Chapter 5

Evaluation

A report on our personal experiences using Snapstore for the past few months is included below. Ideas for future evaluations can be found in section 7.2.

5.1 Advantages

Easy to Get Started

Snapstore fulfills its goal of being easy to setup and use right away for a new user. Users log in and immediately have access to its more basic functionality. Snapshots of their files are automatically created, and everything is being saved to a local and an upstream repository. No configuration or learning of Snapstore is required for any of these benefits.

Automatic Saving

Because snapshots are automatically taken, there's no need to manually save your work. Instead of having to stop and save your work from the Snapstore application, simply using a typical hot-key for save in your editor will create a snapshot.

Working with Branches

Branch creation and switching between branches are both one-click operations. Because the file system changes with respect to the current branch, there is never any confusion about which branch's file the user is working on.

Branches also mesh well with the user's current knowledge of their filesystem; switching branches also switches the files in their Snapstore folder. There's no extra idea for the user to grasp when changing between independent lines.

Reversion

Reverting to a previous snapshot is another, easy one-click operation. It edits the user's filesystem so that their computer and Snapstore are always in agreement about the contents of files. Furthermore, the creation of a new snapshot for a file reversion is an easy concept to understand as a new user. The operational principle behind a snapshot holds: whenever an edit to a file is saved to disk, a new snapshot is created.

File and Snapshot Graph Navigation

Snapstore's built-in file navigation makes it simple to search through all of the files within the Snapstore folder. This makes it easy to locate snapshot graphs, and having all the functionality of Snapstore in one place is convenient.

Furthermore, the intuitive presentation of the snapshot graph makes it easy to navigate for the user. Although this graph may be cluttered (an issue explored in section 5.2), the graph is intuitive and the relationship between snapshots (nodes) in the graph makes sense.

5.2 Issues

Snapshot Graph Search

The frequency with which snapshots are taken results in a cluttered snapshot graph. The snapshot graph loses meaning when there are too many snapshots to navigate it

effectively, and it makes searching through the snapshot graph inefficient. This is not ideal for making use of Snapstore's most fundamental concept, the snapshot.

Cloned Branch Snapshot Selection

During the cloning of a branch, selecting which snapshots to copy to the clone is difficult for the same reason: there are too many snapshots. Choosing subsets from a snapshot graph is difficult, even though choosing the entire snapshot graph would not be. Though we don't know if users will, in fact, want to clone subsets of snapshot graphs, Snapstore does not currently allow them to do so efficiently.

Operations for Each File in Interface

For each file in the UI, there is both a file icon and a file name, but the function that clicking the file name performs is not clear. It opens the snapshot graph for that file. There are no affordances for this functionality, and it is quite possible that a new user would never be able to figure it out.

Lack of Native Operating System Support

Snapstore, unlike Dropbox, does not have any support for native applications such as being able to used from the native file manager, for example. This lack of support is annoying as a user trying to access Snapstore's functionality while completing other tasks.

A user could be exploring their filesystem from their task manager and want to observe a file's snapshot graph. However, they would have to open up Snapstore to do so. This inefficiency is not attractive to the prospective Snapstore user.

5.2.1 Conclusions

Snapstore does well at its two goals. First, it is easy to use for new users; with its opt-in complexity strategy, it's easy for users to get started right away with simpler,

file syncing tasks. Second, it delivers an internal conceptual model that is easy for the user to understand.

Snapstore does have room for improvement. These improvements deal mostly with the presentation of Snapstore's concepts to the user and with the interface in general. Snapstore needs work refining the interface to make the concepts easier to navigate. This is a crucial step in Snapstore's development because without it, the underlying conceptual model of Snapstore cannot become the mental model of the end user. It is important, though, throughout this redesign, that the conceptual model is never compromised for the sake of the interface. The conceptual design and the interface design must complement each other in presenting the concepts to the user.