OPEN SOURCE SOFTWARE DEVELOPMENT

Creating an OSS project

Dr. Igor Wiese
e-mail: igor@utfpr.edu.br

Twitter: @igorwiese

Slides Based on

- https://opensource.guide/starting-a-project/
- Fogel, Karl. **Producing Open Source Software.** https://producingoss.com/
- Steinmacher, Igor; Treude, Christoph; Gerosa, Marco Aurélio. **Let me in: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects**, IEEE Software. http://www.igor.pro.br/publica/papers/IEEESoft_2018.pdf

Starting It

- Use what you have
- Different paths
  - **Own project → Open**
  - **Corporate project → Open**
- Decide the scope and make it clear
  - **What is it about?**
  - **What is it not about?**

Make things clear

- All the knowledge about the project needs to:
  - **Be public**
  - **Be detailed**

- Outsiders need to quickly understand all the details
  - **Design docs**
  - **User manuals**
  - **Next steps (future features)**
  - **How To's**
  - **Code standards**

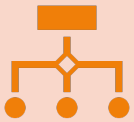
- Reduce the "*hacktivation energy*" *[See Fogel's]*

Name and Surroundings

Choose a name that represents the goal of the project

But easy to say and remember

Look around for trademarks, projects with same name etc.



Having a domain is interesting

checking for the name looking for
the .org, .net, .com also helps looking for already existing things

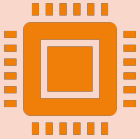
Starting the Readme file

- When creating your page/repository make it clear that the project is free/open
- Create one short paragraph describing the mission of the project

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

List the features and Requirements



What does this software do?

What is planned for it? (Future features)

What do I need to run your project?

OS, compiler, disk space, etc.

Issue Tracker

Public issue tracker is needed

Report bugs

Check the project activity

Contribute

Curating the issue tracker is a constant and infinite task

Version Control

You MUST have your code in a version control system

- **For GitHub it is easy, we know**

It is not only about having it

- **Maintaining**
- **Receiving/reviewing code**
- **Use the same way as ANYONE needs to use**

Create Communication Means

More than the Issue Tracker

Open Source is a collaborative endeavor

Communication is key

Extra ways to communicate are interesting

Slack, Discourse, IRC, mailing list

Challenge!!!!

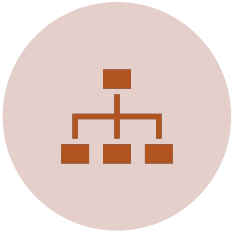
Answering

Empty channels/quiet forums are demotivating

Make It Easy to Contribute

Create a contribution (newcomer-friendly) page or portal

What are the skills needed



Structured documentation (see flosscoach.com)

Give one tutorial-style example of how to do a common task

Maintain a FAQ

Make It Easy to Contribute

Create a contribution (newcomer-friendly) page or portal

Identify and/or dismiss outdated information

Point newcomers to easy tasks

Keep the issue list up-to-date

Make it easy for newcomers to build the system locally

Document the code structure

How?!?

- https://opensource.guide/starting-a-project/

Community profile

Here's how this project compares to recommended community standards.

