How exactly do we teach students the right FOSS etiquette?

Ashutosh Pandey

Akash Singh



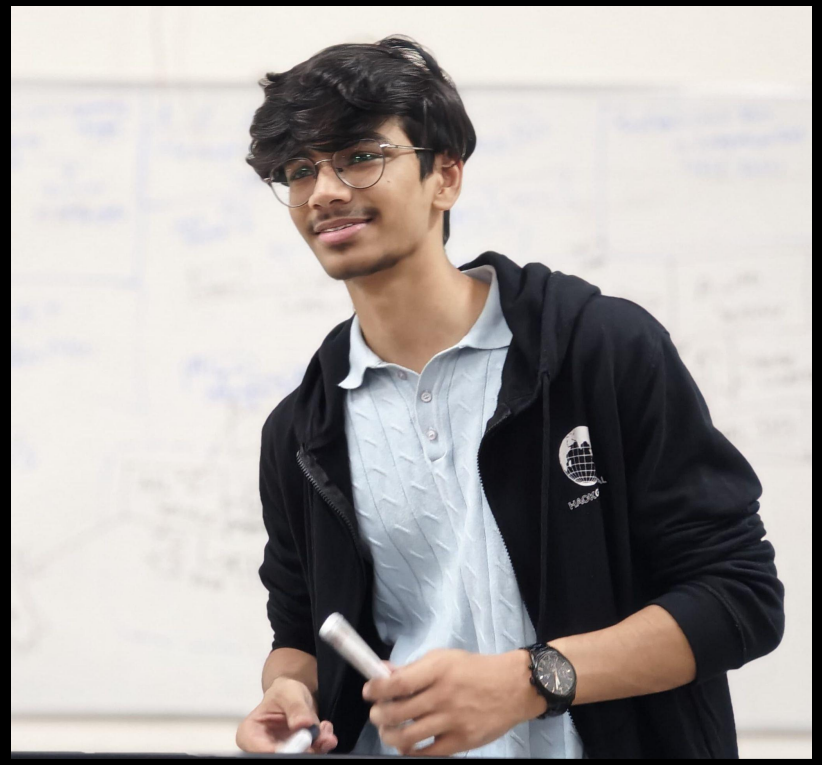
pointblank.club/oss

INDIA FOSS 2026

<. >
Point Blank

SPEAKER PROFILE

<. > Point Blank



AKASH SINGH

- Leads Devops, Infrastructure and MLOps at SABI
- Lead of the Point Blank Open Source Community
- Google Summer of Code 2024 Contributor
- Linux Foundation Mentorship 2025
- Golden Kubestronaut
- Organizes the Cloud & DevOps Devroom @ IndiaFOSS

SPEAKER PROFILE



ASHUTOSH PANDEY

- Compiler Engineer at AMD
- Co-organizes LLVM Social Bangalore
- Google Summer of Code Contributor 2020
- Linux Foundation Mentee 2021
- Founder of Point Blank
- Organizes the Compilers, Programming Languages and Systems Devroom at IndiaFOSS

TARGET AUDIENCE

FOSS Club Leads

Student Developer Communities

Mentors introducing students to FOSS for the first time

First time contributors

Organizers of programmes like Hacktoberfest, GSoC, LFX, Outreachy

THE GAP

Too much focus on ‘How’ not enough on ‘Why?’

WHAT EVERY TUTORIAL COVERS

- How to find an issue?
- How to fork a repository?
- How to create a branch?
- How to submit a pull request?
- How to apply for GSoC or LFX?

