby, while also creating more distinction between ‘messages’ and ‘feedback’.



### Eject

**5 out of 7** users could not identify a specific action for this icon:



*I have no idea what that icon is.* — Hugo Fonseca

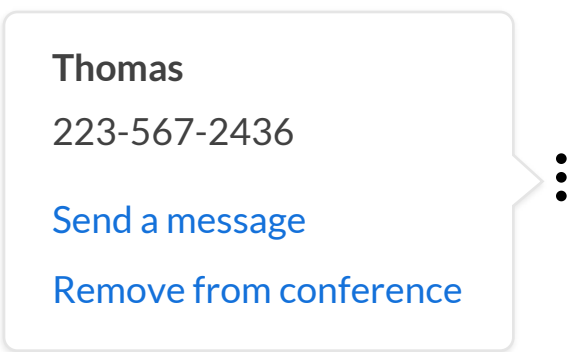


*Who has left the conference? They’re out of office? That is not clear.* — Elva Chu

#### NEW IMPLEMENTATION

##### We created a dropdown menu for actions regarding participants

Most users aren’t familiar with the ability to “eject” a participant. Creating an action menu allows us to utilize words instead of icons and also introduces scability to include participant information and actions such as ‘send a message’.



### Audio & details

**3 out of 7** users thought the ‘audio & details’ section could be more clear:



*Be a little bit more specific.* — Andrew Topping

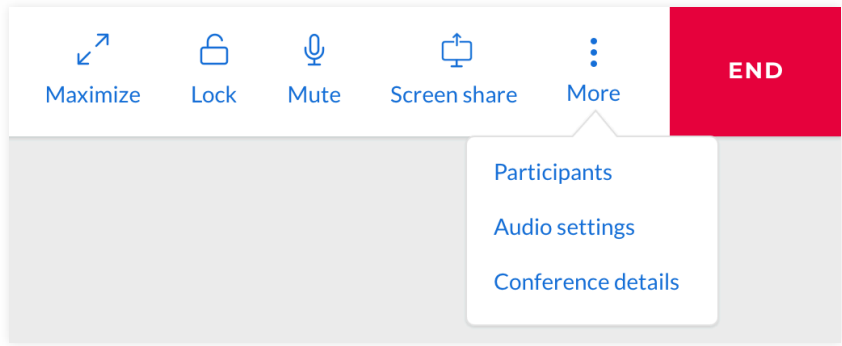


*Would hope this was for sound controls and settings.* — Joseph Danks

#### NEW IMPLEMENTATION

##### We’ve created a dropdown menu for secondary actions

Keeping the most important actions accessible is key for task focus. Utilizing the ‘action dropdown’ pattern, we’re able to provide users more focus on the most commonly used icons, while also allowing for scability and clarity.





01

The location of status is expected, but the selection process should be more intuitive

Users easily found the dropdown status under the account icon, but consistently commented on the idea of an ‘available’ setting. Without the setting of ‘available’, it’s not as intuitive to turn on/off other statuses, such as do not disturb and call forwarding.

- Most users felt that status was a useful setting, but wanted to be able to **set ‘available’** to make it clear that no other settings were currently on.
- If I could see ‘status available’ then I’d know I could change it.*  
— Hugo Fonseca
- Some users wanted **confirmation** of what their status would look like to other users.
- Status bar didn’t look like a status bar. It should always be present. That’s how the Skype indicator works.*  
— Shane Monaco
- It would be helpful if I can see how other people see me.*  
— Elva Chu

NEW IMPLEMENTATION

**Include ‘Available’ as the default setting option**  
  
Revisit the original design to include ‘available’ as the default setting option.  
  
Remove ‘toggle’ functionality to eliminate confusion as to what setting is on.  
  
In the future, consider including ‘away’ as a status option.

02

Setting a status is a way to set expectations for other app users and coworkers

The purpose of a status feature is to not only control other app functionalities, but also to inform other users of your availability. When users have information around your availability it provides them with the ability to priotize and time manage their own work depending on when you may be able to respond.

- Similarly to notifications, status plays a role in a user’s **ability to prioritize** their tasks and manage their time. If a user knows they need an answer from someone, but that person is unavailable, they then have the ability to reprioritize other work in order to create efficiency.
- I always check someone’s availability before reaching out so I know I’m not interrupting them.*  
— Elva Chu
- It would make sense for another app user to see what I’m on do not disturb mode for chat purposes so they don’t think I’m ignoring them.*  
— Hank Garrett (Pilot)

RECOMMENDATIONS

**Prioritize status appearance for app users**  
  
Surface someone’s set status to other Ruby app users to set expectations.  
  
