

**Name:** mcp-381-cswg-workshop-git-repository-layout-and-workflow

**Title:** Git Repository Layout and Workflow

**Status:** Draft -- anyone can edit.

See the [MCP index](#) to create or find documents, or [mcp-0-readme](#) for an overview.

The headers above are machine-readable; please preserve format.

---

Text checkins (10 mins)

- Steve
  - Good progress on <https://github.com/stevegt/aidda/tree/main/x/x3>
  - starting to suspect that one or more of the following are true:
    - aidda should be a grokker subcommand
      - otherwise we have some fragility in aidda's dependence on grokker's library functions.
      - Go's hashing in go.sum makes this much less fragile than it would be in most languages, but the dependence is still there.
    - grokker should be renamed aidda
    - grokker and/or aidda becomes a promisify node
      - merge in code from <https://github.com/stevegt/x/tree/dev/grid-cli>
    - as opposed to building a giant monolith, the current functionality of grokker and aidda becomes callable functions and apps in promisify
      - because grid functions are hash-addressable, this avoids the above fragility
      - essentially this moves the benefits of hashing from go.sum into grid addresses; more general, less fragile
      - lets aidda, grokker, and other things that interact with them do so safely without having to build everything into one big monolithic binary
      - more "UNIX-like" -- little tools and functions that can be composed to make larger things
    - related conversation at <https://chatgpt.com/share/e/f6bee125-b3c3-4a53-856d-7d2e93539f12>
- JJ
  - Graduation: 1 down, two to go
  - House across the street was demolished.
  - Gdocs program
  - On boarding
  - 1.5 more weeks of my club. Blog will be affected.
- Richard
- Donaldo

- Create agenda below
  - Practiced Ticketing
- **How to turn off email notifications on Gitea**
  - You 'could' unwatch a specific repo
    - But not really useful for monorepo setup
  - Settings > Account > Manage Email Addresses > "Only Email on Mention" or "Disable Email Notifications"
- "Create Business for Makers" course last week
  - Mostly review content
  - Goal to launch a maker business, 3d printing, laser cut, print making, digital product, etc.
- Starting Climate Change AI course on Thursday
- Considering attending FAB24 in Puebla, Mexico in August?!
  - <https://fab24.fabevent.org/>

## Git Repository Layout and Workflow (40 mins)

- Overview of Git workflows:
  - <https://gitea.t7a.org/cswg/general/issues/3>
  - Developer-Based Naming
    - Branches are named after developers, common in small teams.
      - Pros: Easy to track individual contributions.
      - Cons: Not scalable for larger teams; multiple devs on similar features; lacks focus on development stages.
    - How can individuals have their own experimental features?
      - **Use username-prefixed branch names** similar to [mob-consensus](#) tool
        - username/feature
        - username/bugnumber\_some\_description
      - a good convention is to always prefix branch names with your username, i.e. **not** do:
        - mycrazyidea\_2
        - bugnumber
  - Functional/Stage-Based Branch Naming
    - Branches named by their function or development stage, such as feature, development, staging, and production.
      - Pros: Supports structured workflows, ideal for parallel development and continuous integration.
      - Cons: More complex, requires strict merge protocols to maintain stability.
    - the [mob-consensus](#) tool also makes use of username prefixes for feature work as well, e.g.
      - username/feature

- An example Mono Repo Structure from [general/core](#)
  - /project Folder
    - 000 - internal tools
    - 001 - training related
    - 002 - server/infra related
    - ...
  - /u Folder
    - User miscellaneous project folder
  - Other stuff
    - Makefile
    - /bin
    - /train - CSV based training tracking
    - ...
- Choosing our Workflow:
  - **The main purpose of cswg/general is for issues and internal docs, not code**
    - we should primarily be using cswg/general only to replace the TODO notes we've been putting in docs
    - we may not even have anything in ./bin, since everyone is running a different O/S
  - Protect the main branch same as general/core
    - Steve only maintainer with write access to main
      - Best we have until we have a good consensus tool we trust
  - Everyone creates their own development branch
    - This way it stays a shared repo with individual branches, versus creating forks
    - Pros:
      - Now less worrying on "breaking stuff"
      - No need to deal with pull requests
      - Shared issue tracking
    - Cons:
      - Accidentally committing big blobs of data
      - Using pull requests would be good practice for contributing to projects outside of this team
  - Starting a New Projects
    - If the project 'deserves' its own repo, then it does not go in cswg/general
      - a project "deserves" its own repo if any of the following are true:
        - will be released on github
        - is already on github
        - is cdint-specific
    - Otherwise add it to the /project Folder
      - A new project started by creating an issue
      - Then, a folder is named based on issue # created.

- Example:
    - <https://gitea.t7a.org/cswg/general/src/branch/donaldo/project/001>
- Merging strategy
  - Merging during code review
  - Mob sessions
    - e.g. [mob-consensus](#) tool
      - currently a shell script, should be merged with aida while converting to Go
  - Async strategies
- Considerations for thought:
  - should we be using forks instead of branches?
    - pros:
      - the main purpose of cswg/general is for issues anyway, not code
        - we should primarily be using cswg/general only to replace the TODO notes we've been putting in docs
      - would be more like github-style projects
      - would be easier for new people to adopt
      - no repo proliferation (easier to keep track of where code is)
      - one issue set
    - cons:
      - mob-consensus and other tooling would want to work differently
      - repo proliferation
      - forks would each have own issue sets (but this works the same as github -- convention is to not create issues in forks)

**Hands-on activity:** Create a Developer Branch on [cswg/general](#) (30 min)

*\*\*Move to after workshop as needed\*\**

1. Add your SSH key to Gitea
  - a. Open terminal
  - b. `cat ~/.ssh/id_rsa.pub`
2. Clone Repo
  - a. `cd lab/`
  - b. `git clone ssh://git@gitea.t7a.org:3022/cswg/general.git`
  - c. `cd general`
3. Check what branch you're on
  - a. `git branch -a`
4. Create a new branch
  - a. `git checkout -b $USER`
  - b. `git status`
5. Try to git push
  - a. `git push`
6. Set a new upstream
  - a. `git push --set-upstream origin $USER`

7. Create your folder
  - a. `mkdir -p project/003`
  - b. `cd project/003`
  - c. `touch README.md`
8. Commit & Push changes
  - a. `git status`
  - b. `git add README.md`
  - c. `git status`
  - d. `git commit -m "Create placeholder for #3"`
  - e. `git push`
9. Done! 🎉

#### Questions & Next Steps

- Feedback questions
- Follow-up items
  - Review with Rebecca
  - Create cswg/general README.md

#### Plan next workshop (10 mins)

- go to workshop proposals ( [mcp-369](#)), follow the instructions there, replace this bullet point with the link to the new doc