Introduction to GitHub and SU2 Development Practices

SU2 Winter Workshop

February 3rd, 2017

Dr. Thomas D. Economon

Department of Aeronautics & Astronautics

Stanford University

SU2

The Open-Source CFD Code

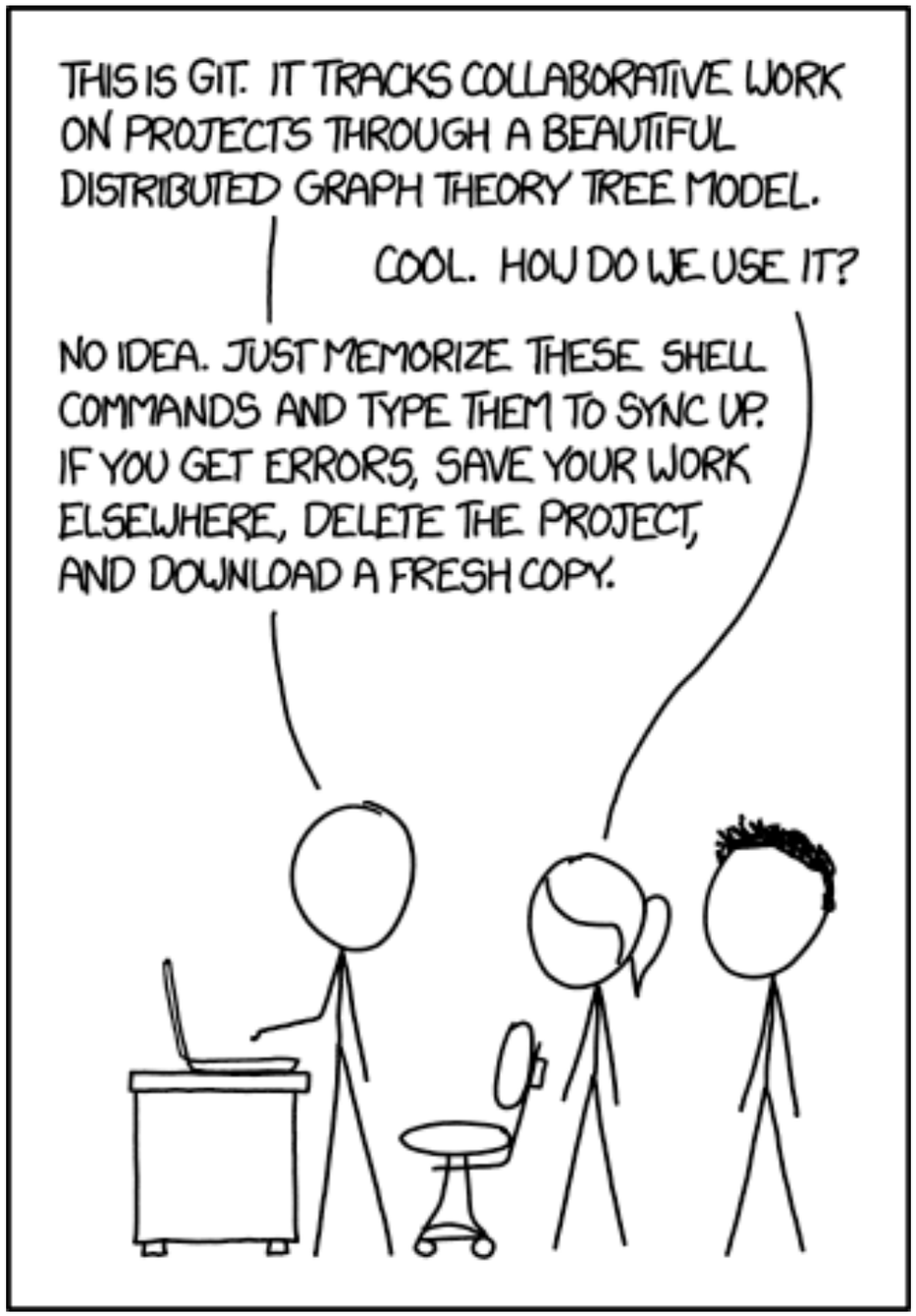
It’s bright where we’re headed.

SU2
The Open-Source CFD Code

How do we get there? Scalable development practices.

