

# Lesson 3: Healthy open source software projects (On-ramps and practices)

Stephen R. Walli

# Objectives

- Understand the general roles in a project community
- Understand the simple community dynamic in community
- Understand the on-ramps and activities successful project communities display

## Notes

<https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/lessons/3#notes>

# What does success mean?

- Inbound
  - A healthy contribution flow
- Outbound
  - Did someone star a GitHub repo because they are a simple end user, or because they follow the developer, or because they are a developer using the source?
  - Did someone clone/fork a repo and use it, or abandon the project fork, or simply review and learn from it?

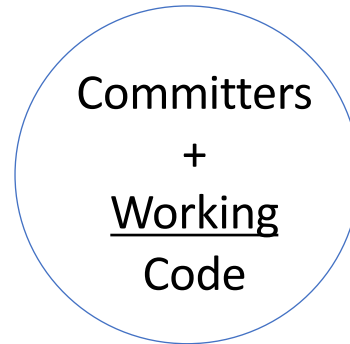
# Roles

- Maintainer: a primary author of a project with full privileges to write the project directory tree.
- User: any person that is using a software project for its intended purpose.
- Developer: any person using a software project but furthermore also modifying the source to their own needs.
- Contributor: any person offering a direct artifact back to the project, including source code patches, bug reports, configuration, documentation.

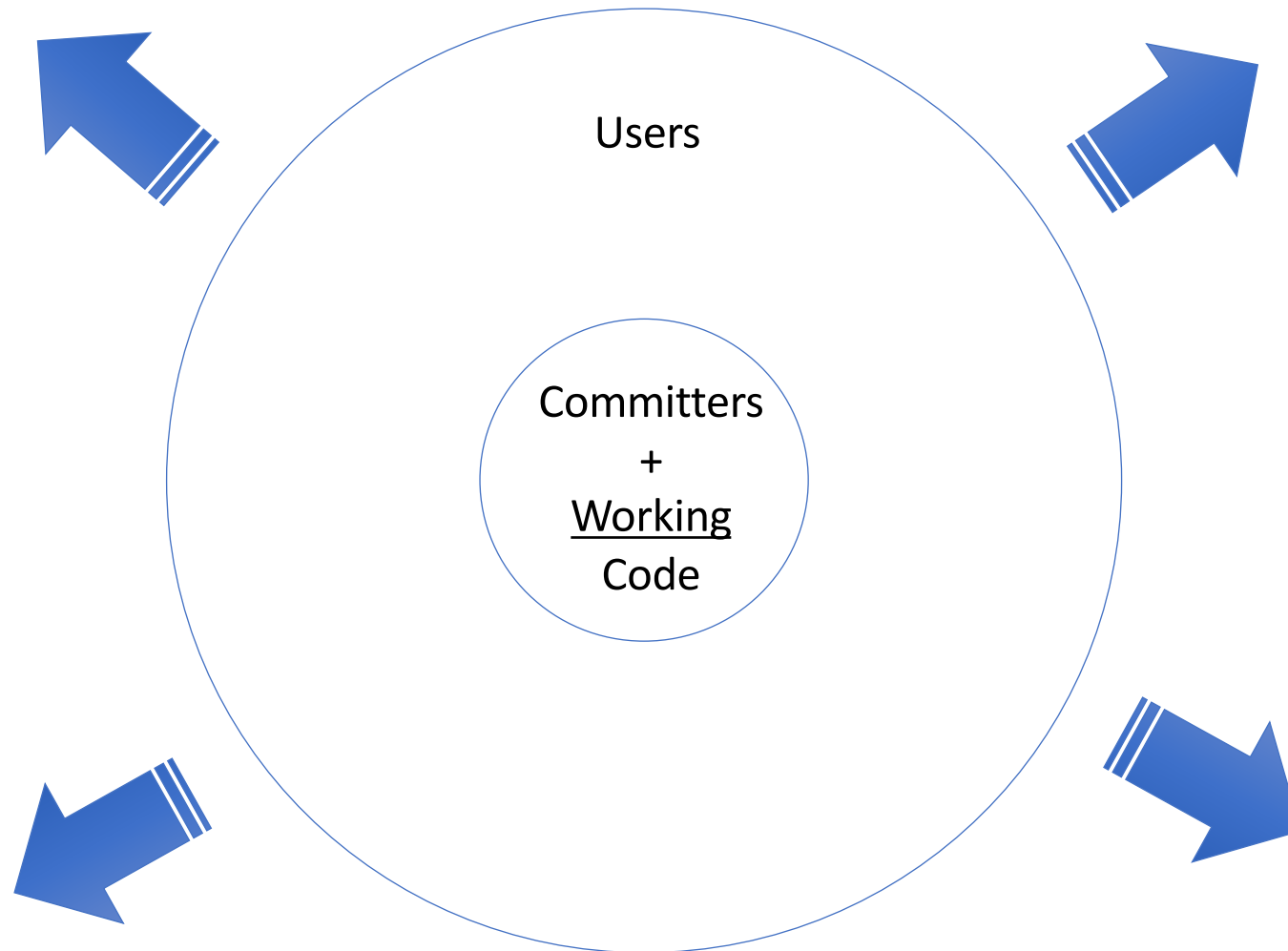
# Contributions

- Code – fixes and new functionality
- Bug reports indicate a new test path or use
- New configuration information broadens the user base
- Documentation (answering forum questions, creating tutorials, etc.) broadens the user base
- Forum time answering questions
- Translations

