# NEMSfv3gfs repository and code management

Jun Wang

# contents

- NEMSfv3gfs code management
- NEMSfv3gfs VLAB access and repository
- VLAB gerrit setup

- FV3 code change procedure:
  - VLAB redmine
  - make changes in VLAB git repository
  - compile and run jobs
  - code review

# NEMSfv3gfs code management

1) proposal of changes, including expected code and science impacts submitted by the developer through ticket system

2) review committee evaluates proposal and provides feedback

3) if okayed, branch is created and software work completed

4) run regression tests in the branch to evaluate impacts on science and computational performance

5) evidence-based review of science impacts before final code review

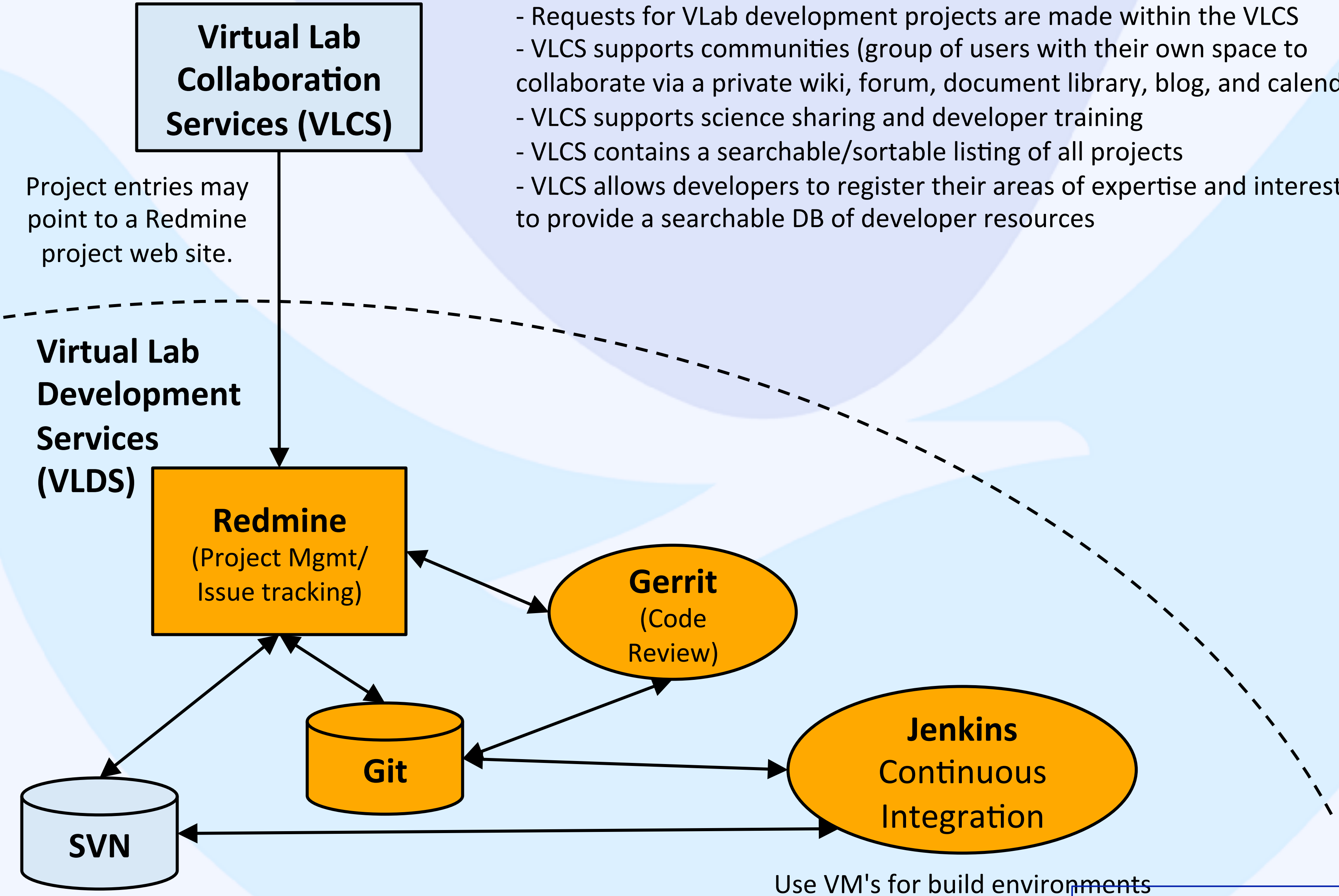
6) Master branch development merged into the branch, submit for code review

