

Internal Open Source

Wen Zhenglin
2016-4-21

Overview

- Technical solution
- Some concepts

Organizations want to understand

How they can learn from success and effectiveness of the open source community

How they can apply internal development to further increase creativity and accelerate development

How developers share their skills and knowledge with other developers, bringing their specialized talents together to solve big problems

How to satisfy the desire to building new relationships with other developers that lead to new knowledge and potentially new opportunities

An open styles

More open-style of development further breeds new creativity, based on exposure to new ideas, new knowledge, and new applications

An open-style of development can also happen internally within corporate development environments

Running internal projects with open-source practices

Internal Open Source

Technical solution

Version Control System

Helps track changes to code and revert to any previous version

Issue Tracking System

Helps developers assign tasks and track work on issues and tasks

Wiki

Help a team share documentation

Other relate

Mailing list (communication)

Discussion lists or Discussion forums, News



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.



Learn Git in your browser for free with **Try Git**.



We want “User Interface” too

Not just pure version control system

Can it support our workflow easily (without manual input command)

Does it support issue tracking, collaborative development, a whole concepts of software development life cycle, etc

We need easily create a project/repo (with one click) and quickly work on it



Personal

Open source

Business

Explore

Pricing

Blog

Support

Sign in

Sign up

How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

Welcome home, developers

GitHub fosters a fast, flexible, and collaborative development process that lets you work on your own or with others.



For everything you build

Host and manage your code on GitHub. You can keep your work private or share it with the world.



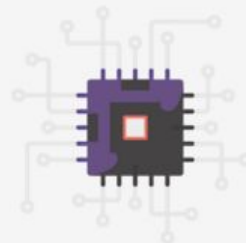
A better way to work

From hobbyists to professionals, GitHub helps developers simplify the way they build software.



Millions of projects

GitHub is home to millions of open source projects. Try one out or get inspired to create your own.



One platform, from start to finish

With hundreds of integrations, GitHub is flexible enough to be at the center of your development process.

Github features

GitHub is the largest code host on the planet with over 31 million repositories. Large or small, every repository comes with the same powerful tools.

1. Integrated issue tracking
2. A flexible issue tracker lets you stay on top of bugs and focus on features
3. Collaborative code review
4. Easily manage teams within organizations

Sounds great, Can we use it ?

Nope, because we got problems

1. For some reason, we may not be able to use it, we can't take the risk
2. Network speed may slow too, We don't want wait too much every time
3. We are not ready or try to open sourcing anything yet

So ?

We need

- A Github similar solution
- No network latency much, but acceptable
- Inside our company (everything is in our control)



Gogs - Go Git Service

A painless self-hosted Git service

Gogs

- An exists solution perfectly fit our need (no need to develop a new one)
- Easy to deploy (small effort, huge benefit)

Benefit

Increases shared knowledge across the company

Manage the communication friction problem by making both the code and issue-tracking “public”

Transparency also produces quality work

Better workflow

Github workflow brief introduction

<https://guides.github.com/introduction/flow/>

Demo

Some concepts

Always get better if we can learn from each other

This can be done with mailing lists, instant-messaging, and “public” code repositories and bug tracking.

Adopting such practices inside a private business can create , a living, breathing community inside our company

Also

When you have a super-wide spread of styles, anytime someone moves they have to relearn everything

That's bad for internal mobility and for code sharing

Traditional approach, define strict rules and stakeholders, mandated from the top, to make people accountable. This didn't solve the problem, but it did generate a lot of meetings

A shared platform for the company's software development life cycle

Shared code repositories,

Shared ticketing systems,

Mailing lists, and code review systems

Applying the open-source model to our internal work

More written communication

More easily access project information and documentation

Without having to set up meetings

Open Source Software practices

Can bring a great deal of developer *joy*

A great opportunity to give developers exposure to code they may otherwise not have the time to examine, and increases shared knowledge across the company

Note about Open source

Open source is not a verb. It's a workflow, a philosophy, a modern approach to software development

PayPal is experimenting with InnerSource to break down silos, grow collaboration and increase productivity

Results were visible after 6 months

The Checkout Platform team spends 0% of its time rewriting code and just 10% reviewing submissions

The team was able to do a major refactoring and a 4x increase in performance without planning for it

The mindset moved from blocking change to mentoring and coaching

Relate to microservice

The original idea of Internal Open source may taken
from “microservice concepts”

——Sam Newman 2015, Building_Microservices.pdf page.196”

Some similar stuff

- C9

An cloud IDE See each other type Code together in real time

- Etherpad

Online editor providing collaborative editing in really real-time

- Wide

Golang web IDE

Summary

Technical solution -- Gogs, A painless self-hosted Git service

Some concepts -- Internal Open Source

Thank You

Wen Zhenglin
wenzhenglin@bjjdsy.com.cn

2016-3-16