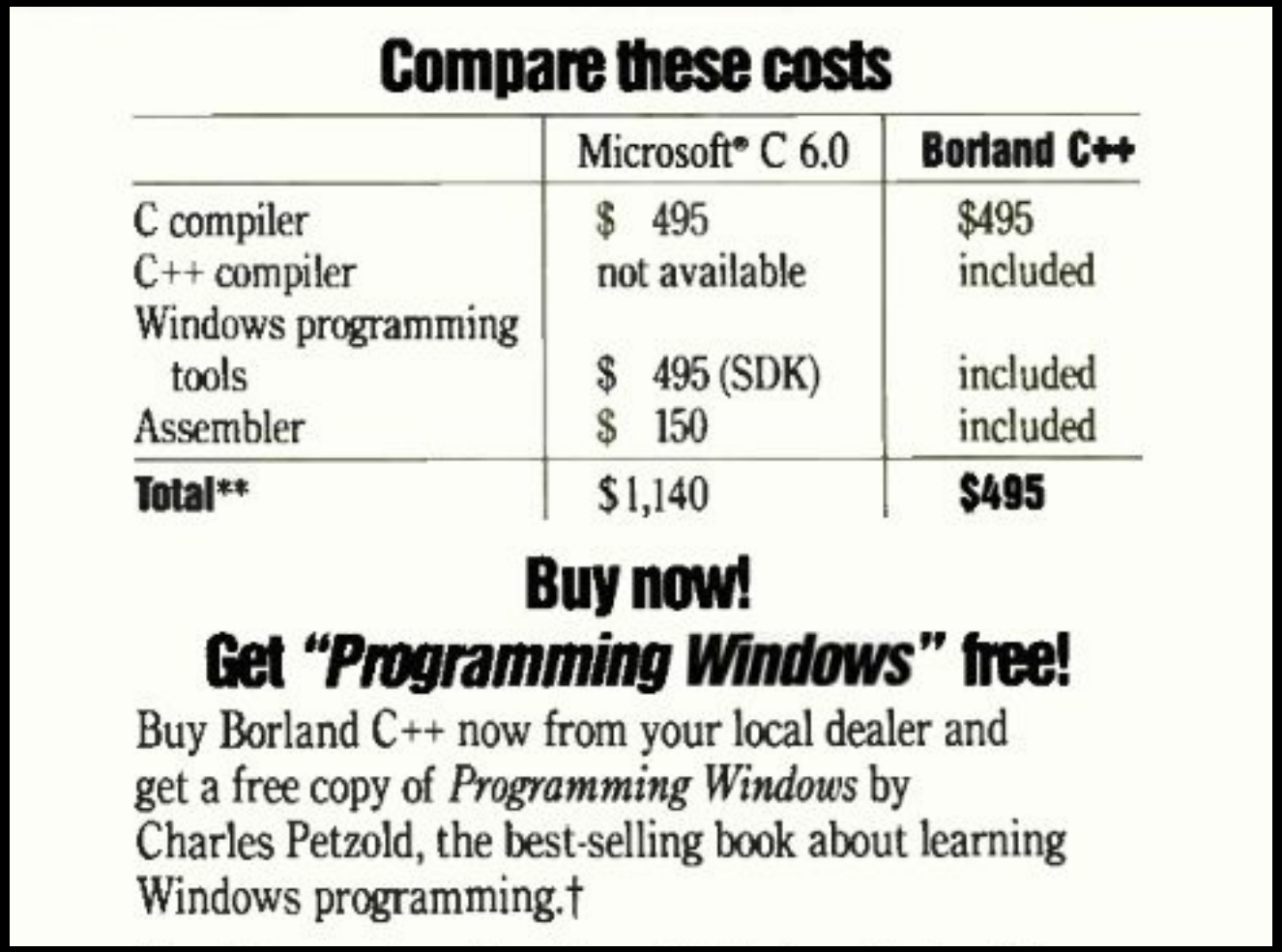
WHAT THEY DON'T

- When it is okay to **ping** a maintainer?
- How to ask for **help** correctly?
- How to take a **rejection**?
- How to get **issues** assigned?
- How to **use AI** without violating org policies?

What to Do? *Versus* *How to Behave?*

We Begin with a history lesson

Students inherit compilers, operating systems, browsers, languages, editors and tools they can read and change. **None of that was inevitable.**



Compare these costs

	Microsoft® C 6.0	Borland C++
C compiler	$ 495	$495
C++ compiler	not available	included
Windows programming tools	$ 495 (SDK)	included
Assembler	$ 150	included
Total**	$1,140	**$495**

Buy now!

Get "*Programming Windows*" free!

Buy Borland C++ now from your local dealer and get a free copy of *Programming Windows* by Charles Petzold, the best-selling book about learning Windows programming.†

1983

Richard Stallman announces the **GNU** Project.

1992

Linus Torvalds releases the Linux kernel under the GPL.

WHAT FREE SOFTWARE ACTUALLY MEANS

The four essential freedoms

The LICENSE File is more than just GitHub filler

0 **Run the program**

1 **Study and modify**

2 **Redistribute copies**

3 **Distribute modified versions**

BEFORE THE FIRST COMMIT

Teach gratitude, not entitlement

FOSS IS NOT FREE CODE ON THE INTERNET. IT IS PEOPLE.

- Volunteers
- Paid engineers
- Maintainers
- Researchers
- Docs writers
- Testers
- Translators
- Organisers
- Funding companies

WHAT THEY OFTEN THINK

“This project is public, so the maintainers owe me attention.”

WHAT WE TEACH THEM TO THINK

“This project is public because people chose to share it. How do I take part without adding to their load?”

Connect ethics to consequences

FOSS communities run on trust. Damaging it can have serious consequences for your career