7) code review done by review committee and code manager

8) Code is committed to repository

# Virtual Lab High Level View

- Requests for VLab development projects are made within the VLCS
- VLCS supports communities (group of users with their own space to collaborate via a private wiki, forum, document library, blog, and calendar)
- VLCS supports science sharing and developer training
- VLCS contains a searchable/sortable listing of all projects
- VLCS allows developers to register their areas of expertise and interests to provide a searchable DB of developer resources



**Virtual Lab Collaboration Services (VLCS)**

Project entries may point to a Redmine project web site.

**Virtual Lab Development Services (VLDS)**

**Redmine** (Project Mgmt/ Issue tracking)

**Gerrit** (Code Review)

**Git**

**SVN**

**Jenkins** Continuous Integration

Use VM's for build environments

# Access VLAB FV3GFS code

- To access VLAB FV3GFS
  - All users need to have VLAB access before they can access fv3gfs code
  - NOAA users: please go to following webpage:

https://vlab.ncep.noaa.gov/group/fv3gfs/access

  - Non-NOAA users will need a NOAA sponsor to submit the External Partner Account Request Form at:

https://vlab.ncep.noaa.gov/group/guest/external-user-request

# VLAB gerrit setup

- NEMSfv3gfs VLAB gerrit git repository:

  **ssh://vlab.ncep.noaa.gov:29418/NEMSfv3gfs**

  - Submodules were set up under NEMSfv3gfs repository:

    **ssh://vlab.ncep.noaa.gov:29418/FV3**

    **ssh://vlab.ncep.noaa.gov:29418/NEMS**

  - To get the NEMSfv3gfs code:

    *git clone --recursive gerrit:NEMSfv3gfs*

- The VLAB gerrit alias needs to be set up for developers to access the git repository using gerrit

  1. Log into: https//vlab.ncep.noaa.gov/code-review
  2. Upload your public ssh key
  3. Set up an ssh alias called gerrit
  4. Set up user.name and email that git users
  5. Test your configuration by running “ssh gerrit” and “*git clone --recursive gerrit:NEMSfv3gfs*”

# VLAB redmine project

- Web site that supports project management
  - Issue tracking
  - WIKI
  - Browse project repository
  - News
- Users log in via Google SSO or their NOAA LDAP credentials
- Administrator or project manager assigns members to projects with appropriate roles
- Import tool available to migrate issues and wiki pages from Trac

Kenneth Sperow

# NEMSfv3gfs VLAB redmine project

- The nemsfv3gfs/fv3 VLAB redmine website

https://vlab.ncep.noaa.gov/redmine/projects/nemsfv3gfs

https://vlab.ncep.noaa.gov/redmine/projects/comfv3

  - redmine issue page is in above directory
  - Comfv3 issue page contain all the model source code changes
  - NEMSfv3gfs issue page contain the changes in model configuration (namelist, compile options, modules) and regression test and log files

- Create new issues
  - Each development work needs an issue associated with it. The issue number will be used at code review.
  - A “new issue” green” +” button is shown on the top right corner of the redmine website, clicking on it will bring you to the new issue webpage.
  - Information such as subject, issue description, status and due date needs to be filled out.

# NEMSfv3gfs VLAB redmine project (cont.)

- Update an issue
  - You can record your working progress on the issues.
  - The "edit" button is on the top right corner of the panel, clicking on it allows you to update your issue.

    I. In  the log section, write down the number of hours worked, the activity conducted and brief comments

    II. Using "notes" sections for more details. When your code is pushed for review, you can link your issues with gerrit code review number by adding a line in the notes section:

    "gerrit code review #XXXXX": https://vlab.ncep.noaa.gov/code-review/#/c/XXXXX/.  Also you can load files for more documentation related to your work.

