Extensibility

Like other version control software, Mercurial has the ability to have custom extensions written for it. Being able to create your own extensions adds a level of customizability to Mercurial which also increases its functionality to levels as high as you want it to be. If a user wants to add new commands, wrap an existing command, function or repository in Mercurial, or add new repository types, then extensions will allow them too. A common example of adding new functions to a Mercurial repository is the use of a "reposetup" function. This function wraps a repository in a subclass so that extensions can greater modify the behaviour of the repository. An example of adding a new repository type can be seen in the "hgsubversion" project. This project allows repositories from Subversion (another VCS from Apache) to be managed in Mercurial. This can help avoid possible difficulties of migrating a repository from one VCS to another or simply to help centralize all of a team's code in one VCS.

Mercurial also offers support for continuous integration systems through the use of hooks. Mozilla has implemented these hooks in order to create a log of all pushes to their repositories. This technique creates a history of changes to a repository which helps developers in many ways. Such as helping them figure out the origin of a bug in their code by aligning the date of a push and the time when the bug began happening.

Lessons Learned

The development of Mercurial has served as a sort of showcase of the strengths and weaknesses of Python. Mercurial makes the most of Python's extensibility yet also inherits its relatively slow compiling time.