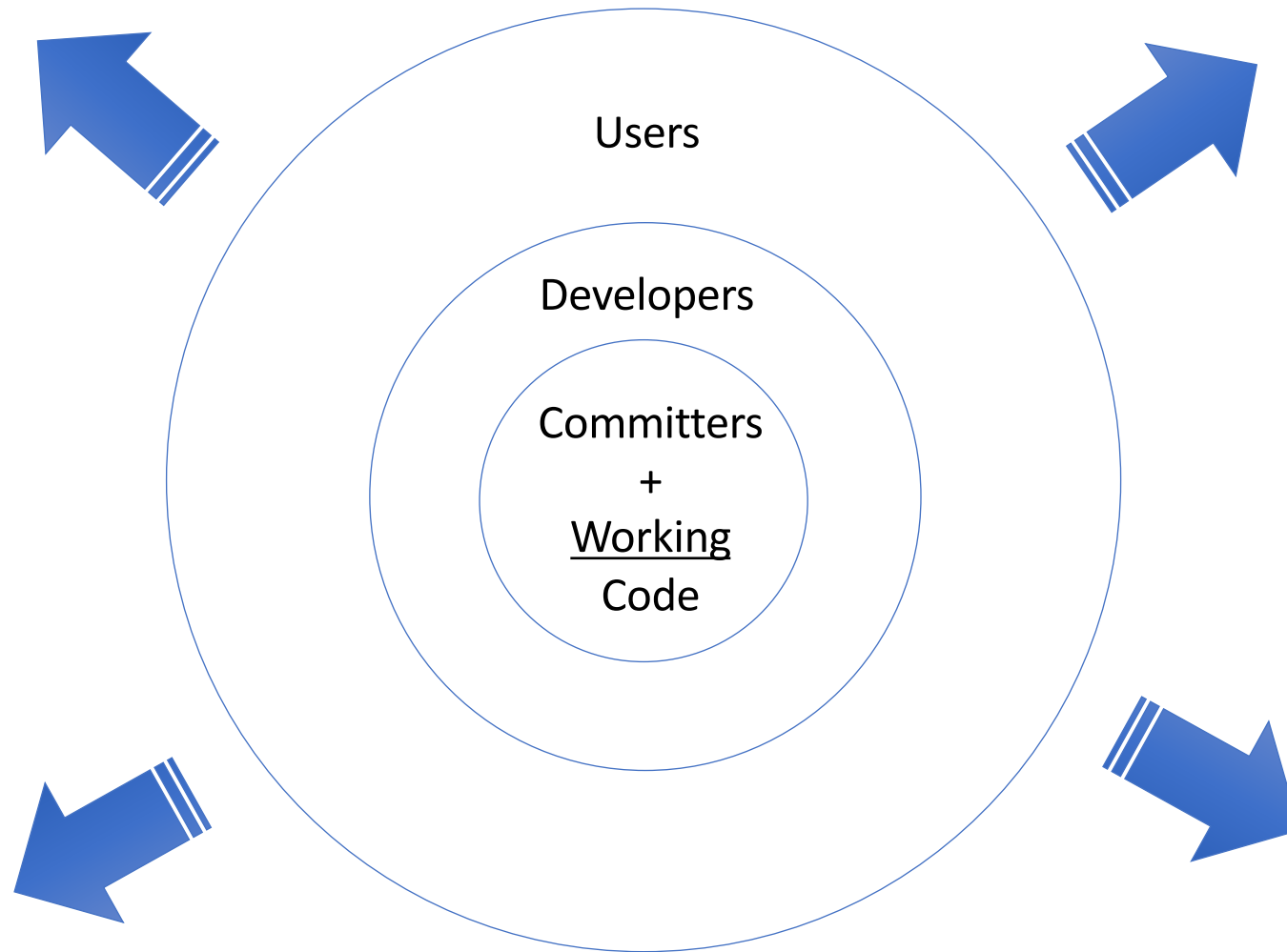
# Three On Ramps Need to be Built



# Three On Ramps Need to be Built

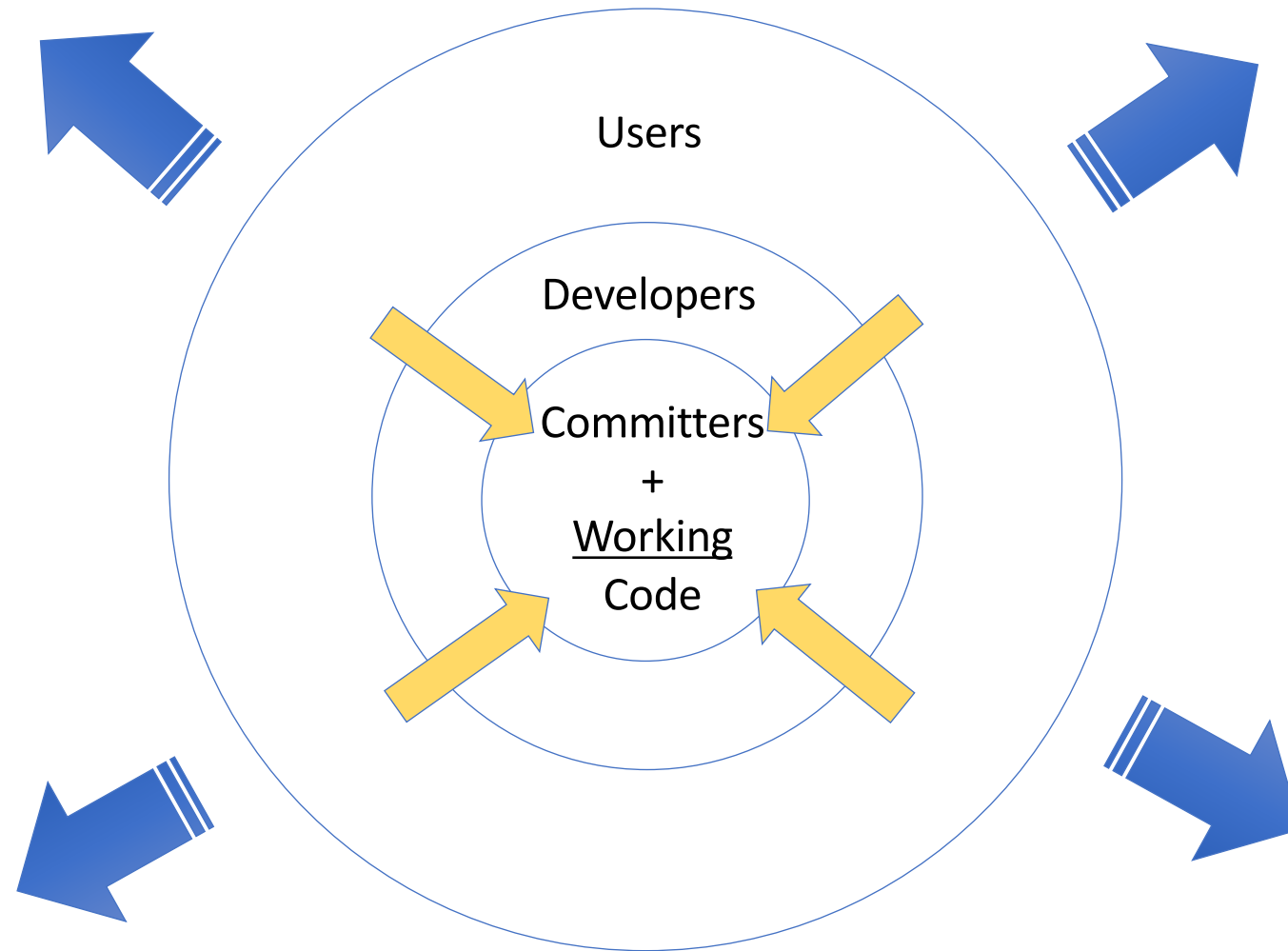


# Three On Ramps Need to be Built

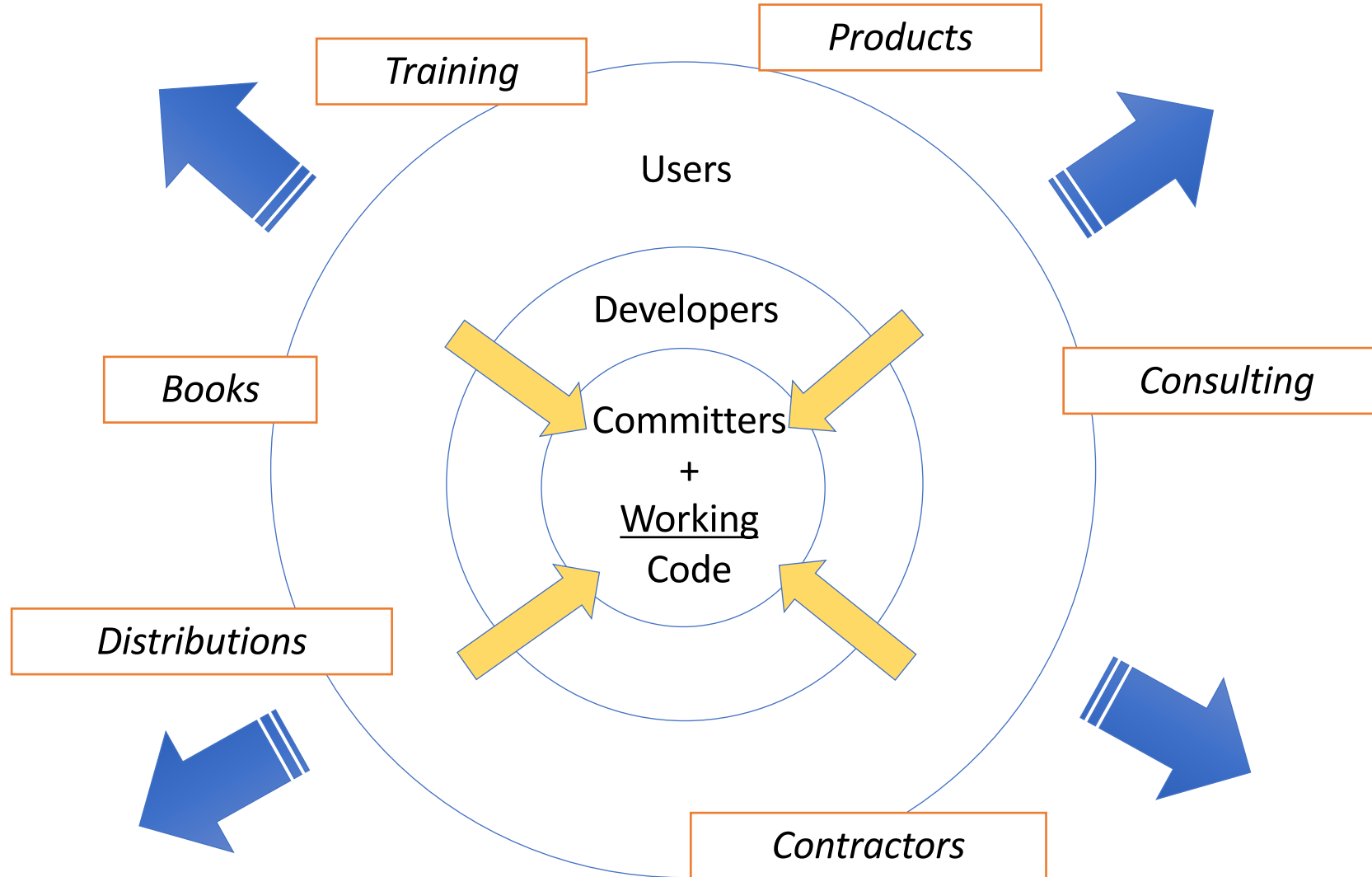




# Three On Ramps Need to be Built



# Three On Ramps Need to be Built



# Three On Ramps for Community Building

**How do you encourage people to use your project?**

(Because that's where you'll find bugs reports & developers)

**How do you encourage developers **selfishly** to experiment?**

(Because these are your future contributors)