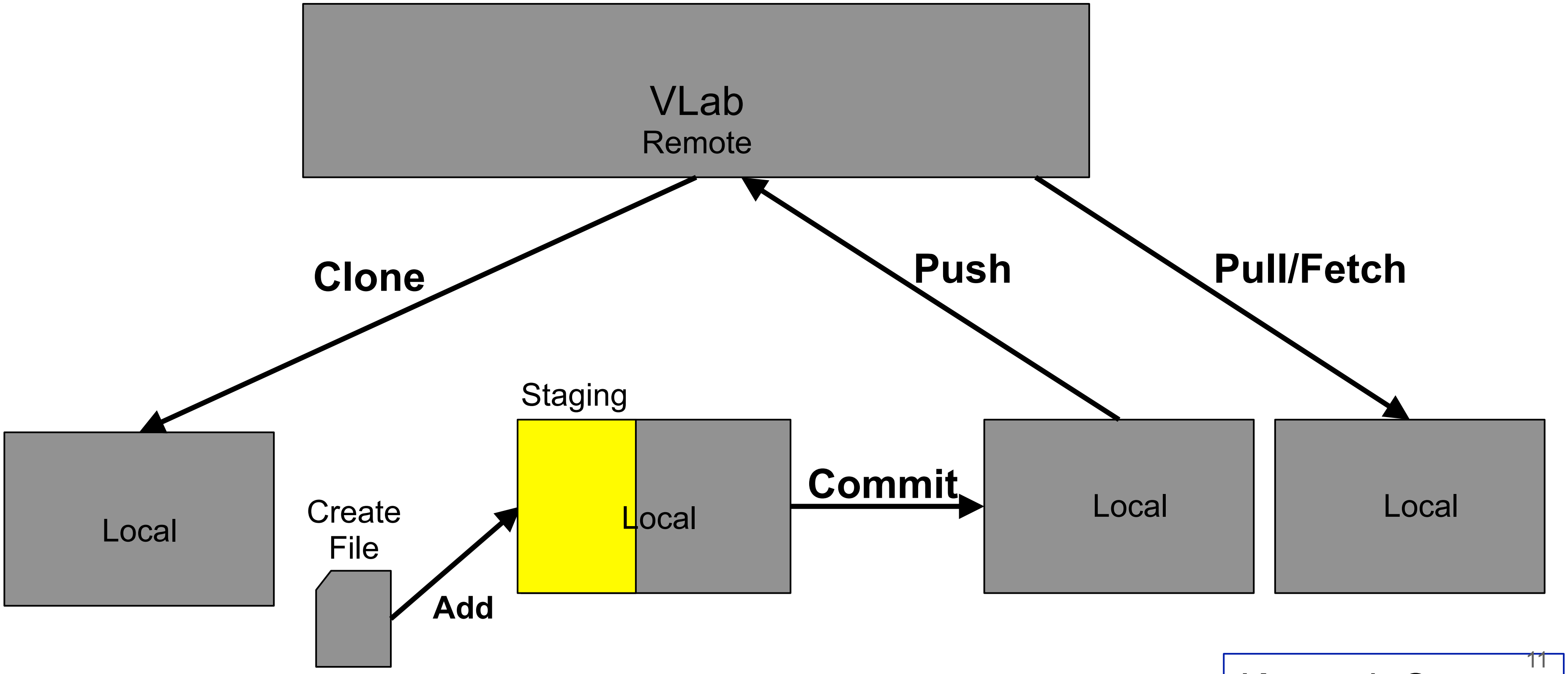
    III. Update the "% done" to indicate how much work is done.

    IV. When the issue is resolved, please change the "status" to "resolved".

# Make changes in VLAB git repository

- Create a new issue (ticket) and get it approved  by review committee
- Create a new NEMSfv3gfs git branch with new fv3 branch (and/or nems branch) associated with it through submodule.
- Compile and test the code in the branch, merge with the latest master branches, evaluate test results
- Run regression test on all the platforms
- Push for code review, make corresponding changes if reviewers suggest any
- Code changes is committed

# Understanding Git Repos



Kenneth Sperow

# Create new branches

## Step 1: clone the nemsfv3gfs repo with message hooks

```
git clone --recursive gerrit:NEMSfv3gfs && scp gerrit:hooks/commit-msg NEMSfv3gfs/.git/hooks/
```

On wcoss:

```
git clone --recursive gerrit:NEMSfv3gfs && scp  -F ~/.ssh-local/config gerrit:hooks/commit-msg NEMSfv3gfs/.git/hooks
```