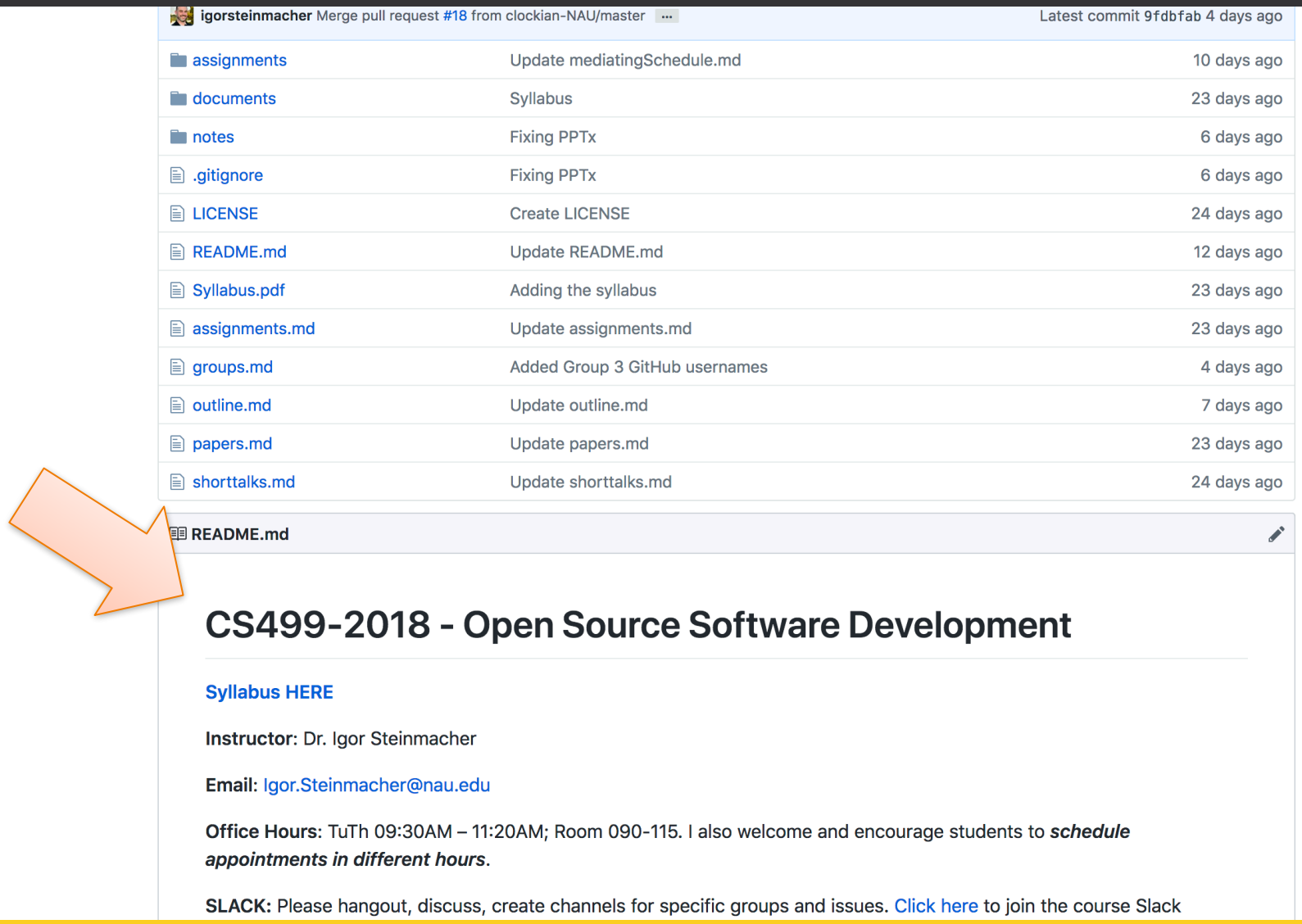
Checklist

- ✓ Description
- ✓ README
- ● Code of conduct — What is a code of conduct? **Add**
- ✓ Contributing
- ✓ License
- ● Issue templates — **Add**

README File

- What does this project do?
- Why is this project useful?
- How do I get started?
- Where can I get more help, if I need it?
- How you handle contributions,
- What are the goals of the project?

README File


igorsteinmacher Merge pull request #18 from clockian-NAU/master ...
Latest commit 9fdbfab 4 days ago
assignments Update mediatingSchedule.md 10 days ago
documents Syllabus 23 days ago
notes Fixing PPTx 6 days ago
.gitignore Fixing PPTx 6 days ago
LICENSE Create LICENSE 24 days ago
README.md Update README.md 12 days ago
Syllabus.pdf Adding the syllabus 23 days ago
assignments.md Update assignments.md 23 days ago
groups.md Added Group 3 GitHub usernames 4 days ago
outline.md Update outline.md 7 days ago
papers.md Update papers.md 23 days ago
shorttalks.md Update shorttalks.md 24 days ago
README.md
CS499-2018 - Open Source Software Development
Syllabus HERE
Instructor: Dr. Igor Steinmacher
Email: Igor.Steinmacher@nau.edu
Office Hours: TuTh 09:30AM – 11:20AM; Room 090-115. I also welcome and encourage students to schedule appointments in different hours.
SLACK: Please hangout, discuss, create channels for specific groups and issues. Click here to join the course Slack

Contributing File

- How to file a bug report
- How to suggest a new feature
- How to set up your environment
- Coding standards/code styles
- Your roadmap or vision for the project
- How contributors should (or should not) get in touch with you

Contributing File

CONTRIBUTING file can be simple (for starters)

FAQ section fits good here

Link to your CONTRIBUTING file from your README

Nice template: https://github.com/nayafia/contributing-template/blob/master/CONTRIBUTING-template.md

Code of Conduct

You can also do

Create Issues and Pull requests templates

Checklist of what is expected for the Pull request or Issue

Communicate what you want the contributors to report/do

Documentation

- [ ] Project has a LICENSE file with an open source license
- [ ] Project has basic documentation (README, CONTRIBUTING, CODE_OF_CONDUCT)
- [ ] The name is easy to remember, gives some idea of what the project does, and does not conflict with an existing project or infringe on trademarks
- [ ] The issue queue is up-to-date, with issues clearly organized and labeled

Code

- [ ] Project uses consistent code conventions and clear function/method/variable names
- [ ] The code is clearly commented, documenting intentions and edge cases
- [ ] There are no sensitive materials in the revision history, issues, or pull requests (for example, passwords or other non-public information)

People

If you're an individual:

- [ ] You've talked to the legal department and/or understand the IP and open source policies of your company (if you're an employee somewhere)

If you're a company or organization:

- [ ] You've talked to your legal department
- [ ] You have a marketing plan for announcing and promoting the project
- [ ] Someone is committed to managing community interactions (responding to issues, reviewing and merging pull requests)
- [ ] At least two people have administrative access to the project