

How an Engineering Team Uses Quip

If you asked our engineers what tools they used at previous companies to get their work done, you'd get answers from wikis, to IRC, to Google Docs, to post-it notes on whiteboards. Every company has its own cobbled-together collection of products — often a mix of internal and external tools — to help engineers be productive. What you'll also hear is that there was a huge amount of inconsistency, even at the same company. One engineering team would communicate via email, while another did their coordination using a wiki page. Some design docs are sent out as Word Files, others kept in Google Docs. There was a lot of complexity to manage — a lot of "work" involved in just getting work done.

At Quip, things have been different. We use Quip for virtually every part of planning, work tracking, and communication. While we still rely on products like [GitHub](#) and [Zendesk](#) for our bug database and outbound support, Quip itself is simple and flexible enough to have replaced a host of other tools to support what we do everyday.

Every stage from design to launch

Our engineers rely on Quip to build Quip. And they've found one of the most distinctive aspects of Quip — that each document has a corresponding messaging thread — to be extremely powerful. Nowhere is this more evident than when a new feature or piece of functionality is in the design stage. Someone will create a design doc and rather than posting it somewhere or sending it out for review and then trying to make sense of the comments that dribble in via email, he'll simply add it to the Design Docs folder, which automatically shares it with the whole team and opens the door for collaboration and discussion.

Engineering | Design Docs - Shared with 16 people

Quip Folders

Overview

In its current form, Quip enables sharing much like e-mail: you must list every recipient individually for every thread, and every thread in your inbox has equal priority. Likewise, you have no way to archive or organize your documents. Documents show up in an endless stream in your desktop ordered by the last time you edited them.

Our goal with folders is to address these usability issues with a single concept:

- **Folders are the way you share with groups.** You make a folder, add people to the folder, and any document you put in the folder is available to the whole group without having to list people individually.
- **Folders are the way you organize your documents.** Your Desktop is the set of documents you are actively working on, and you can file away the documents you aren't

Notification Control

Summary

We want to enable people to control the frequency of Quip notifications they receive in a granular way. We will introduce a standard set of "notification levels" which can be applied to individual folders and threads. On any thread and folder, you can lower and raise the notification level to reduce or increase the number of notifications you receive.

Notification Levels

There are three notification levels:

- **Notify** (formerly Push) - Receive push notifications to my phone / email account about this folder/document/thread in real-time.
- **Inbox only** (formerly Passive) - See

Desktop/Archive Button Behavior

Goals

1. Archiving or adding to the desktop does not affect the shared state or participants of folders or threads
2. Only shared folders should have symlinks

"Desktop" refers to the "Add / Move to Desktop" button. "Archive" refers to the "Archive" button. "add-folder", "remove-folder" and "move-into" refer to the doc_web HTTP handlers.

THREAD

- On desktop
 - Archive:
 - add-folder(archive)
 - remove-folder(desktop)
 - Desktop: `add`