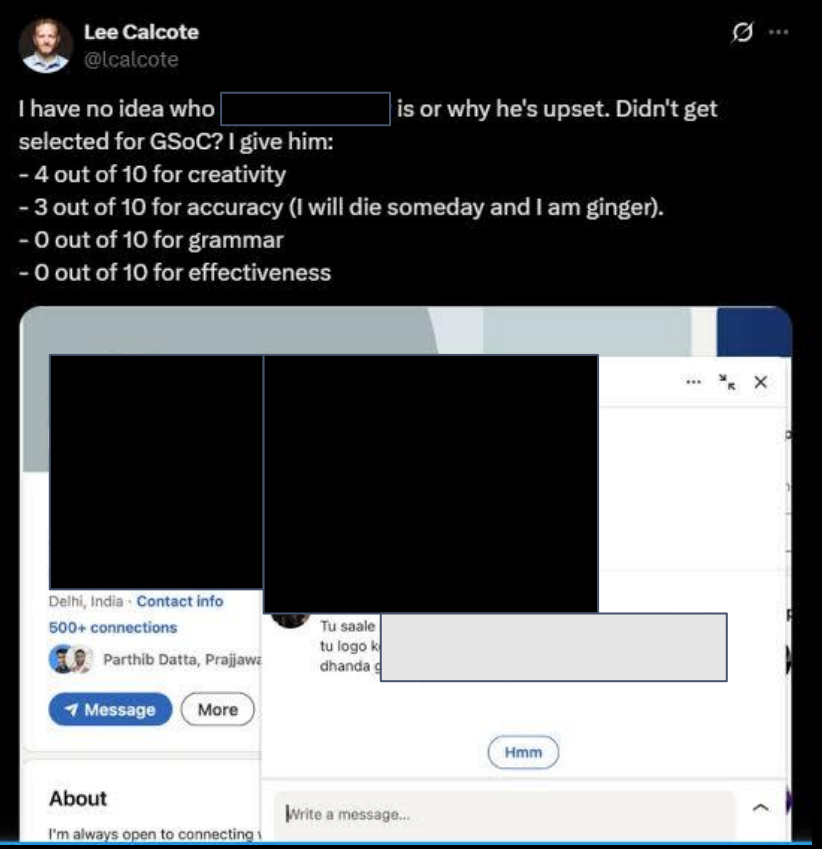
THE BELIEF

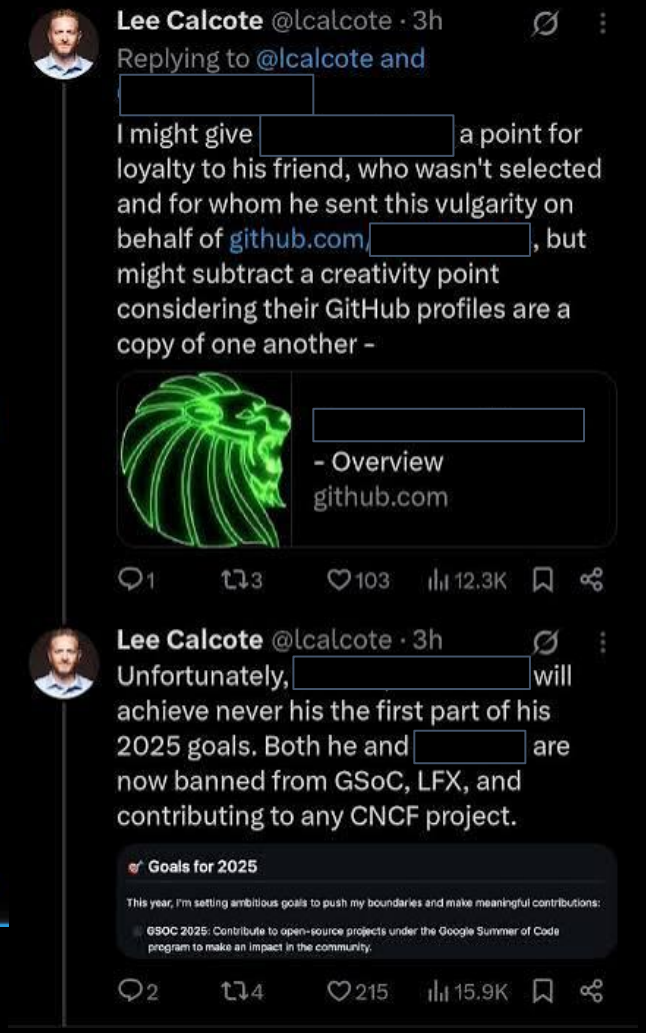
- Getting selected is all that matters
- Being noticed beats being useful
- Aggressive persistence proves commitment
- A shallow contribution beats none
- The ends justify the means

THE COST

- Maintainers stop trusting you
- Legitimate future work gets extra scrutiny
- Other contributors avoid collaborating
- You represent more than just yourself
- Projects close themselves to newcomers
- Your digital footprint is permanent