**How do you encourage developers to share their work?**

(Because contribution flow is the growth and success of your project)

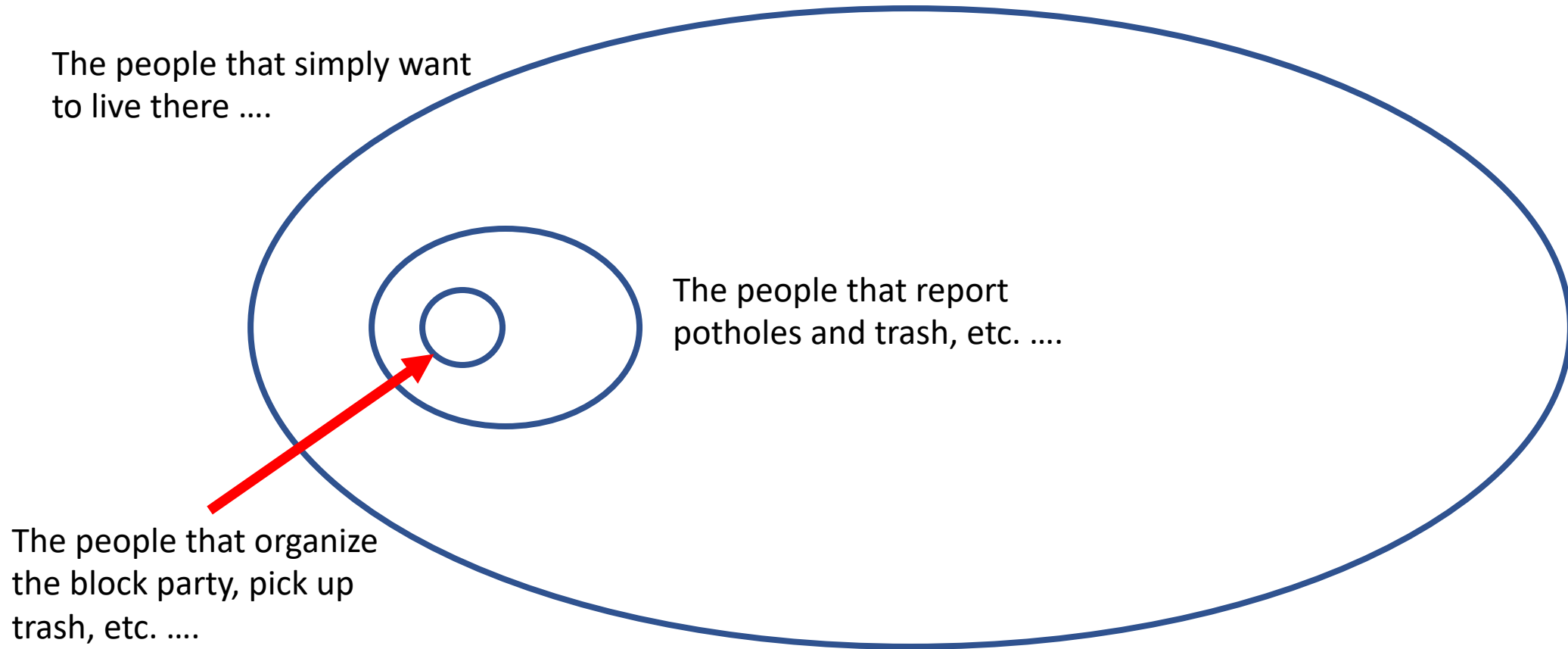
# Side Note:

- Nobody cares about your community
- Your project solves a problem for a user or a developer
- They want to selfishly use your work ... that you liberally licensed to allow them to do just that.
- You can have NO expectations in return
- It is your job as maintainer to make it easier to do the right thing economically and contribute back to the project