- How do we avoid code conflicts?
  - **Branching model in git for decentralized, parallel development.**
- How does one contribute code contributions to the repo?
  - **Pull requests through GitHub.**
- Quality assurance?
  - **Automatic, pre-merge regression testing (Travis CI) and code reviews.**
- How do we minimize the overhead of software development in a research environment?
  - **All of the above + streamlined release process at regular, frequent intervals.**

1. Decentralized development in git



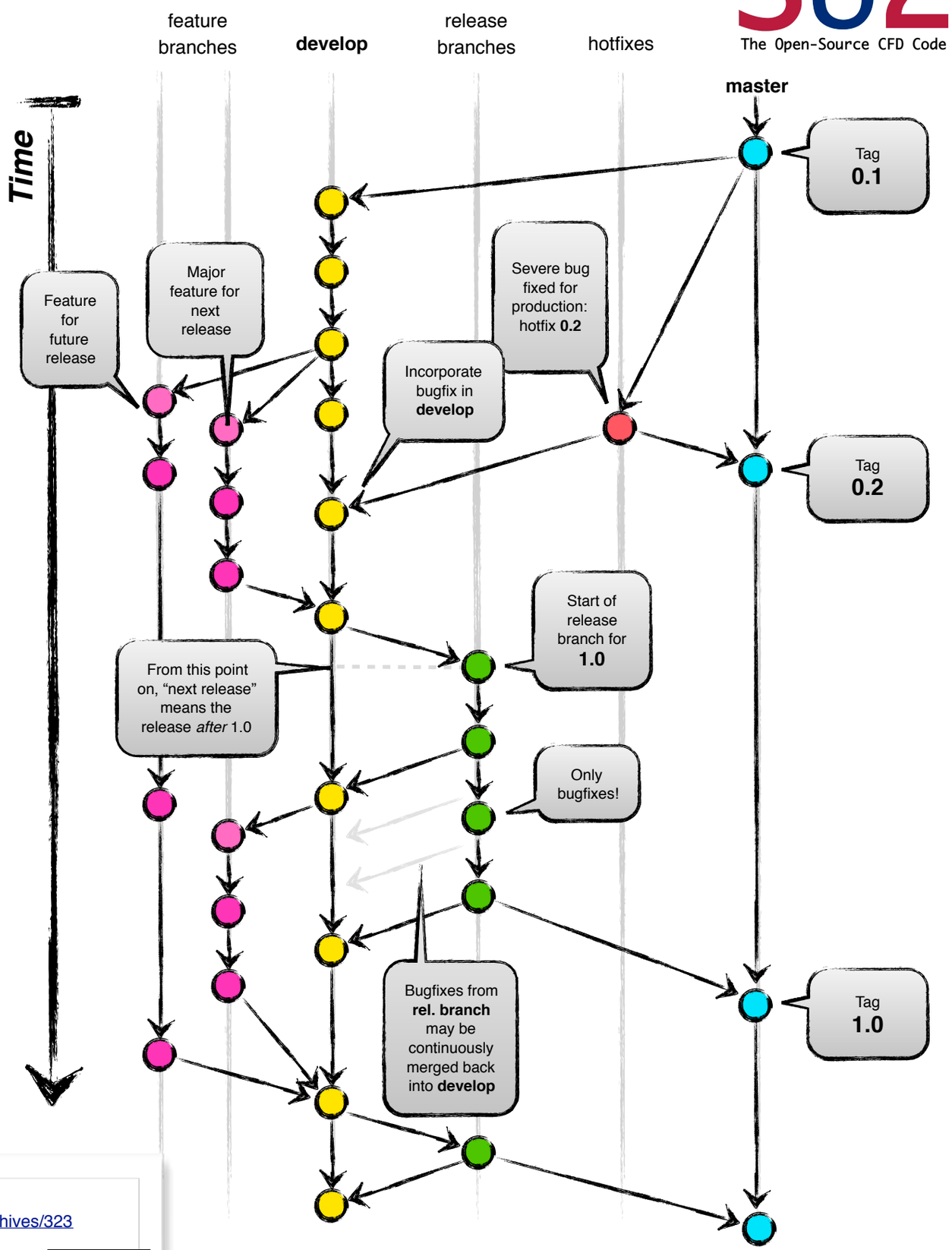
source: http://xkcd.com/1597/

1. Each new feature/capability should have its own branch. Note: internal devs should create branches directly in SU2 repo (not forks) to increase collaboration.

2. All branches operate in parallel, with "owners" updating their feature branches from develop regularly, i.e., '$ git merge develop'.

3. Once ready, owners prepare a pull request for feature. Code is reviewed, and after tests pass, merged into develop. Remove feature branch.

4. At regular intervals, develop is staged for a release. Once ready, it is pushed to master, tagged, and released. Note: master is always stable.



1. Decentralized development in git

master	Updated 13 days ago by economon	✓	Default	Change default branch
feature_hom	Updated 10 hours ago by vdweide		0 | 150	New pull request
develop	Updated a day ago by fpalacios	✓	0 | 15	New pull request
feature_renaming	Updated 2 days ago by fpalacios	✓	0 | 12	#304 Merged
feature_pyWrapper	Updated 5 days ago by tobadavid	✓	0 | 16	New pull request
feature_DE_AD	Updated 6 days ago by rsanfer		84 | 94	New pull request
feature_driver_prototype	Updated 12 days ago by LaSerpe		0 | 30	New pull request
feature_DE_AD_ALE	Updated 17 days ago by rsanfer		84 | 91	New pull request
feature_output	Updated 19 days ago by economon	✓	33 | 0	New pull request
feature_nlopt	Updated 19 days ago by talbring		35 | 1	New pull request
feature_incompressible	Updated 21 days ago by talbring	✗	84 | 13	New pull request
feature_DE	Updated a month ago by rsanfer		84 | 49	New pull request
feature_FSI_FEA	Updated a month ago by rsanfer		84 | 6	New pull request
feature_HB	Updated a month ago by arubino		126 | 12	New pull request
fix_addGlobalElementIndex	Updated a month ago by rsanfer		141 | 5	New pull request