Handle Rejection Gracefully



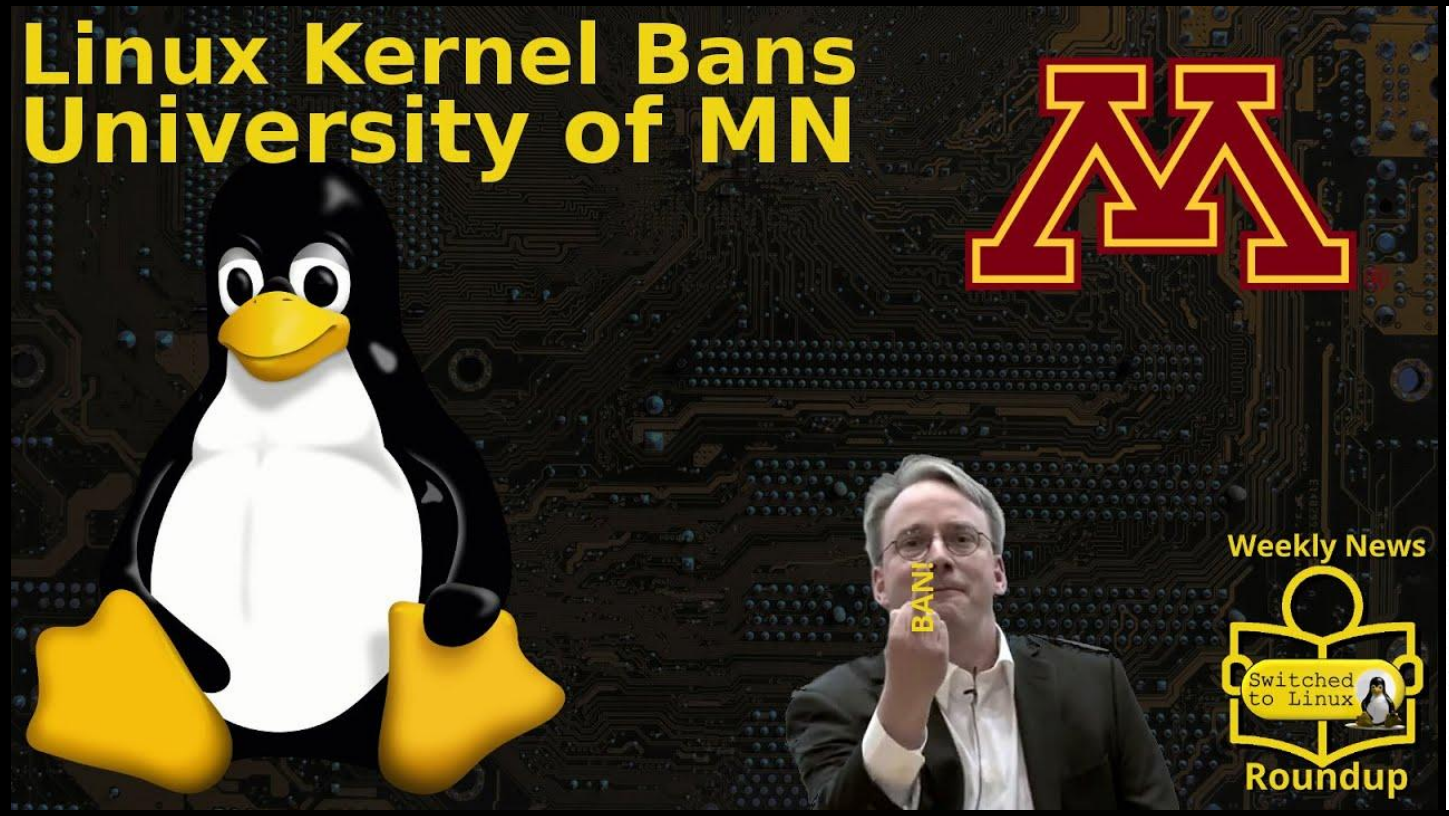


WHAT STUDENTS SHOULD TAKE FROM IT

- Rejection **never** justifies harassment
- Private messages become public
- Acting “for a friend” does not remove personal responsibility
- One emotional message can take down more than one person
- Communities remember how you behave
- Consequences can be **permanent**

One message got two people banned from GSoC, LFX and CNCF for Life

Good Intent and Good Faith alone are not enough



LKML Archive on lore.kernel.org

search | help / color / Atom feed

From: Kangjie Lu <kjlu@umn.edu>
To: open list <linux-kernel@vger.kernel.org>
Cc: Qiushi Wu <wu000273@umn.edu>, Aditya Pakki <pakki001@umn.edu>
Subject: An open letter to the Linux community
Date: Sat, 24 Apr 2021 17:30:50 -0500
Message-ID: <CAK8KejpUVLxmqp026JY7x5GzHU2YJLPU8SzTZUNXU2OXC70ZQQ@mail.gmail.com> (raw)

April 24, 2021
An open letter to the Linux community

Dear Community Members:

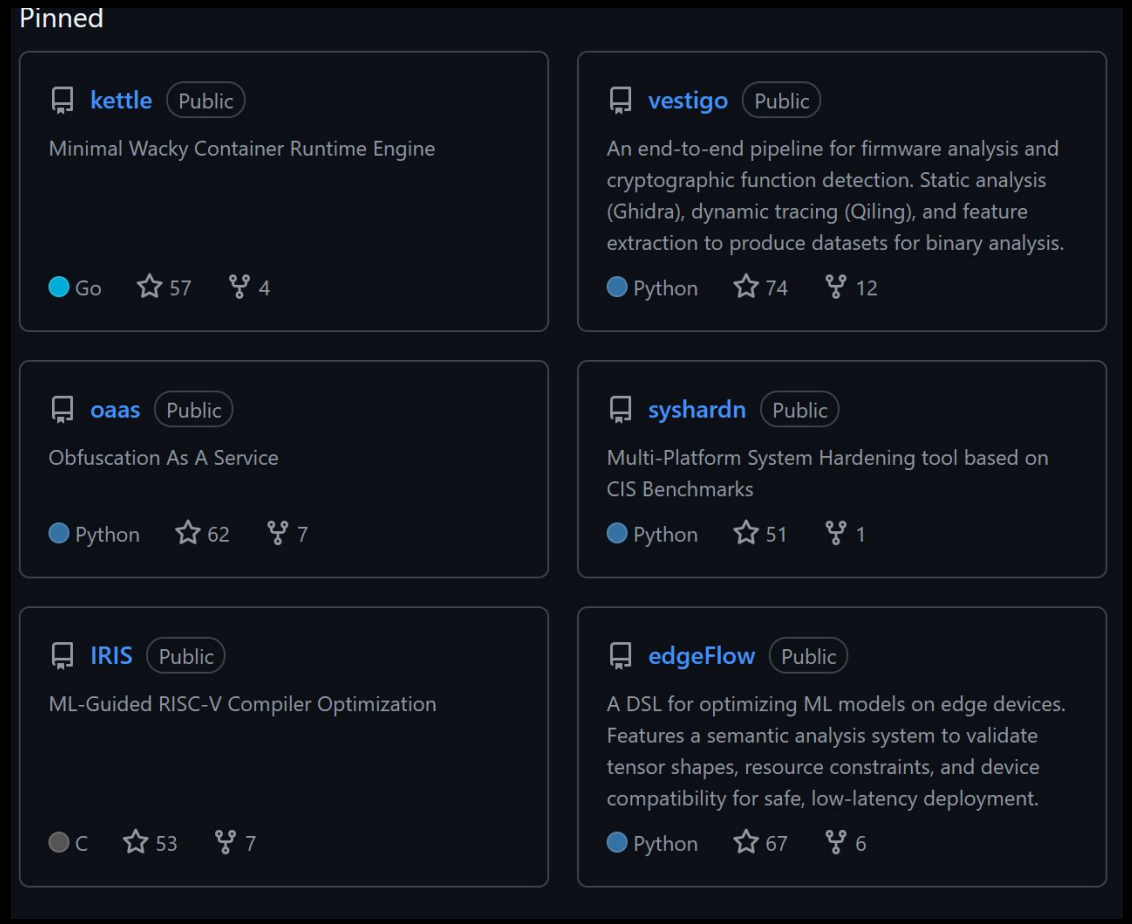
We sincerely apologize for any harm our research group did to the Linux kernel community. Our goal was to identify issues with the patching process and ways to address them, and we are very sorry that the method used in the "hypocrite commits" paper was inappropriate. As many observers have pointed out to us, we made a mistake by not finding a way to consult with the community and obtain permission before running this study; we did that because we knew we could not ask the maintainers of Linux for permission, or they would be on the lookout for the hypocrite patches. While our goal was to improve the security of Linux, we now understand that it was hurtful to the community to make it a subject of our research, and to waste its effort reviewing these patches without its knowledge or permission.