Invitation and Sign-Up Architecture

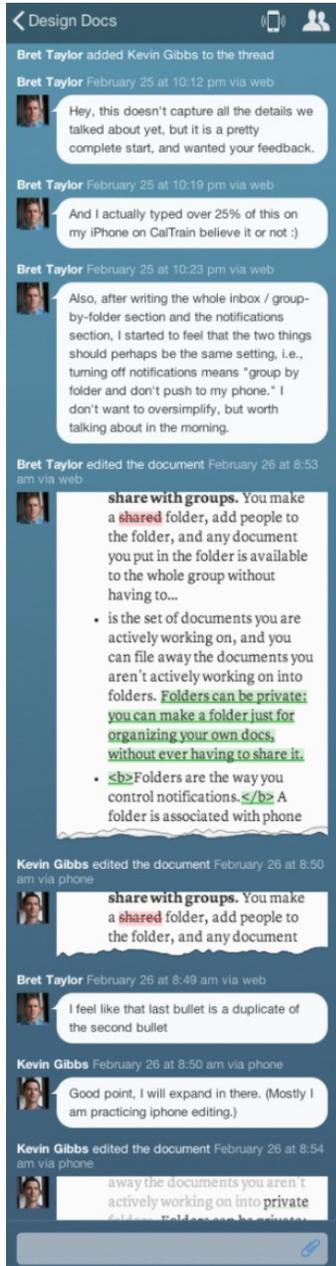
Overview

Unless a user has fully signed up for the service, they are not present in the User table. Likewise, while we will allow to people to "add" non-Quip users to threads and folders, we are going to store them complete separately so that our code (in general) does not need to handle special case users. A few limited views like the thread sharing dialog will integrate the list of invited participants into the real list of participants to present in an integrated view, but most of our codebase will be able to ignore the concept of invitations or "pending" users in general.

Users in Quip can be in five states:

- **Pending Device** - You have installed an app on this phone. [https://quip.com/invite/...](#)

Revisions can happen directly on the document, sometimes taking place quickly as a result of the instantaneous feedback, and when the document is finalized, there is also a log of all the changes and discussions along the way.



Quip Folders

Overview

In its current form, Quip enables sharing much like email: you must list every recipient individually for every thread, and every thread in your inbox has equal priority. Likewise, you have no way to archive or organize your documents. Documents show up in an endless stream in your desktop ordered by the last time you edited them.

Our goal with folders is to address these usability issues with a single concept:

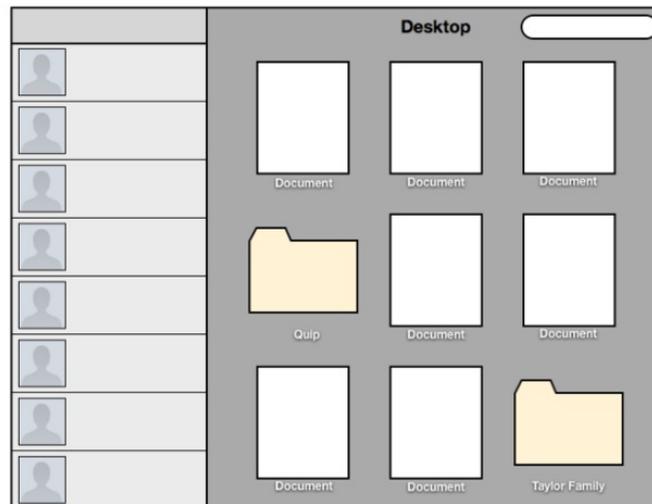
- **Folders are the way you share with groups.** You make a folder, add people to the folder, and any document you put in the folder is available to the whole group without having to list people individually.
- **Folders are the way you organize your documents.** Your Desktop is the set of documents you are actively working on, and you can file away the documents you aren't actively working on into private folders.
- **Folders are the way you control notifications.** A folder is associated with phone and email notification preferences. You control your notifications preferences for a document by placing it in a folder. For example, you will stop receiving push notifications for a document if you move it from your Desktop to your "Archive" folder. You might start receiving push notifications if you add a document from a large group folder to your Desktop.

We are not introducing a separate concept of a "group." A shared folder is a group. The Quip company "group" is just the "Quip" folder, and it can have sub-folders that reflect our internal organization (e.g., "HR" or "Engineering").