2. Submitting code to the repository

Fix periodic #294

Merged **economon** merged 38 commits into `develop` from `fix_periodic` on Jul 29

Conversation 3 | Commits 38 | Files changed 35

economon commented on Jul 29 — SU2 code member

This PR contains two items:

1. Improvements to the periodic BC. The periodic BCs are now more stable, especially in parallel.
2. Memory clean up. The RANS solver is now entirely free of memory leaks. I recommend that everyone should run valgrind (http://valgrind.org) on their branches after integrating, in order to check for memory problems and to fix them. While all tests are passing, it is possible that you may see segfaults or other memory problems on your individual branches, now that the class destructors are all active when the code exits. Please let me know asap if you have any problems.

economon added some commits on Jul 3

Fixed a bunch of memory issues and leaks.	5c9bc92
A few fixes for pure serial version.	bbe4441
More fixes.	8149e90
Periodic transform memory.	a83f35f
Adjusted nMarker_Max for overhead.	0a40152
Cleaned up console output for deallocations and fixed issue for cvect…	7927a41
Running Travis on fix_periodic.	✗ 9545d91
Surface and Volume grid movement mem fix.	✗ c6497a6
Fixed memory issues related to the elasticity solver.	✗ 2f84eb3
Fixed memory issues for discrete adjoint. Fixed a few leaks. Fixed wa…	✓ 175bec9

- Submit contributions through pull requests on GitHub.
- Pull requests should target the develop branch.
- Both internal (internal branches) and external developers (external forks).
- Reasons for pull request method:
  - Keeps team informed (emails, PR description, commit logs).
  - Allows for code review (GitHub).
  - Automatic, pre-merge testing (Travis CI).

3. Continuous integration

su2code / SU2 build passing

Travis CI

Current Branches Build History Pull Requests Build #480 More options

✓ Pull Request #294 Fix periodic

#480 passed

Commit 572fa3c

#294: Fix periodic

Thomas D. Economon authored and committed

Elapsed time 37 min 27 sec

Total time 2 hrs 12 min 24 sec

about a month ago

Build Jobs

✓	# 480.1	Compiler: g++ C++	CONFIGURE_COMMAND="./preconfigure.py --pr	31 min 50 sec
✓	# 480.2	Compiler: g++ C++	CONFIGURE_COMMAND="./preconfigure.py --er	29 min 2 sec
✓	# 480.3	Compiler: g++ C++	CONFIGURE_COMMAND="./preconfigure.py --pr	34 min 37 sec
✓	# 480.4	Compiler: g++ C++	CONFIGURE_COMMAND="./preconfigure.py --er	36 min 55 sec

- Pull requests are automatically tested against our suite of regression tests… we know upfront if there are problems and won't merge!
- New features should also include new tests to ensure that the functionality is protected long-term.
- The develop branch is frequently tested automatically, but folks can activate for their own branches while they develop (and change notification to just their own email).

SU2
The Open-Source CFD Code

4. Releases

- We put out releases at frequent intervals:

  SU2 vX.Y.Z where X = major, Y = minor, Z = maintenance

- Released through GitHub (tags) and binaries are created and posted for download on su2.stanford.edu.

- Release schedule is dictated by a combination of features, events (e.g., AIAA for impact), and maintenance needs.

- Feature “hiding” is a practice we use to stage developments and get some early testing for features that aren’t ready for public consumption.

4. Releases

Latest release

v5.0.0

6d6727d

Edit

SU2 version 5.0.0 "Raven"

economon released this 15 days ago

SU2 v5.0.0 contains major new features and improvements, such as the following:

- New in-memory Python wrapping of SU2 using SWIG with accompanying high-level API.
- Class enhancements for multiphysics applications, including interpolation and transfer.
- Free-form deformation (FFD) extensions, including bezier curves and improved usability.
- Reorganization of the incompressible solver for future expansion.
- Harmonic Balance flow analysis capability.
- Algebraic transition model implementation.
- More and better boundary conditions (accuracy and convergence improvements).
- Extensions to scripting for automated database creation (compute_polar.py).
- Critical improvements in I/O, including more feedback to the user.
- Additional bug fixes, stability improvements, and general code maintenance.

The following binary versions are available for download (serial only):

- macOS Sierra 10.12.2: Apple LLVM version 8.0.0 (clang-800.0.38)
- Linux (Redhat 7.0): g++ (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)
- Linux (Ubuntu 16.04): g++ (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609

Download the binaries, source code, and test cases from the SU2 download page:
http://su2.stanford.edu/download.html

Downloads

Source code (zip)

Source code (tar.gz)