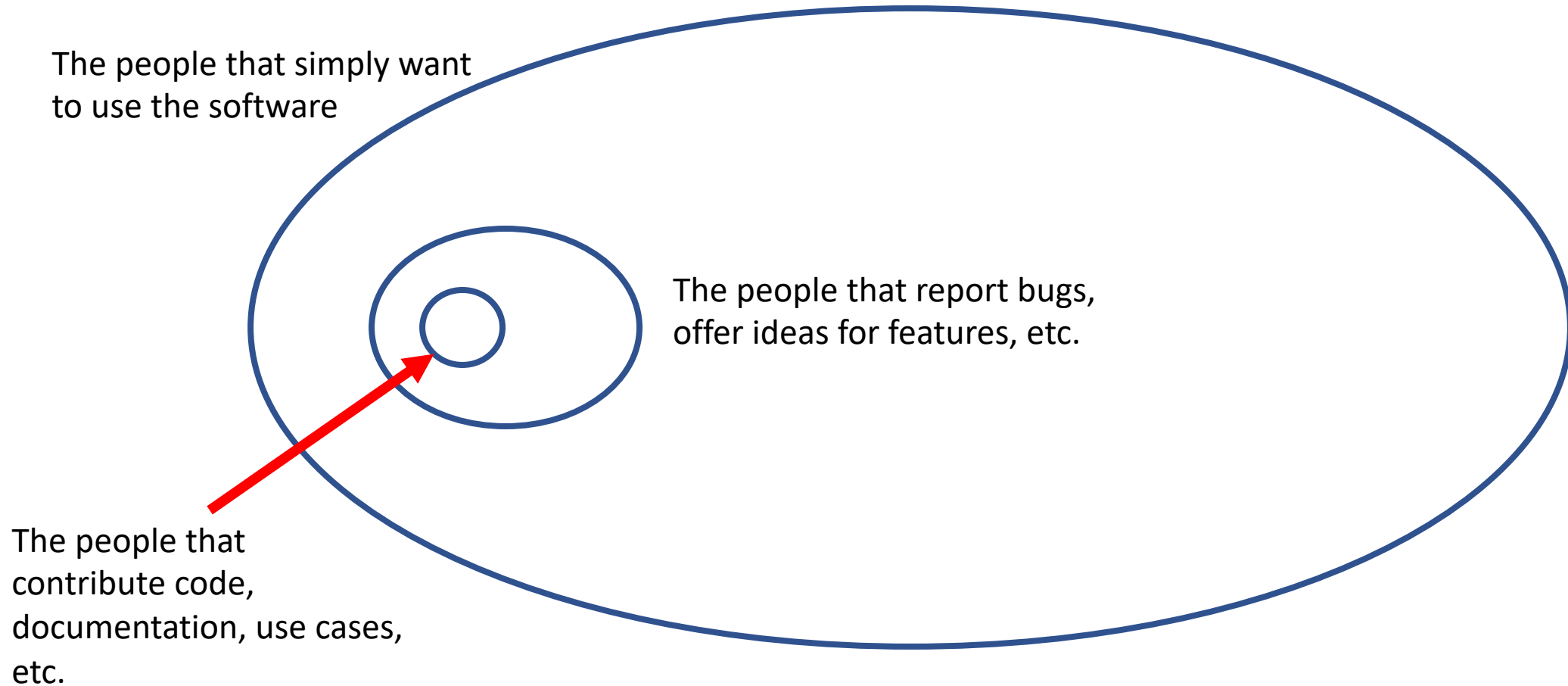


In the World of  
Atoms: You choose  
your neighborhood  
for very personal  
reasons

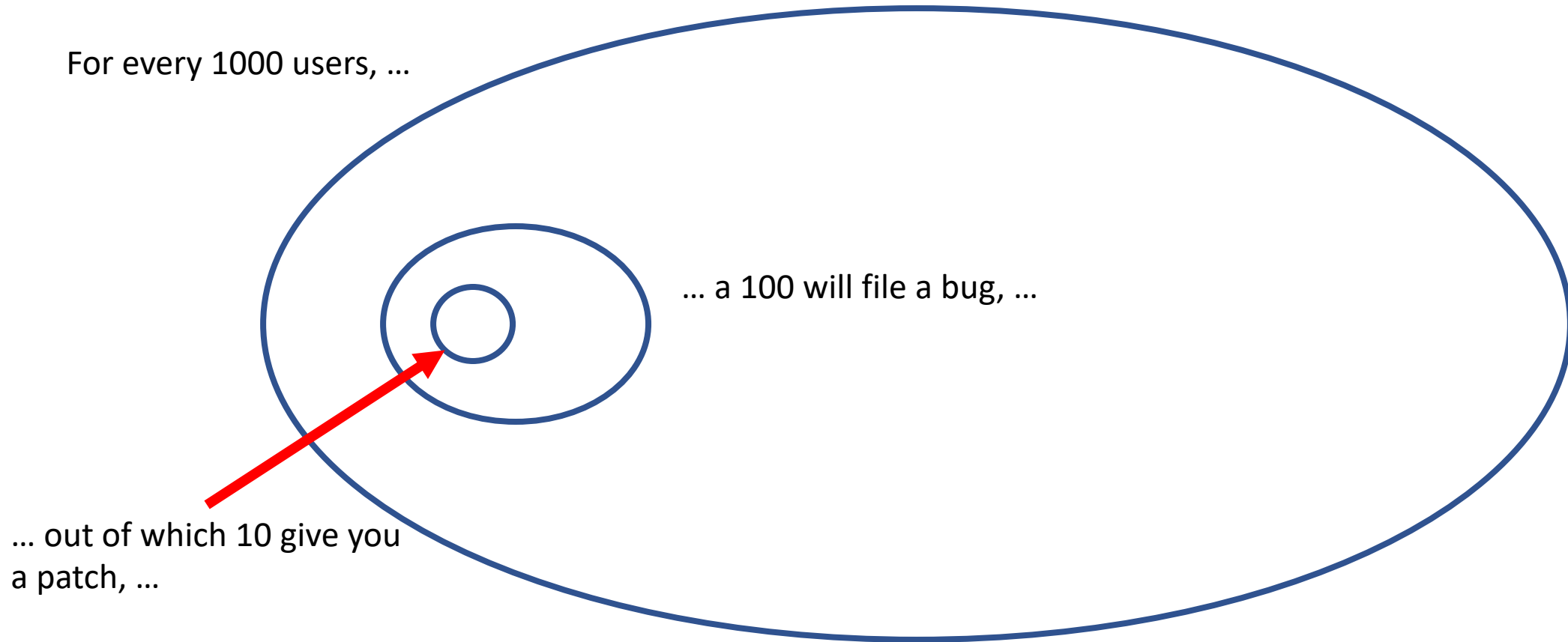
# Three Sorts of Neighbours in Your Community