Architecture

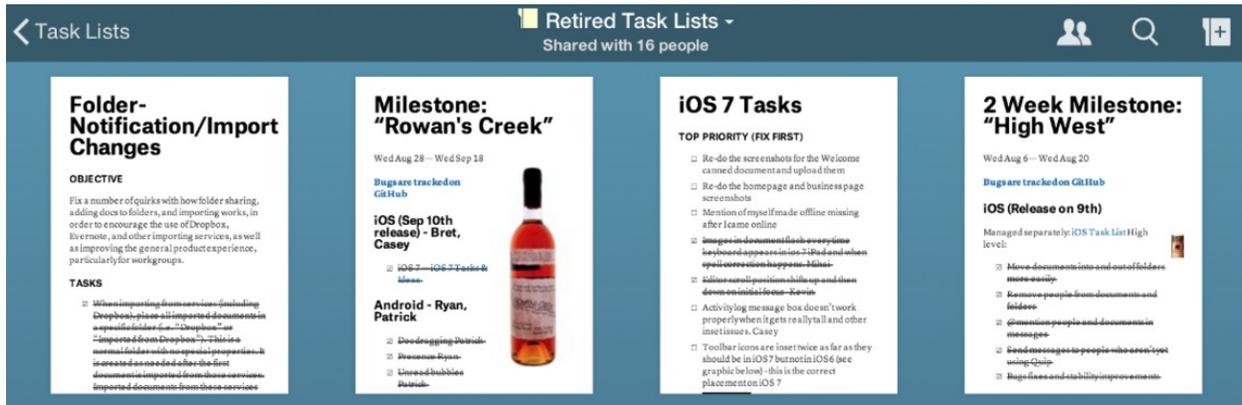
BROWSING

Your Desktop is your own, personal root folder (think your "My Documents" folder in Windows or your Dropbox folder). You can arrange all the documents and folders in your desktop just like you can arrange icons on your iOS home screen, i.e., position is sticky. Folders are mixed in with documents.



When a new document is added to a folder, it shows up at the top of a folder with a label indicating it is new, pushing the older documents down. This is essentially the inverse of iOS, where new icons show up at the end. The "New" label stays until you take an action on it (view it, move it, etc). Documents don't change order automatically once they show up in a folder - everything is explicitly controlled by the user.

Once a design has been nailed down, the engineer working on the feature creates a list of the specific implementation tasks required. Each major feature we work on has its own task list. And because Quip offers the ability to quickly and easily turn text into a checklist, each task list is simply its own document, which means that the engineers don't have to go to a separate product for project management.



The team uses @mentions to assign owners to sections or items in the list. People can see at-a-glance who is responsible for what. The checklist format also makes it really easy to follow along on progress. If one engineer is blocked on something that another person needs to implement first, they can immediately see when it gets done. (It's also very helpful for those of us working on customer and press communications — to see when something has been done or fixed, or to get a sense for how we're tracking on a deadline.)



In addition to major features and launches, the engineering efforts at Quip are broken down into a series of three-week milestones. There is a corresponding task list for each milestone. Small tasks are listed directly in the milestone's task list. For more complicated features, like our full Android application, there will be an @mention to the corresponding feature's task list.

