

Introduction to collaboration and automation tools

Getting the most out of the git ecosystem

A small overview

collaboration	 GitLab • open-source collaboration platform based on git • hosted on gitlab.com or self-hosted • Community & Enterprise edition available	 GitHub • closed-source collaboration platform based on git • hosted on github.com, owned by Microsoft (since 2018) • Community and Enterprise features available
large files	 git-annex • extends git support to non-text files • flexible, symlink based file handling • powerfull command line interface • supports diverse remote types	 git-lfs • extends git support to non-text files • pointer based file handling • supported by GitHub & GitLab
version control	 git • protocol for systematic version control of text files • comes with git command line interface • no integrated visualization / gui	

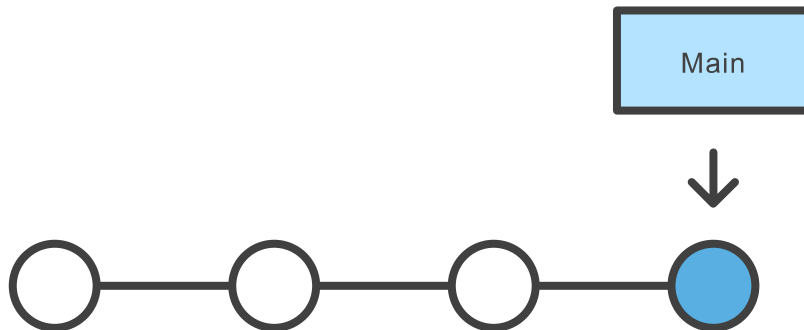
Key version control features: **Git commit**

Version control requires tracking of

- *who* did
- *when*
- *what* changes
- and *why*

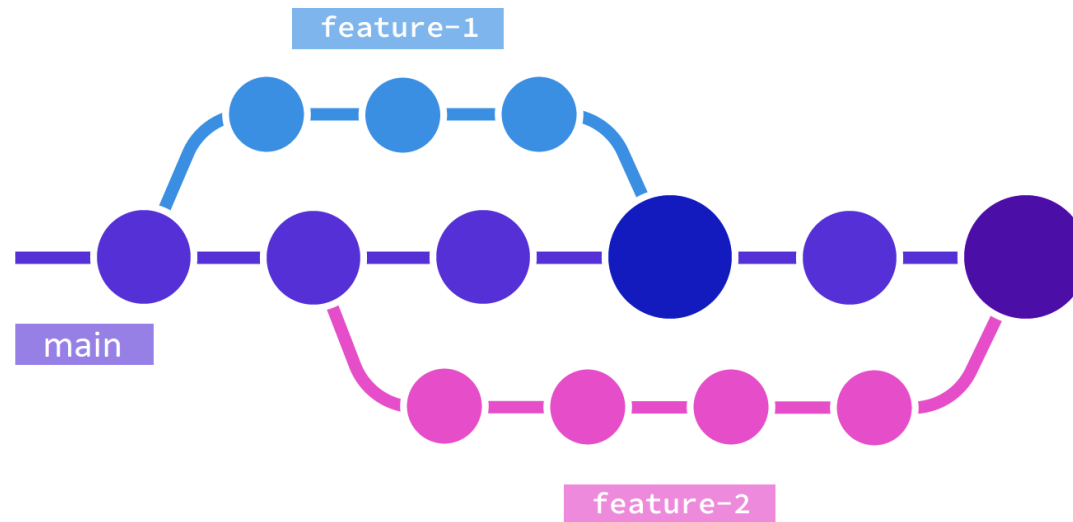
Git commit attributes

- unique commit hash
- author
- date
- parent commit hash
- differential changes
- commit message



Key version control features: Git branch

- *alternative history with common origin*
- can be merged back into `main` branch via `merge commit`





Collaboration | Do we need it?

- *"The whole is more than the sum of its parts."*
- shared knowledge → robust knowledge preservation
- need for communication → better documentation
- essential for contributions from diverse backgrounds

→ Requires systematic collaboration approach to be efficient and maintainable



Collaboration | Github Feature: **Issues**

- External Feedback
 - User Feedback
 - Bug reports
 - Feature requests
- Collaboration coordination
 - Issue discussion
 - Issue assignment
 - Issue labels
 - Milestones

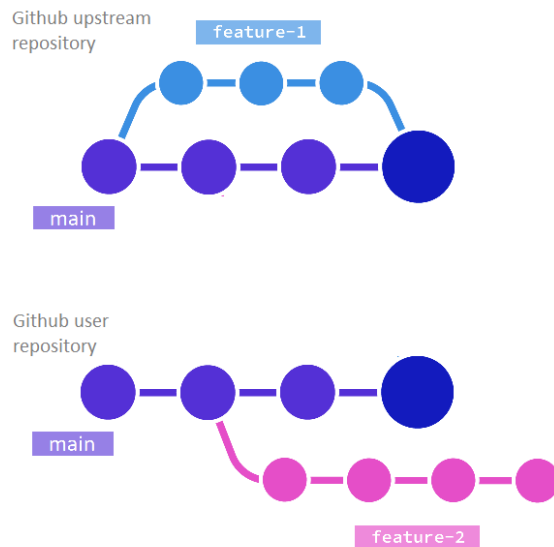
Example: [IAEA-NDS/open-benchmarks issues](#)