## Step 2: check out master branch if you are not in it, and then create a nemsfv3gfs branch from it

```
%git branch (if you are not in master branch, do: %git checkout master)
%git checkout -b your_nemsfv3gfs_branch_name
```

## Step 3: Create FV3 branch

```
%cd FV3; %git branch (you should see you are in master branch, if not, do: %git checkout master
%git checkout -b your_fv3_branch_name; %git commit -a
%git push origin your_fv3_branch_name
```

## Step 4: add the new fv3 branch to your nemsfv3gfs branch

```
%cd .. (now you are in your nemsfv3gfs branch)
%git status ; %git add FV3; %git commit –a; %git push origin your_nemsfv3gfs_branch_name
```

# Make changes in branches

- Switch to your branch *%git checkout your_branch_name*
- Add a file/directory: *%git add file_name/directory_name*
- Remove a file/directory: *%git rm file_name/directory_name*
- Check branch status: *%git status*
- Commit to local branch: *%git commit -a -m”message” (or -F message_file)*
- Commit to remote branch: *%git push origin your_remote_branch_name*
- Merge with latest branch: % git pull origin your_remote_branch_name (if you are on different computer)
- Merge with latest master: *%git pull origin master*
- unstage changes: *%git reset HEAD*
- Revert last commit: *%git reset --hard HEAD~1*

# Compile and run NEMSfv3gfs

- No change on compiling and running NEMSfv3gfs test
  - Compile source code:

*%cd NEMSfv3gfs/tests*

*%./compile.sh PATHTR MACHINE_ID MAKE_OPT BUILD_NR CLEAN_BEFORE CLEAN_AFTER*

Or

*%cd NEMSfv3gfs*

*%./NEMS/NEMSAppBuilder app=standaloneFV3*

  - Run a test:

*%cd NEMSfv3gfs/tests*

*% ./rt.sh –l rt.conf1 (rt.conf1 is modified from rt.conf)*

Or:

*%cd NEMSfv3gfs*

*./NEMS/NEMSCompsetRun '{testname}' (see test name in compsets/all.input)*

# Run regression test

- Running NEMSfv3gfs rt.sh test on wcoss cray and theia

  *%cd NEMSfv3gfs/tests*

  *% ./rt.sh –f*

  *To create new baseline:  ./rt.sh –c fv3*

- Running NEMS test on all platforms: wcoss phase1/2, cray, theia, xjet, svjet, gaea c3 and c4

  *%cd NEMSfv3gfs*

  *%./NEMS/NEMSCompsetRun –f*

  *To create new baseline: ./NEMS/NEMSCompsetRun  --baseline fv3*

  *To run the test in a specific directory:*

  *./NEMS/NEMSCompsetRun –f –temp-dir=my_directory*

  *To run the test in a specific platform:*

  *./NEMS/NEMSCompsetRun –f - -platform=gaea.c3*

# Push for code review

**Step 1: commit all changes to your FV3 branch**

```
% cd FV3
% git remote update
% git commit -a -m”message”
% git push origin your_branch_name
```

**Step 2: merge your fv3 branch with master**

```
% git checkout master
% git pull origin master
% git checkout your_fv3_branch_name
% git merge master
% git commit -a -m”message”
% git push origin your_branch_name
```

**Step 3: check out master and merge with your branch**

```
%git checkout master
%git merge --squash your_fv3_branch_name
%git commit -a -m”This commits #refs YYYYY ….”
```

**Step 4: push for code review**

```
%git push origin HEAD:refs/for/master
```

# Push for code review (cont.)

## Step 5: submit for code review

- Go to https://vlab.ncep.noaa.gov/code-review/

Under top line, there is “my” item, under it, click on “changes”, you will see “my review” page. Under “outgoing reviews”, click on the code review message you submitted in step 3).