iOS (Sep 10th release) - ●Bret , ●Casey

iOS 7 - 📁 iOS 7 Tasks -

Android - ●Ryan , ●Patrick

Doc dragging - ●Patrick -

Presence - ●Ryan -

Unread bubbles - ●Patrick -

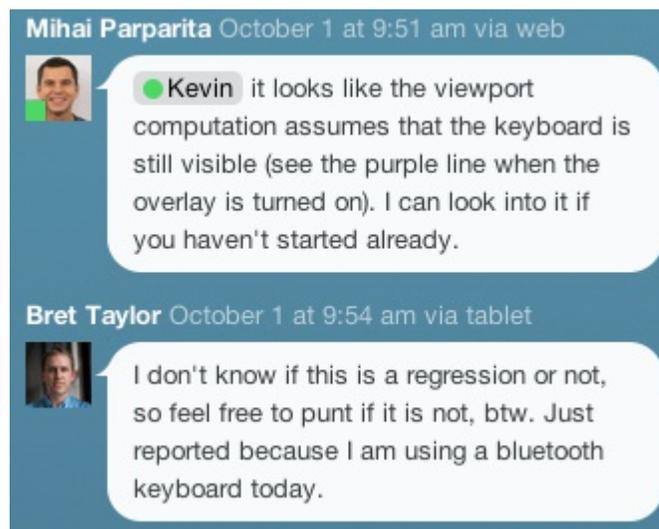
Address book import - ●Ryan -

@mentions in Editor

Get editor working - ●Kevin

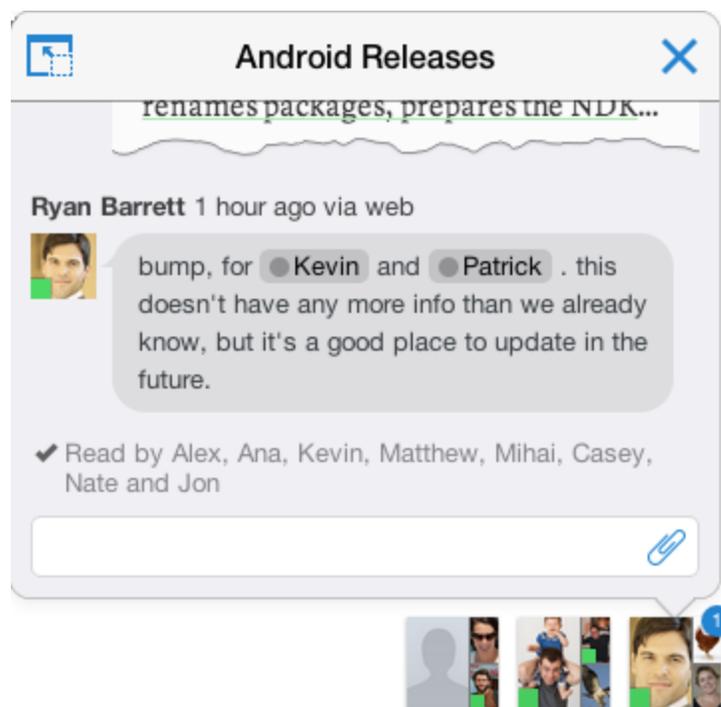
📁 Android 1.0 Launch Task List

The task list serves as the project management tool, but naturally also becomes the place where bugs get surfaced. We rely on GitHub for tracking random bugs we encounter, but when it's an issue in an area where another engineer is actively working, it's more productive to surface it in relevant the task list instead.



This makes it easy to see if regressions are known in areas that are rapidly changing, and because of the natural conversations that arise when the messaging thread is part of the document, the task list itself becomes the perfect place for a quick discussion of the correct behavior and how to fix it.

When it's time to actually launch the feature, Quip becomes the best tool for release coordination as well. Conversations happen on the task list, or in the Engineering thread that everyone is part of. Everybody is usually already working on documents within Quip, so they can easily follow along on conversations via notifications and [chat tabs](#), without having to constantly switch back and forth to another product. And with several of our team members having long commutes via public transportation, it means they can stay in the loop from anywhere, and keep their task lists up to date even without internet access.



When a task list is finished, it gets filed into the Retired Task Lists folder, which serves as an archive of the implementation of every feature we've built, complete with details of exactly who did what, in what order, and any issues that arose for discussion.

Intranet, research repository, design library and more

In addition to using Quip day-to-day for managing a launch or splitting up work on a feature, we've also come to rely on Quip for practically everything else. We keep a single document with the agenda and notes from our weekly engineering meetings. Before each one, folks will contribute discussion topics, which bubbles the document up to the top of everyone's Inbox. Weekly user stats and other relevant graphs are also added, which means that this document becomes an ongoing log of our vital statistics, as well as what we worked on, each week.



OCT 1ST, 2013

Discussion topics

- SOC review - Kevin
- Android Launch - Kevin
- Redesign Demo - Patrick
- iOS release - Bret
- web iOS 7 style UI release plans - Mihai - Matthew
- your topics here*

