

Scientific code development.

Christian Blau

2016-05-19



You implemented a great method. Now you can:



You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**



You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**
 - **People might just opt for the crappy method that is run on some webserver.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**
 - **People might just opt for the crappy method that is run on some webserver.**
- **publish the source code.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**
 - **People might just opt for the crappy method that is run on some webserver.**
- **publish the source code.**
 - **Get people to find and download and compile. Keeping it up to date might be hard.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**
 - **People might just opt for the crappy method that is run on some webserver.**
- **publish the source code.**
 - **Get people to find and download and compile. Keeping it up to date might be hard.**
- **make your tool part of a community.**

You implemented a great method. Now you can:

- **have everybody come to you if they want to apply your method.**
 - **Might get you a lot of second authorships. Lots of work.**
- **publish your paper and have everybody redo the programming for themselves.**
 - **People might just opt for the crappy method that is run on some webserver.**
- **publish the source code.**
 - **Get people to find and download and compile. Keeping it up to date might be hard.**
- **make your tool part of a community.**
 - **Your tool might live long and prosper, even when you're long dead and gone to industry.**



Being scientist and software developer at the same time is hard.



Scientist wants to

Developer wants to



Scientist wants to

- try something and you have no clue where your idea will bring you.

Developer wants to

- know precisely what your code is supposed to do.



Scientist wants to

- try something and you have no clue where your idea will bring you.
- publish and move on.

Developer wants to

- know precisely what your code is supposed to do.
- maintain code for a long time (think 15 years).



Scientist wants to

- try something and you have no clue where your idea will bring you.
- publish and move on.

Thus you

- write more code than you read.

Developer wants to

- know precisely what your code is supposed to do.
- maintain code for a long time (think 15 years).

Thus you

- read more code than you write.



Scientist wants to

- try something and you have no clue where your idea will bring you.
- publish and move on.

Thus you

- write more code than you read.
- are punished for coding slowly.

Developer wants to

- know precisely what your code is supposed to do.
- maintain code for a long time (think 15 years).

Thus you

- read more code than you write.
- are punished for coding ugly.

What we want.

Christian Blau.SciLifeLab.TCB





- Document vigorously



- Document vigorously
- Write tests



- Document vigorously
- Write tests
 - it lets you sleep better.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize
 - break code up into the smallest possible units to perform a very specific task



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize
 - break code up into the smallest possible units to perform a very specific task
 - makes testing easier.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize
 - break code up into the smallest possible units to perform a very specific task
 - makes testing easier.
 - makes code-review easier.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize
 - break code up into the smallest possible units to perform a very specific task
 - makes testing easier.
 - makes code-review easier.
 - makes moving on easier.



- Document vigorously
- Write tests
 - it lets you sleep better.
 - makes you faster in the long run.
 - Testing is documenting. Every test tells you about the supposed outcome of a function.
- Modularize
 - break code up into the smallest possible units to perform a very specific task
 - makes testing easier.
 - makes code-review easier.
 - makes moving on easier.
 - contains bug infections - if you break something, the rest still works

What we really want.

Christian Blau.SciLifeLab.TCB





- 1 read the manual

jenkins.gromacs.org/job/Documentation_Nightly_master/javadoc/



- 1 read the manual

jenkins.gromacs.org/job/Documentation_Nightly_master/javadoc/

- 2 get the code

git.gromacs.org

- all information is in the source ; all interaction is through git



- 1 read the manual

jenkins.gromacs.org/job/Documentation_Nightly_master/javadoc/

- 2 get the code

git.gromacs.org

- all information is in the source ; all interaction is through git

- 3 meta-discussion, no source

redmine.gromacs.org/projects/gromacs



① read the manual

jenkins.gromacs.org/job/Documentation_Nightly_master/javadoc/

② get the code

git.gromacs.org

- all information is in the source ; all interaction is through git

③ meta-discussion, no source

redmine.gromacs.org/projects/gromacs

④ new code

gerrit.gromacs.org

- push to gerrit through git
- play in the drafts branch, only you and reviewers see your code (beware: auto-addition of two reviewers if reviewers left blank)



① read the manual

jenkins.gromacs.org/job/Documentation_Nightly_master/javadoc/

② get the code

git.gromacs.org

- all information is in the source ; all interaction is through git

③ meta-discussion, no source

redmine.gromacs.org/projects/gromacs

④ new code

gerrit.gromacs.org

- push to gerrit through git
- play in the drafts branch, only you and reviewers see your code (beware: auto-addition of two reviewers if reviewers left blank)

⑤ check your coding

jenkins.gromacs.org

- trigger manually when uploading drafts through query and trigger gerrit patches