Case study of GitHub

1) Who does the developement?

→ The Github service was developed by chris Wanstrath, P.J. Hyett, Tom Perston werner and scott chacon using Ruby on rails and started in february 2008. The company GitHub, Inc. has existed since 2007 and is located in san francisco.

2) How it is organized?

→ GitHub project board's give you a platform to visualize and organize cards into different colums. There are two types of project bords available Repository Board's for use in a single repository organization boards for us in a GitHub organization across multiple repository.

3) How is it licensed?

→ Applying a license to a repository with an existing license. The license picker is only available when you create a new project on GitHub. You can manually add a license using the browser. ~~for~~ licensed caches the licenses of dependencies and checks their status. licensed is available as a Ruby gem for Ruby environments

4) How is source code managed?
→ Source code management enables coordination, sharing and collaboration across an entire software developement team. Github supports terms to rapidely collaborate and iterate on new features and delivery business value. Source code management is where developement team sharing and collaboration.

5) How are stable releases done?
→ Information about recent features added in ~~ott~~ Github releases can be found on the Github project releases page In the interface, by adding a releases not to an existing Git tag. usin the releases Github tag. New features are regularly released to Github saas. The self-manangeed released is a semever versioned package of features that are already released.

6) What communication methodes are used?
→ A somewhat wide collaction of various kernelmode - user mode comunication methods in one ~~&~~ repository. Github shares best practices for making remote work par of when comes to Github's asynchronous communiction.

Github is the best place to communicate about them.
1> code review
2> pull request
3> Issue tracking
4) Notification
5) Wiki.
6) Webhooks
7) Gists.

7) How are bugs tracked?
→) When Github~~s~~ support recives a bug report, we have too goals understanding an evaluation. The whole process can be summarized~~ed~~. Github extension for visual studios: contribute to github by creating an account. Integrate instabug with Github to track bugs ~~isd~~ inside your projects while maintaining your existi~~st~~ng workflow.

8) How does it interact with other project?
→ Whe we fork a project, Github will make a copy of the project that is entierly yours; it lives if other people have pulled it down and clone more work on it. In the world of software projects it is invitable that we will find. And ~~mo~~ as more and more projects switch to Git and increasing...

9) What has the project documented about itself?
-> Good documentation is key to the success of any project. Making documentation accessible enables people to learn about a projects, making it easy to update ensures that documentation stays relevant. Two common ways to document a project are ~~RA~~ README files to and wikis : 1> README files are a quick and simple way for other users to lern more about your work.
2) Wikis on Github help you present in dept information about our project in a usful way.

10) What does it do ?
-> GitHub is a code hosting platform for version control and collaboration. It lets you and other work together on projects from any where. This tutorial teaches you GitHub essentials like repositories, branches commits, and Pull Requests. We can read commentary and speculation all over the web about the GitHub will do with the money.

11) Who maintains it?
-> The fundamental software that underpins GitHub is Git itself, written by lius torvelds, creator of llux. the additional

software that provides the GitHub user interface was written using Ruby on Rails and Erlang by Github website owned ~~G~~ by GitHub pages

12) How is project structured ?
→ There is no perticular layout imposed on GO projects on github, save that is would be nice to make it go gettable. The structure of a GitHub-based project lillustrating project structure and interaction. use framework combined with application - specific files underneath, so the directory structure for a project. Project boards are found on your Github organization page or on a specific repository. See the projects tab in the same row as issues and pull requests. There you can set a name and description for the project.

13) How do the developers communicate ?
→ It works pretty simply : whenever something happens, say a comment was writen for a commit, an HTTP POST payload is sent to the configured URL (and here you must set up your software to process this payload). Thus, webhooks allow GitHub to be integrated into your communication scheme, which makes it extremly useful.

* some rules for communicate with GitHub.
- Prefer asynchronous communication
- Dont underestimate high - fidelity mediums.

14) What has happened recently?
→ Explore is your guide to finding your next project, catching up with whats trending and connecting with the GitHub. The activity view helps you keep track of everything happening in your teams project and see exactly whats changed since the last time.

15) How do I build it?
→ Cloud Build provides a cloud Build GitHub that allows you to automatically buid your code each time you push a new commit to Github. This tutorial explains how to ~~un~~ install and configure the app. and how to automatically trigger builds on GitHub.
1) prepare a Github repo with some source code to build.
2) Install and configure the google cloud build Github app.
3) observe that the google cloud build app builds your code and publishes results to a pull.
4) Learn about the different ways to configure your builds.

15) How do I contribute ?

→ This is a guide to contributing to an open source project that uses GitHub. Its mostly based on how I've seen Zend framework, slim framework and joined in operate. set up a working copy on your computer. Get it working on your machine. Do some work.

Scanned by CamScanner