The Blast Radius for bad behaviour can be huge

Build a playground before you send them out

No student should go from a single Git workshop straight into a large public project. They need somewhere mistakes can be corrected without spending a stranger's time.

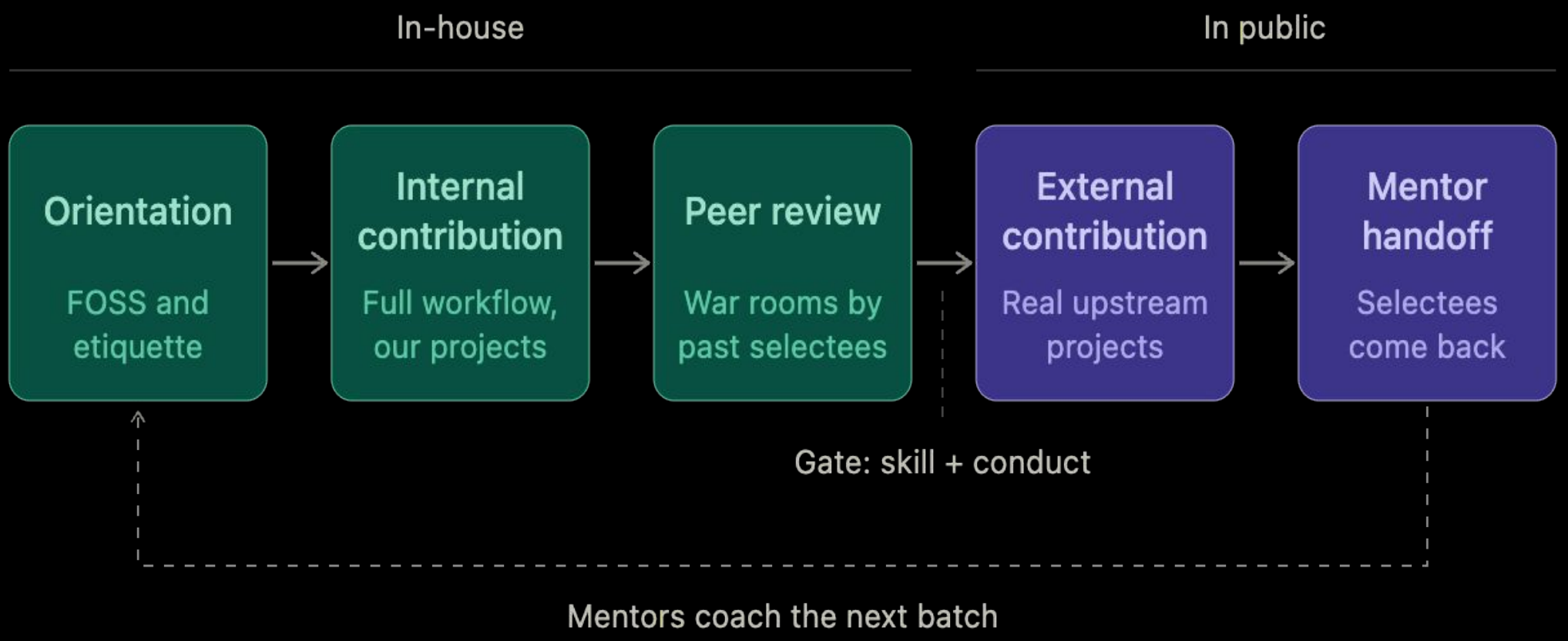
THEY START ON OUR OWN REPOS



FORGIVING ENVIRONMENT. REAL PROCESS.