Consider including an ‘away’ status.

03

Provide more clarity and task affordance around setting statuses

Overall, users generally understood the functionality of ‘do not disturb’, but adding clarity in a FTUE experience or instructional/help text would provide more confidence when setting the feature. Other functionality to consider for status is the ideas of treating calls and messages separately and scheduling.

- Users generally understood to use ‘do not disturb’ when they were busy or unavailable, but many **users weren’t clear** as to whether it would prohibit calls and messages from coming through, stop all notifications, or just control one or other.
- I always check someone’s availability before reaching out so I know I’m not interrupting them.*  
— Elva Chu
- Some users wanted to **handle calls and messages differently**. Users also mentioned the **ability to schedule** different statuses depending on their meetings and/or office/business hours.

RECOMMENDATIONS

**Provide clarity and control around status in ‘Settings’**  
  
Consider adding more help text for users when they’re in the moment of setting a status.  
  
Approach calls and messages differently; users utilize the two communications in different ways so they should be handled differently in every context, including Recents, Quickstart, Notifications, and Status.  
  
In the future, do more research around what our external audience uses status for in order to assess if scheduling is a useful functionality.









USER EXPECTATIONS

Users have access to many communications tools, which influences their expectations for Ruby features.


01

Because users interact with communications tools daily, they have very clear feature expectations

When users expect a feature to exist, but they can't find the feature, it creates hesitation and confusion. The user attempts to look for the feature or starts to create a workaround to simulate what they expected to be able to accomplish.

-  Many users expected they would be able to **place a 3-way call** when prompted with needing to dial-in other users.
-  *We use a lot of 3-way calling because we can easily add people in from their desk.*  
— Elva Chu
-  Many users wanted the ability to **chat a conference participant** in private **without leaving** the conference area.
-  *I would be afraid to leave this screen and go to messaging because I might be missing something in the meeting and if I'm sharing my screen I want to know what everyone else is seeing.*  
— Hank Garrett (Pilot)
-  Many users expected to be able to **invite participants** to a **scheduled conference**.
-  *Intuitively, I'd look for a calendar option because I probably would've scheduled a meeting.*  
— Hugo Fonseca

RECOMMENDATIONS

 **Use findings like this to determine feature priority on the roadmap**

Perform technical research to understand what the backend **currently** has the ability to support.








Prioritize features for **small wins first**, then tackle larger lifts in the future.

Consider a feature-sorting workshop to **understand what users want and expect** out of an app like Ruby.


02

Many of the people interviewed were familiar with most of the features and had new ideas

Our interviewed users were very savvy and used communications tools on a daily basis for personal and work. They intuitively understood most of the features that were presented to them, and also had suggestions for what else they would like to see to differentiate the app.

-  The red indicator for notifications was very familiar to most users. They could easily see that things needed to be addressed in the marked areas.
-  *This is on par with what I'm used to.*  
— Shane Monaco
-  Every user has a different workflow, so the ability to customize the dashboard to slightly cater to their personal workflow might be beneficial.
-  *If you had the ability to change your layout that would set it apart.*  
— Joseph Danks
-  Some users like system-managed options and others prefer manually-managed options. When designing features, these two ways of working should be considered to create an optimal experience for a broad audience.
-  *If it was someone I frequently called I'd expect them to show up in the top 6.*  
— Nidhi Desai
-  *I'd like to be able to create a team favorite [...] kind of like Myspace top friends.*  
— Joseph Danks

RECOMMENDATIONS

 **When creating features, assess 'system-managed' vs. 'user-managed' functionality**

Some features may benefit from system-triggered management with the ability to override (favorites)

Other features would benefit from user-managed settings (notifications, quickstart)

Take into consideration how each feature is going to be used and design a workflow that creates the most ease-of-use

 **Move toward personalization and customization in the future**

Consider customizable layouts with display options (think Gmail layouts - Default, Comfortable, Compact)

Consider the option to display/hide certain features (notifications summary on dashboard)

Consider the option to display/hide certain content (include/don't include messages in Recents)

 **Continue to leverage learned user-patterns from other tools**

User-adoption will happen more quickly and easily if design patterns from common tools are leveraged

Users have specific mental models created by other tools, introducing new flows creates hesitation

Adopting existing patterns creates a familiarity that people are comfortable with



