Why GitHub? (under development)

One answer to the “why GitHub”? question is that it’s cool. Learning GitHub now will be like learning how to markup html in 1992 - you’ll be on the leading edge of a big change that is coming. While the way you’re going to use GitHub will probably change dramatically in the next few years, this experience will put you in a position to better understand and take advantage of that change.

But there is a practical, immediate application for GitHub, and it’s how it lets you collaborate with a large group of people unknown to you in a manageable way.

What’s Wrong with Current Systems?

If you want to collaborate with people on a document you’ve written, why not use a Word document that you email around, where you can use track changes to know who added what? Or how about a Google doc that anyone can change, or just comment on?

It doesn’t take a lot to imagine this scenario: the creator of a document uploads it to a shared area on a server and then informs everyone where it is (probably by email). Let’s say it’s details about a company picnic - location, time, who’s bringing what, etc. And let’s assume I want to let people add stuff (like what they’re bringing), so the Google doc allows anyone in the company to change it and the Word doc is set up for track changes.

For a Word doc, people download it to their local computer. Someone may add that they’re bringing some food item and lawn darts. Another person notes that the original location has the wrong address and another person asks a question about the time. All these changed Word docs get emailed back to me and I then either accept them into the master document or respond to the person who sent them. When the file is updated, I then re-upload it and email everyone to tell them the file has been changed.

If it’s a Google doc, it stays on the server - but without a track-changes feature, it’s not easy for me to see who changed what, and I can’t reject only one change but instead have to go back through all the changes in sequence to undo a change.

These approaches do work pretty well for cases where you have a small number of people who you know you want to include as collaborators. But what if we’re talking about a different kind of collaboration where you think there are people you don’t know who might be able to add value to the document. However, if you open up the document to their contributions, the size of the group might get so big that you can no longer manage their contributions with emails and shared documents. If I put the document on a website and let people make changes to it (whether they download the file over FTP, work on it and upload it when done; or edit the document in their browser as with Wikipedia), there’s the potential for edit conflicts where two or more people check-out a document at the same time. Some of the ways around this are lock-outs (where whoever downloads a document locks-out other potential users until they’re done), best-practices and protocols (e.g., with Wikipedia) or manual coordination (how Linux used to be run).

An alternative to all that is to take advantage of all the value that exists outside your own network by sharing openly the work you’ve done, invite an undefined group of people to collaborate to make it better, and even allow some of those people to take what you’ve done and do something else with it (giving me credit for the work you have done prior to that).

What GitHub Can Do?
from
http://aaronparecki.com/articles/2013/02/28/2/using-github-to-improve-communication-in-an-organization

manage text documents

use "Issues" to replace email discussions.

Track changes: Github provides a central location for the current state of a document, and clearly shows changes and who made them.

Issues: a log of a discussion with the entire history on a single page.

Uses email for notification rather than to deliver content

Get another user’s attention by mentioning them in an issue

Get a complete audit trail of all changes and discussions, and who was involved in certain decisions.

Search through issues

http://www.wired.co.uk/news/archive/2013-09/02/github-mainstream

GitHub's big innovation is the "pull request." It's what you do after forking something - an electronic note that GitHub sends a software developer, saying, "Hey, I was checking out your project and I found a way to make it better. Look here and you can see what I've changed; press this button and the changes will become part of your project." The pull request makes it easy for anybody to fix a typo in a document, or a bug in a software program, or propose new language to a legal document.