- On the right panel, there is “reviewers” line, clicking on the “add reviewer” button on the the right

## Step 6: push code review for the nemsfv3gfs branch

```
%cd NEMSfv3gfs
%git branch (if you are in master, please do: git checkout your_nemsfv3_branch_name)
%git commit -a -m”get the updated changes”
%git push origin your_nemsfv3_branch_name
%git checkout master (you will see FV3 is modified)
%git pull origin master
%git merge --squash origin/your_nemsfv3_branch_name
%git commit –a –m” m”NEMSfv3gfs, this commits #refs YYYYY ….”
%git push origin HEAD:refs/for/master
```

## Step 7: Update gerrit code review page

- Go to https://vlab.ncep.noaa.gov/code-review/, add reviewers.
- Only after regression test log files run successfully and are updated in the code pushed for review, can code manager update the nemsfv3gfs master.

# Code review

- Developers are members of code reviewer group.
- VLAB code review web page at: https://vlab.ncep.noaa.gov/code-review
- Review other developer's code changes:

1. You will receive an email with title: **Change in NEMSfv3gfs[master]: "commit message"** and with a code review link in in the content. You can go to VLAB code review page, click on "my" in the top raw of the web page, then click on "changes" to see the code reviews in the "incoming reviews" section.
2. Code changes can be viewed in the bottom panel in "Files section". Click on each file to review the code changes from current master.
3. If you want to checkout the code and verify the code changes, you can clone the repo, then do:

i) clone NEMSfv3gfs and cd NEMSfv3gfs

ii) go back to NEMSfv3gfs code review page, on the upper right corner, click on "download.", copy the git command in the checkout line, execute under NEMSfv3gfs directory, then do:

Git submodule update - - init - -recursive (to get submodules)

iii) If just check FV3 code changes: cd FV3 ; go to the FV3 code review page, on the upper right corner, click on "download.", copy the git command in the checkout line, execute under FV3 directory.

4. On the top of the right panel, click on "reply…", write down your review comments, give a score for "code-review".

# Code review (cont.)

## Respond to reviewer’s comments

Reviewer’s comments will be automatically sent to the developer and other reviewers through email. It is required that all the issues brought up by the reviewers need to be resolved before the code can be committed.

- Developers can go to the code review page to answer reviewer’s questions.
- Sometimes reviewers make comments in the source code, these comments can be seen in the “Files section” in the code review page.
- If source code modifications are required:

**Step 1:** clone NEMSfv3gfs  and cd NEMSfv3gfs, see step 3 ii) in previous slide.

**Step 2:** Make all the code changes

**Step 3:** Amend the code changes

*%cd FV3*

*%git commit --amend*

A commit message window will pop up, you will see your commit message from 3c and you can add extra commit message, then **in the end of this file**, please add:
Change-Id: xxxxxxxx

**Step 4.** push for code review: *%git push origin HEAD:refs/for/master*

# References

- Jun Wang & Mark Potts:

https://docs.google.com/document/d/1zYCHRYoWnpYZoPatsz-uEce-_e8DoqsK9HIE3o7nvXE/edit#

- Mark Potts:

https://drive.google.com/drive/folders/0B-DFeEVjBKT8WVB2Ulp0M3VGVEU

- Ken Sperow:

https://docs.google.com/a/noaa.gov/presentation/d/11Zv-U_uh9XblBY9g9xA9UfCWZA5XUhMFDZwi-VkKpg8/edit?usp=sharing

- Git SVN Crash Course: http://git.or.cz/course/svn.html
- Git for SVN users: https://people.gnome.org/~newren/eg/git-for-svn-users.html
- Git for SVN Users Cheat Sheet:

https://www.git-tower.com/blog/git-for-subversion-users-cheat-sheet/

- Git website: https://git-scm.com
- Redmine wiki page: https://vlab.ncep.noaa.gov/redmine/projects/vlab/wiki
- Gerrit Help: https://vlab.ncep.noaa.gov/redmine/projects/vlab/wiki/Gerrit_Help
- Basic Git Guide: https://vlab.ncep.noaa.gov/redmine/projects/vlab/wiki/BasicGitGuide

# Future development on NEMSfv3gfs repository

- Components such as fms are planned to be removed from FV3 repository and become an independent component under NEMSfv3gfs and other coupled system

- To facilitate the Unified Forecast system, other approaches besides git submodule are proposed and under consideration.
  - Umbrella repository:
    - Each UFS application has an umbrella repository
    - Model components, workflow , etc. have their own authoritative repository with associated code management
    - Umbrella repository contains no application source code or data but with configuration files
    - The repository has proper documentation and relies on connecting tool manage externals

# Thank you!