# Three Sorts of People in Your Project Community

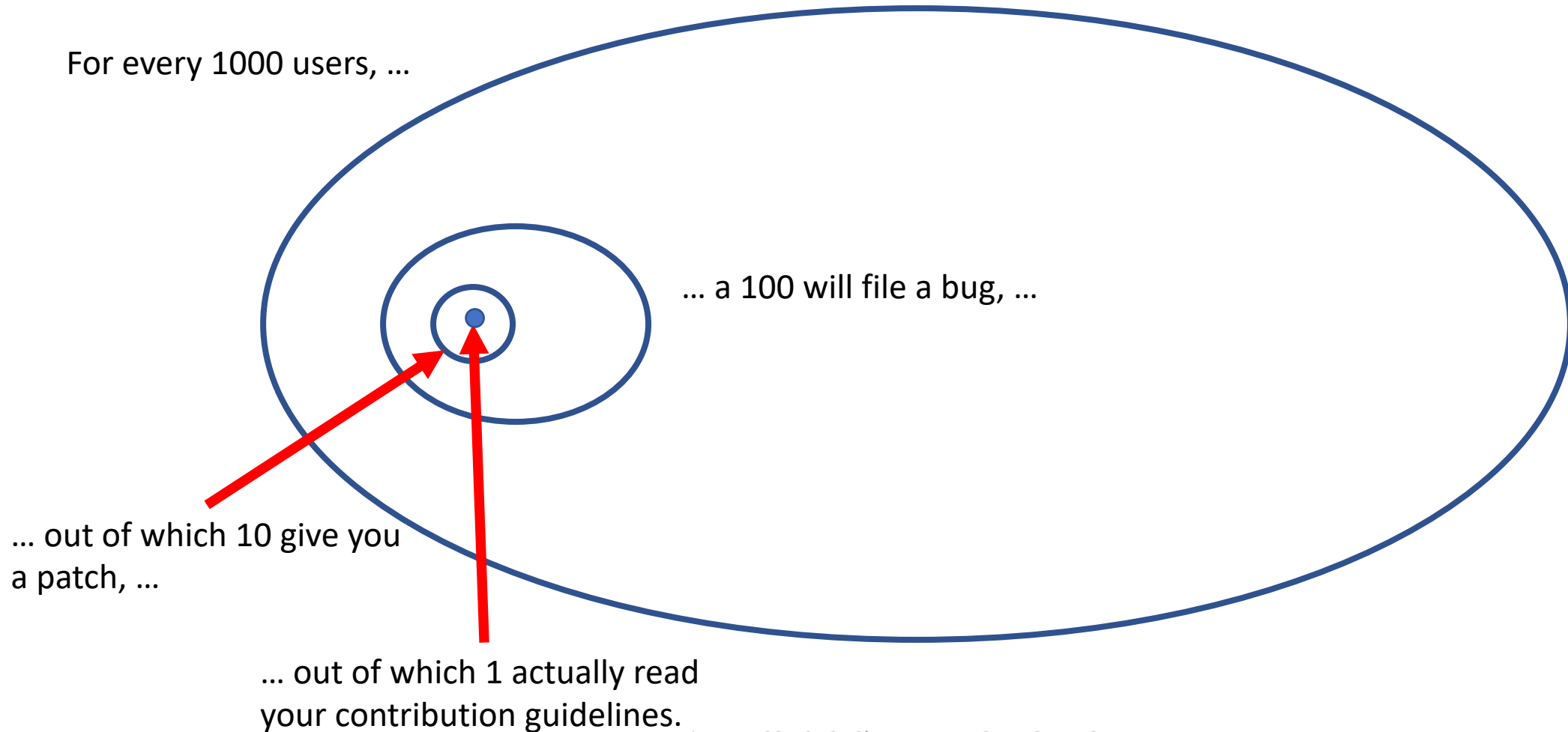


# Rules of Thumb and Orders of Magnitude





# Rules of Thumb and Orders of Magnitude



# Rules of Thumb and Orders of Magnitude

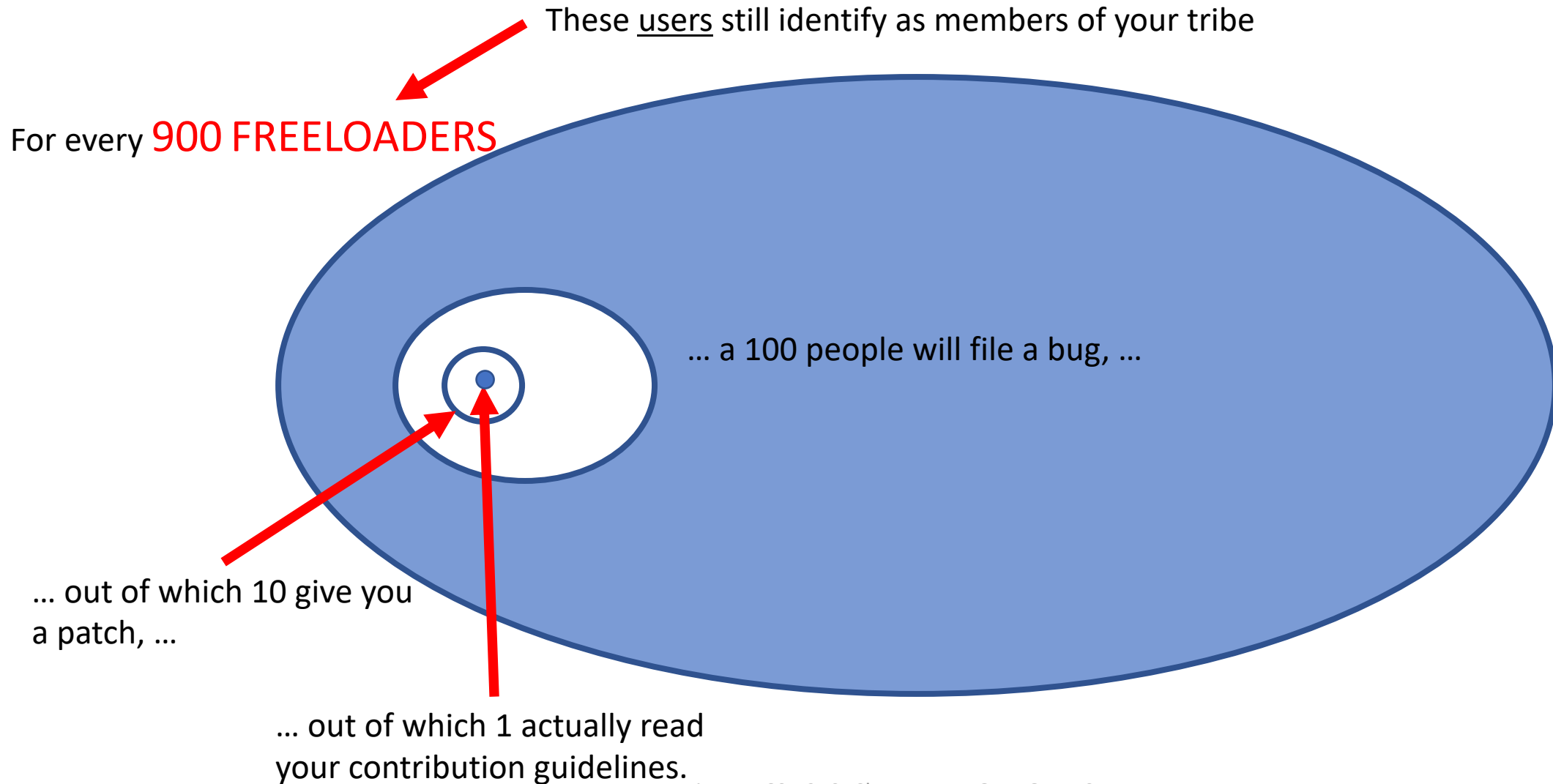
For every **900 FREELOADERS**

... a 100 people will file a bug, ...

... out of which 10 give you  
a patch, ...

... out of which 1 actually read  
your contribution guidelines.

# Rules of Thumb and Orders of Magnitude



Freeloaders means you're  
doing it right!

# Three On Ramps for Community Building

**How do you encourage people to use your project?**

(Because that's where you'll find bugs reports & developers)