- Read the contribution guidelines
- Claim or discuss work properly
- Write meaningful issues
- Submit focused pull requests
- Pass CI
- Respond to review and revise
- Wait, instead of demanding

Five stages - In Order



A QUICK RECAP ABOUT POINT BLANK

+ A student-run Open Source Community. Several peers and Alumni with a long history of OSS contributions.

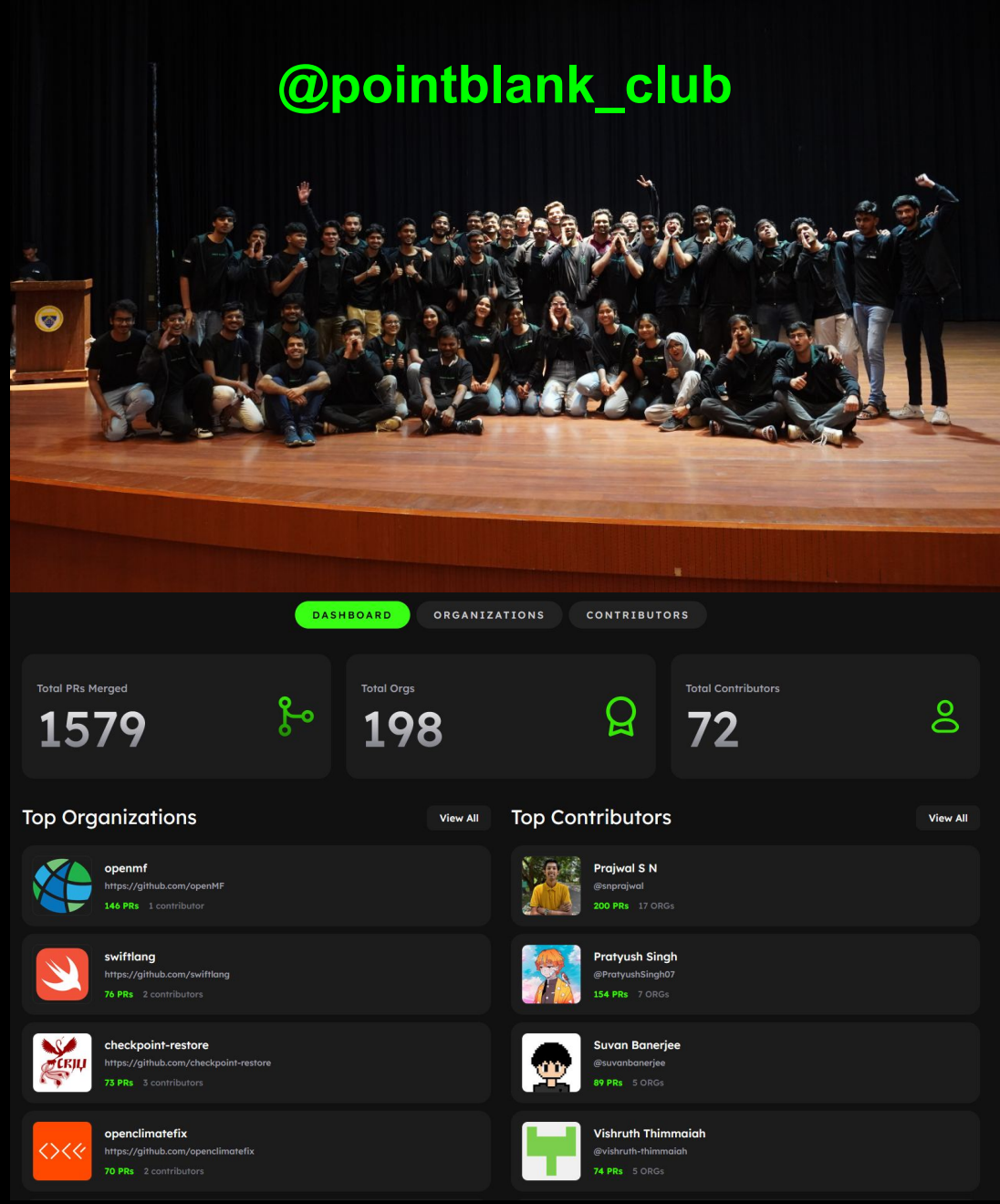
+ Nobody handed us a roadmap. Seniors showed us where to look and how to ask.

+ Community always comes before code.

+ Point Blank is open to all branches. No CGPA requirement. Not just a ‘CS’ only club. **Diverse representation** of students.

+ All of our talks and events are open to all and cross posted on our socials (Blog, YouTube).

@pointblank_club



pointblank.club/oss

THE CURRICULUM, PART ONE

Teach the basic rules out loud

1. Read before you write
2. Learn what the project prioritises
3. Do not demand attention



Maintainers are collaborators, not customer-support agents.



4

Do not self-assign without permission

- Follow the project's system.
- "I want this" does not reserve an issue forever.