Collaboration - Github Feature: Forks

Distributed development

- Permits non-centralized, shared software development
- Robust system architecture



Fork features

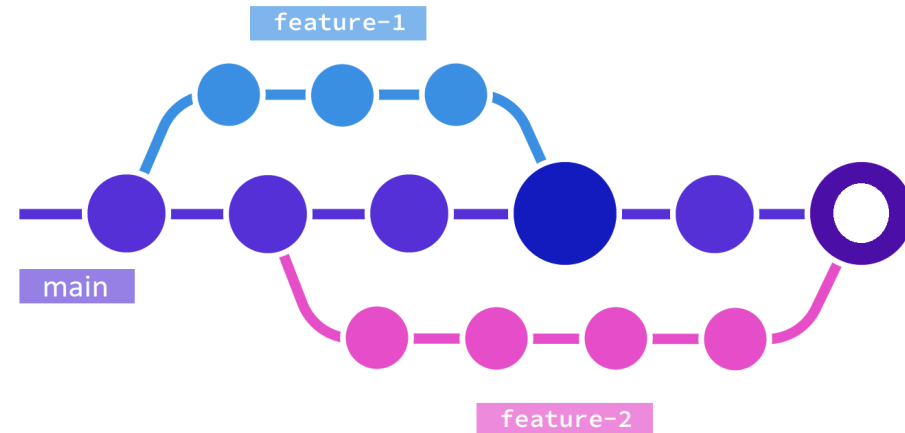
- User specific repository *copy*
- Shared version control history
- Changes can be integrated in upstream repository via Pull request (PR)

Example: [dbrown170/X4I](https://github.com/dbrown170/X4I)



Collaboration | Github Feature: Pull Requests

- Internal project organization
 - Grouping changes by category and features
 - Discussing proposed changes
 - general comments
 - line-by-line specific comments & suggestions
 - Sharing experimental features
- Collaboration coordination
 - Change discussion & review
 - Grouping via labels, milestones