**How do you encourage developers **selfishly** to experiment?**

(Because these are your future contributors)

**How do you encourage developers to share their work?**

(Because contribution flow is the growth and success of your project)

# Three On Ramps for Community Building

**How do you encourage people to use your project?**

(How do you make it easy to install/configure/use the software?)

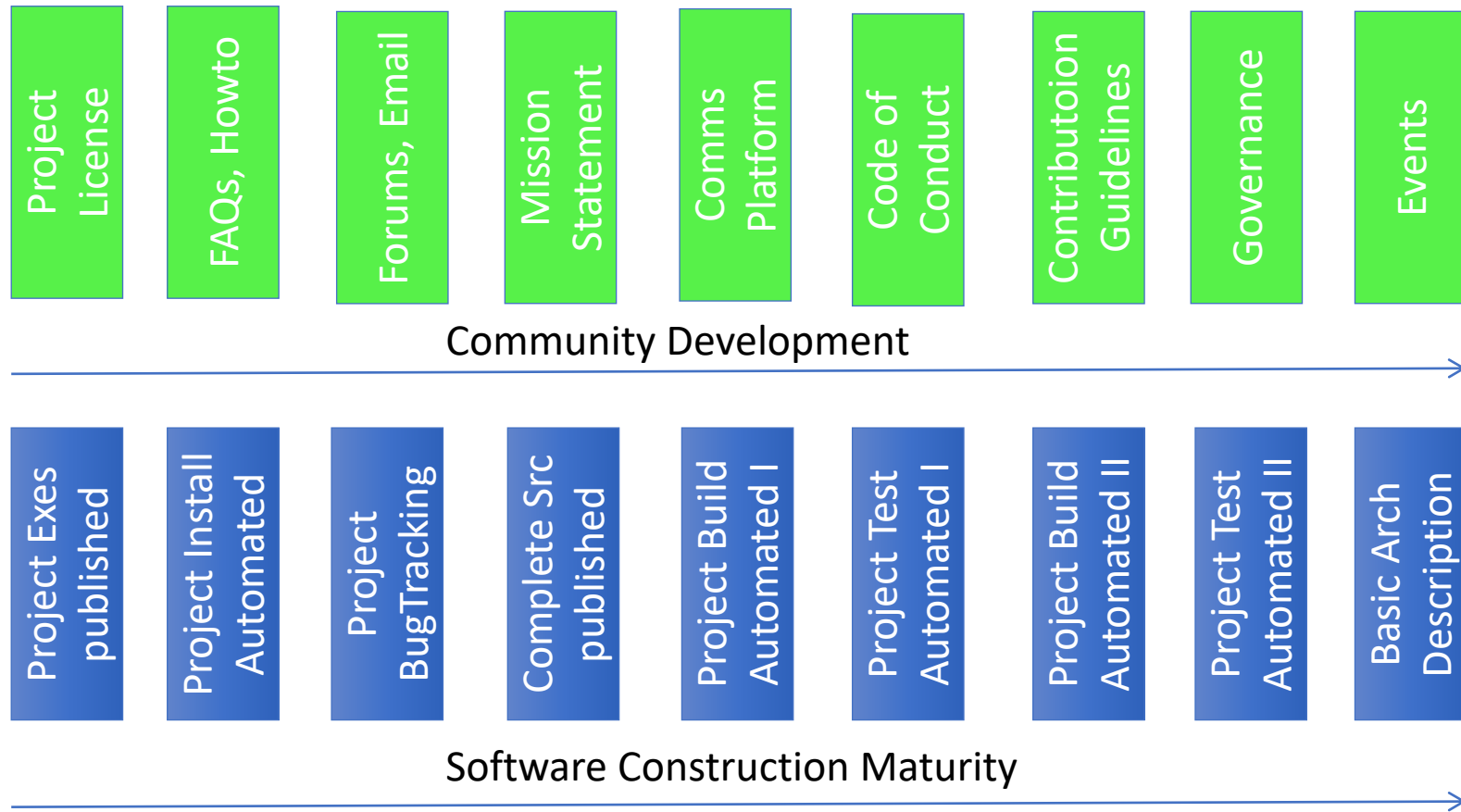
**How do you encourage developers **selfishly** to experiment?**

(How do you make it easy to build/test/experiment?)

