Organizing People for Successful Collaboration on GitHub

Chun-Yu Chen (Jason)

2024.11.19

About Me

- Academic Background
  - University of Bristol – Computer Science (2021, MSc)
  - National Yang-Ming University – Genome Sciences (2017, MRes)
  - National Chung Hsing University – Biotechnoloy (2015, BSc)
- Application Engineer at Allxon (09/2023 – 10/2024)
  - Establish partnerships with ngrok and portainer.io
  - Advocate and support Allxon Octo SDK
  - Pre-sale demo and post-sale technical support for Allxon solutions
  - Prepare demo and training materials for members, customers, and partners
  - Research on Nvidia Jetson series technology (BSP update, OTA deploy, JPS)

Philosophy of Collaboration on GitHub

- Key activities in collaboration:
  - Communication
  - Contribution
  - Administration
- Key entities for managing collaboration:
  - Individuals
  - Organizations
    - Enable shared **ownership** and **administration** of repositories
    - Example: System and Service Team
  - Teams
    - Groups of users within an organization
    - Example: AE Team, Engineering Team

About Repositories

- GitHub's core services center around repository functionalities

- Repositories are used to manage work and collaborate effectively:
    - **Issues**: Track bugs, tasks, and discussions
    - **Wiki**: Share documentation or project resources
    - **Discussions**: Facilitate team conversations

- Public vs. Private Repositories:
    - **Public**: Accessible to everyone on the internet
    - **Private**: Accessible only to you, invited collaborators, or organization members
    - Permissions and visibility settings apply to the entire repository, including code, issues, and the wiki

About Organizations

- Organizations are shared accounts where **multiple members** can collaborate

- Users can belong to multiple organizations

- Permissions are assigned based on roles:
  - **Owner**: Full control over the organization

  - **Member**: Limited access based on assigned permissions

  - **Outside Collaborator**: Access to specific repositories without membership

- **Teams**: Sub-groups of members within an organization
  - Control repository access levels (admin, read, or write)

  - Enable organization members to notify the entire team efficiently

How to Collaborate with External Collabrators

- **Join an Organization**
  - **As Members**: Collaborators are added to the orgnization (teams)
    - Permissions can be adjusted for all repositories a team has access to
    - Ideal for long-term collaborators requiring access to multiple repositories
  - **As Outside Collaborators**: Collaborators are added to specific repositories
    - Have no visibility into the broader organization or other repositories
    - Best for temporary or repository-specific access
- **Repository-Level Collaboration**
  - The access level is the same as that of outside collaborators
  - Offers flexibility for **individual** collaboration

Let's see a demo on GitHub! ✌️