5

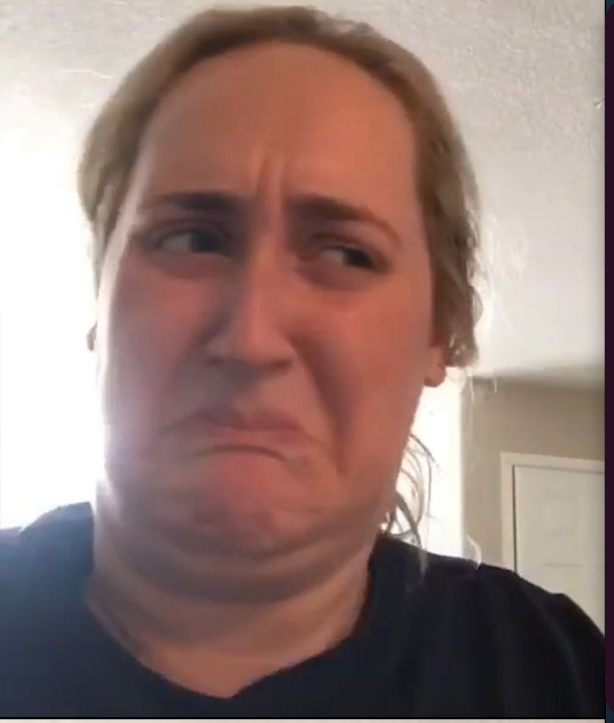
Do not duplicate active work

- Check assignees and recent comments.
- Look for an **open PR first**.
- **Ask** before taking over abandoned work. Never race someone mid-way.

6

Ask in the right order

- **Docs** and **past discussions** first.
- Then experienced people in your club.
- Then the maintainers, clearly and in public.



AI SLOP

- Code you cannot explain
- Fabricated benchmarks, or APIs that do not exist
- Generic issue comments written to look active
- Large unfocused PRs you have not read yourself
- Replies that do not answer the maintainer’s question
- Proposals full of claims you cannot defend



AI SMART

- Understanding unfamiliar code
- Generating test cases
- Explaining error messages
- Improving documentation
- Exploring possible implementations
- Reviewing your own work before you submit

AI does not transfer responsibility

Your code is your responsibility, however it was produced.

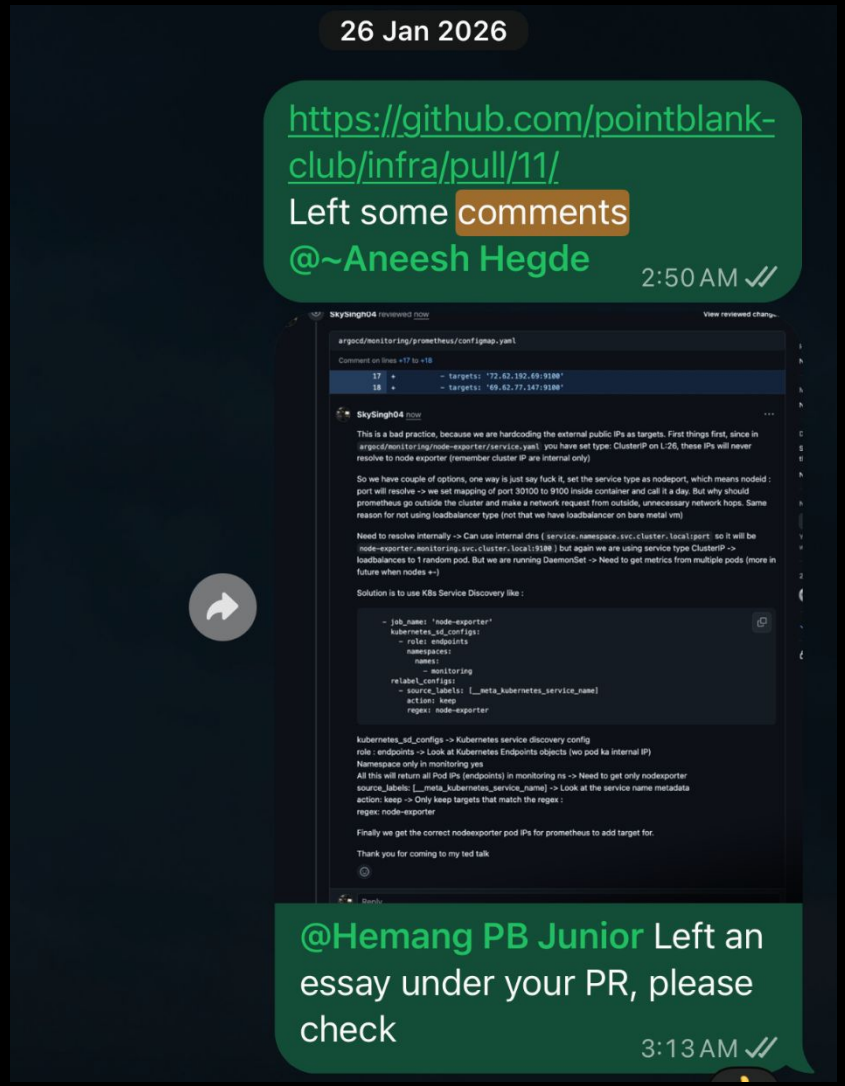
We Coach them through rejection

WHY GOOD WORK STILL GETS CLOSED

- It does not fit the project's direction
- It raises long-term maintenance cost
- A solution already exists
- The timing is wrong
- The contributor misread the problem
- The maintainer has no review capacity right now

HOW TO ANSWER

- Ask for clarification before arguing
- Thank the reviewer for their time
- Separate criticism of the work from criticism of you
- Revise when revision is the right answer
- Accept that some decisions will not be reversed
- Move on without attacking the project or the maintainer