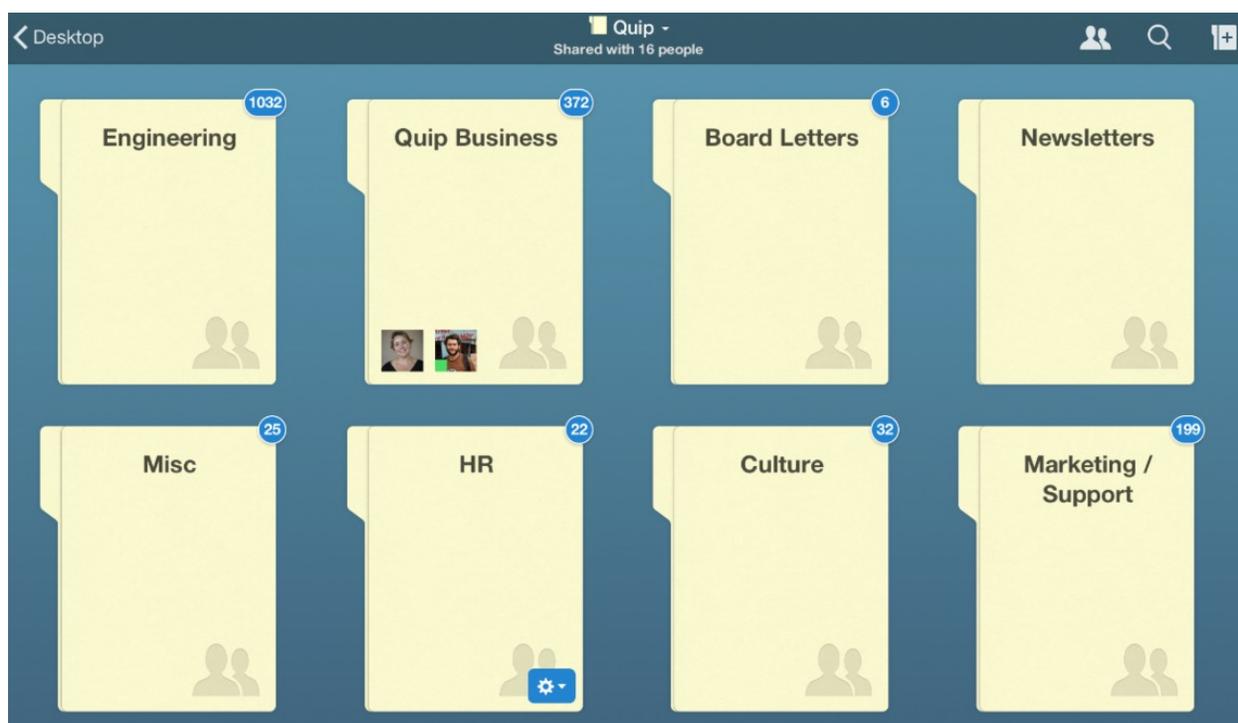
We also heavily depend on [folders](#) (and a few folks who like to keep all our folders and documents well-organized). There's a "Research" folder with documents for tricky tasks like how to run the server after upgrading to Mavericks, or tables of what Android devices are most popular. This folder isn't used every day, but it's there as a permanent store of shared knowledge.

There's another folder where our designer, Matt, uploads screenshots and other files, and new email copy gets hammered out. Quip makes it easy to attach files to threads and to insert them into documents, which makes it easy for our engineers to have access to whatever assets they need for the site at all times.



Quip has become not only our intranet, where all company materials are easily accessible via search, but also our mailing lists. Everyone was recently invited to the company Halloween party when a document containing all the details was shared with them.

Quip has emerged as the way everyone keeps themselves up-to-date on different parts of the company, be it the next newsletter that's going to be sent to customers, or the newest forms for benefits enrollment. And it's the way we communicate with one another — we don't use email internally.



Of course, there are only 10 engineers at Quip, and 13 of us total. If we were a much larger company, individual teams would probably use Quip the way we do right now as a company. That's why our engineering team has come to be some of our best test users — seeing how successfully they're using the product has demonstrated Quip's ability to seamlessly fulfill many of an organization's various needs.

The ironic thing is that if you asked one of our engineers how often they used "documents" before Quip, they'd probably say they didn't. Before Quip, they juggled a litany of tools to serve different purposes, usually alongside mountains of email. In Quip, our engineers have found (and built) a simple, yet flexible product that makes it easier for them to keep making great products.