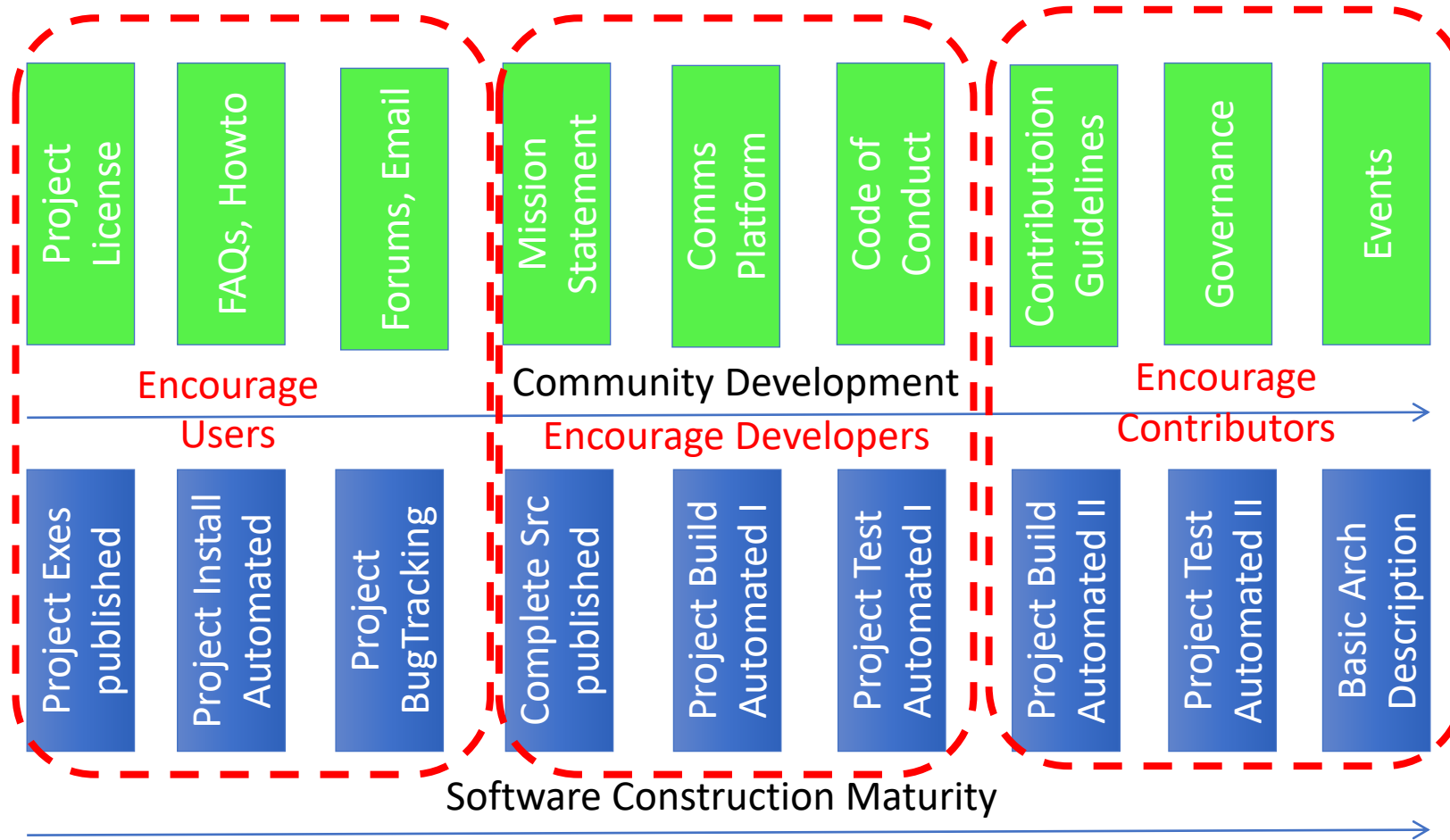
**How do you encourage developers to share their work?**

(How do you make it easy to contribute?)

# Open Source Community Practices



# Open Source Community Patterns







# What Does Your 10-Minute Rule Look Like?

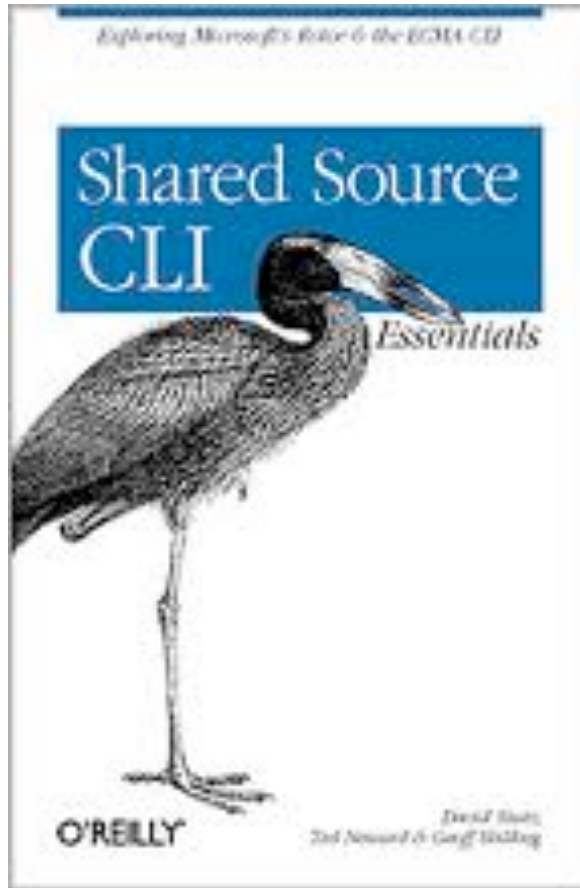


# What's your 'hello world' scenario?



# What's your mean time to dopamine

# ROTOR



500K LoC

500K Lines-of-Test Harness

Ran on Windows, Mac OS X, FreeBSD

One script to set environment

One command to build everything

One command to test it all

Minimal documentation

24 hours later ...

24 hours later again ...



# Ask Not What Your Community Can Do For You