Sad day for Aneesh and Hemang

Contribution quality beats programme branding

Students treat **GSoC** and **LFX** as résumé badges. THEY ARE NOT

WHAT THEY ACTUALLY ASK

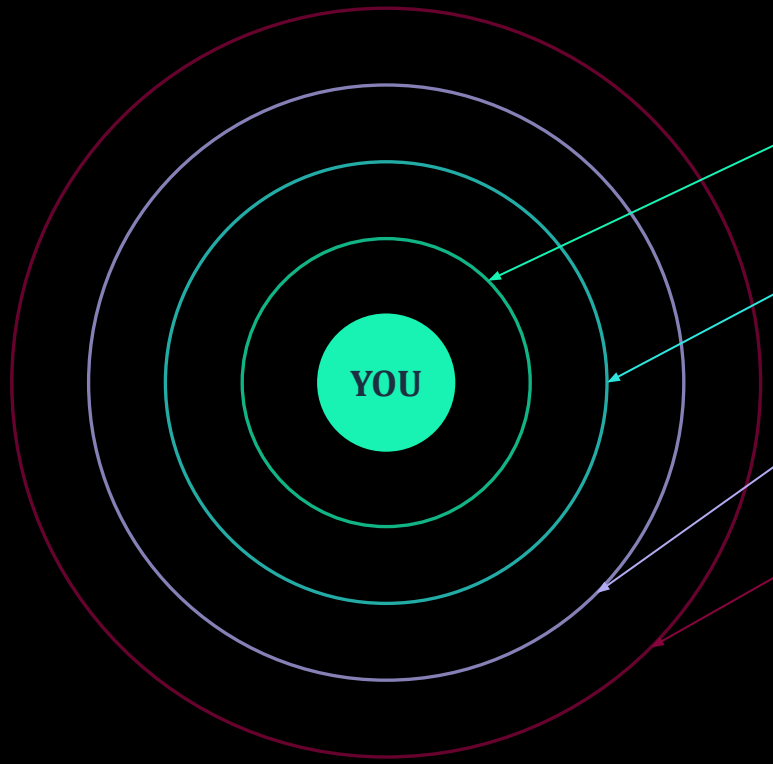
- What exactly did you build?
- Which technical decisions were yours?
- What was merged?
- How did you respond to review?
- Did you keep contributing after the stipend stopped?
- Can the maintainers confirm your role?

WORTH JUST AS MUCH

- Maintaining a project for years
- Fixing hard bugs outside any programme
- Repairing neglected documentation
- Reviewing other people’s contributions
- Building release and CI infrastructure
- Growing a community around something you made

AT POINT BLANK, WE TEACH PEOPLE TO BE OSS CONTRIBUTORS, NOT RUN AFTER THE TAG

Your behaviour represents more than you



- Your FOSS club
- Your college and your mentor
- Your regional community
- Indian students and developers, broadly

It is unfair. It is also how communities actually update their priors i.e. one student's behaviour makes the next student's contribution harder to trust.

A repeatable model for FOSS clubs

1. **Teach history and values**
   Why software freedom and maintainer labour matter.
2. **Publish an explicit etiquette guide**
   Do not assume unwritten norms transmit themselves.
3. **Provide internal practice repos**
   Mistakes, supervised.
4. **Review public communication**
   Read issues and messages before they are posted.
5. **Run a proposal-review gate**
   No proposal reaches a maintainer as an unreviewed draft.
6. **Require contribution before application**
   A relationship, not a March résumé sprint.
7. **Teach rejection explicitly**
   Role-play closed PRs and declined proposals.
8. **Build a mentor handoff**
   Selected contributors train the next cohort.
9. **Discuss failure cases openly**
   Spam, plagiarism, abandonment, programme chasing, AI misuse.
10. **Reward maintainers, not only the selected**
    Celebrate review, docs, ownership and long-term upkeep.

FRAME THIS SLIDE

The pre-submission checklist

- [ ] Have I read the contribution guidelines?
- [ ] Have I searched for previous discussions?
- [ ] Is someone already working on this?
- [ ] Does this address a real project need?
- [ ] Can I explain every part of what I am submitting?
- [ ] Have I tested it properly?
- [ ] Does it pass CI and formatting checks?
- [ ] Is my message clear, respectful and short?
- [ ] Am I creating more work than value?
- [ ] Am I ready to accept rejection professionally?
- [ ] Have I attributed other people's work?
- [ ] **Would I be comfortable if this were publicly tied to my name?**

FRAME THIS SLIDE

Etiquette is not a soft skill bolted on after Git

A contributor who writes excellent code but cannot communicate, coordinate or accept review will not last in a community project.

Our job as mentors is not more pull requests, more proposals or more selections.

It is **creating contributors** maintainers are glad to work with.

Teach the commands. Teach the workflow. Then teach them how to be trustworthy members of a community.

> ***Before you send a student out: "Would I be happy maintaining a project this student contributes to?"***

THANK YOU

ASHUTOSH PANDEY

linkedin.com/in/ashupdsce · X: @ashpande18

AKASH SINGH

linkedin.com/in/skysingh04 · X: @skysingh04

SEE YOU AT THE CLOUD AND DEVOPS DEVROOM TODAY

2:00PM ONWARDS