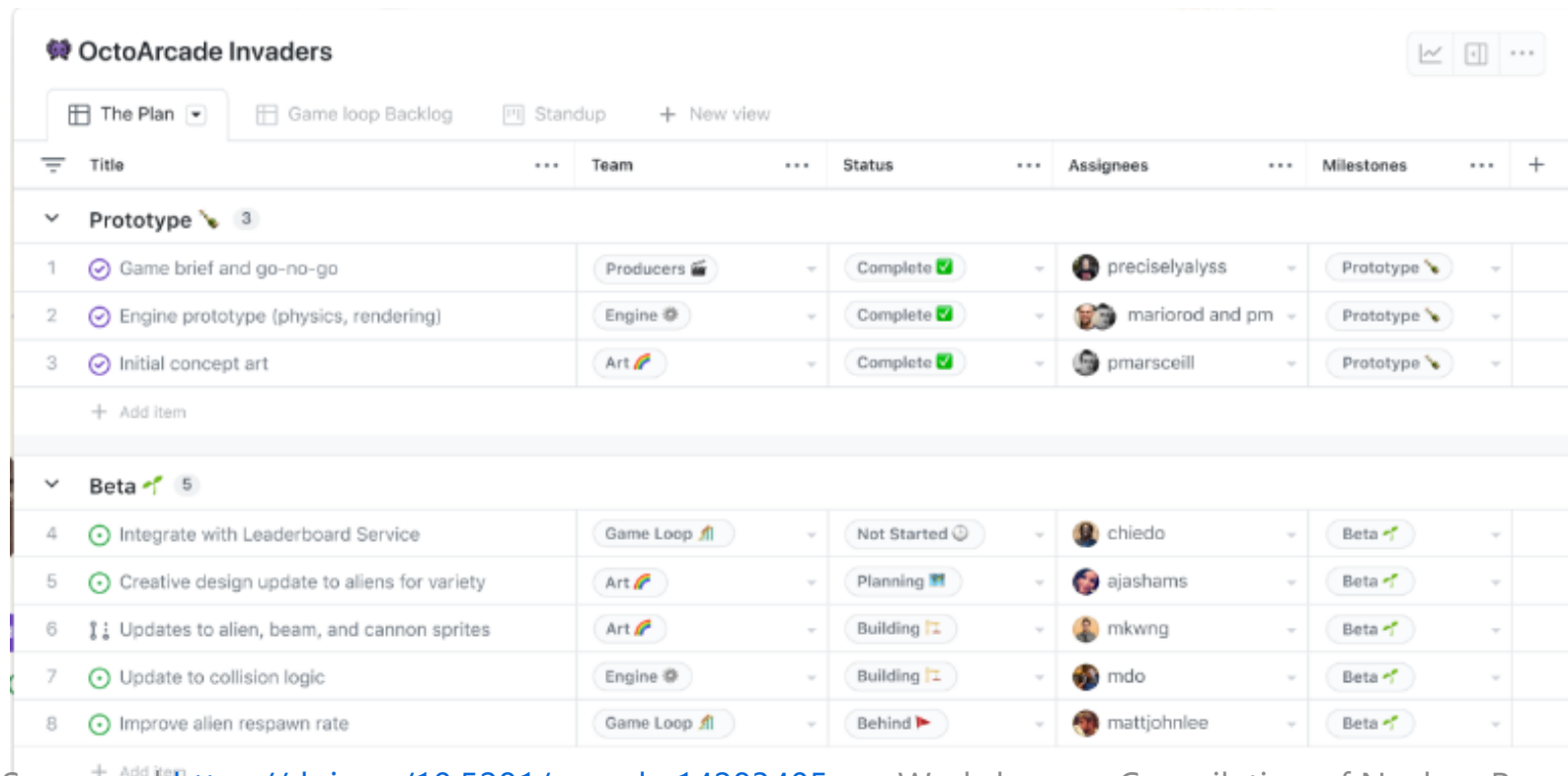


Example: [dbrown170/X4I/pull/36](https://github.com/dbrown170/X4I/pull/36)



Collaboration | Github Feature: Projects

- meta level organization of issues and pull requests
- can be also used across repos of an organization
- diverse view modes (Kanban, Table)



The screenshot displays the Github Projects interface for the 'OctoArcade Invaders' repository. The interface includes a header with the repository name and icons for chart, table, and menu views. Below the header, there are tabs for 'The Plan', 'Game loop Backlog', and 'Standup', along with a '+ New view' button. The main content area shows a table of project items, organized into two sections: 'Prototype' (3 items) and 'Beta' (5 items). Each item row includes a number, a title, a team, a status, assignees, and milestones.

	Title	Team	Status	Assignees	Milestones	
▼	Prototype 🚧 3					
1	✓ Game brief and go-no-go	Producers 🎬	Complete ✅	preciselyalys	Prototype 🚧	
2	✓ Engine prototype (physics, rendering)	Engine ⚙️	Complete ✅	marirod and pm	Prototype 🚧	
3	✓ Initial concept art	Art 🎨	Complete ✅	pmarsceill	Prototype 🚧	
+ Add item						
▼	Beta 🚧 5					
4	🔄 Integrate with Leaderboard Service	Game Loop 📊	Not Started 🕒	chiedo	Beta 🚧	
5	🔄 Creative design update to aliens for variety	Art 🎨	Planning 📅	ajashams	Beta 🚧	
6	🔄 Updates to alien, beam, and cannon sprites	Art 🎨	Building 🏗️	mkwng	Beta 🚧	
7	🔄 Update to collision logic	Engine ⚙️	Building 🏗️	mdo	Beta 🚧	
8	🔄 Improve alien respawn rate	Game Loop 📊	Behind 🚦	mattjohnlee	Beta 🚧	
+ Add item						



Collaboration | Github Feature: Notifications

- Users can subscribe to watch specific
 - issues
 - pull requests
 - complete repositories specifying custom notification conditions
- Users can follow organizations
- Users can be *pinged* via `@username` syntax

Collaborative documentation

- Documentation = just another type of text
- can be build and hosted directly on github and related platforms
 - build via automation
 - sending built docs to host platform, e.g.
 - [readthedocs](#)
 - [github pages](#)
 - custom webspace
- [Wiki](#) functionality on a per-repository bases

Project overview / statistics

- Contribution statistics available via *Insights*

Example: [IAEA-NDS/endf-parserpy/graphs/commit-activity](https://insights.iaea.org/IAEA-NDS/endf-parserpy/graphs/commit-activity)

Automation

Purpose

- replace tedious manual tasks
- make scripted processing steps more reproducible
- reliably run tests upon changes
- scheduled updating
- dependency update automation

Implementation

- Jobs can run on GitHub runners (free, enterprise)
- [GitHub Actions](#) are defined in YAML format
- highly customizable trigger conditions

Automation | Testing

Tests: Guarantee expected functionality of package / consistency

- typically run automatically upon mayor changes

Example: [IAEA-NDS/endlf-parserpy](https://github.com/IAEA-NDS/endlf-parserpy)

Automation | Publishing

- Build and publish package in a consistent manner
- typically triggered manually

Example: [IAEA-NDS/endf-parserpy](#)

Automation | Scheduled Jobs

- Run jobs on a regular basis, e.g. for automatic **updates** and **notifications**

Examples: multi-repository updating and automatic publishing & DOI registration

- [NRDC Documents @ NEA Gitlab](#)
- [exfor-main @ NEA Gitlab](#)

Discussion

Which software or other projects could benefit from using

- git-based collaboration tools
- automation