Social and Political Infrastructure

How does FOSS really work?

Questions you Ask

- How does it work?
- Who keeps it running?
- Who makes decisions?

- Meritocracy
- Cooperation
- Running Code

These are just the general words of HOW it works, but there is more to it.

What is Successful?

- Operational Health
  - Ongoing ability to incorporate new code control and new developers.
  - Responsiveness to Bugs
- Survivability - Can the project exist independently of any one person or sponsor. Can it exist on its own?
- Technical Quality
  - Robust Developers
  - Strong Social Foundation with Maintainers
    - W/O this it may not be able to handle the growth of its success or departure of charismatic individuals.

How Do You Achieve This?

- Formal Governance Structure
    - Debates are resolved
    - New Devs are invited in (or out)
    - New Features Planned
    - The list goes on.
- Less Formal Structure
    - Need more self restraint to produce an atmosphere of fairness that people can rely on as a de facto form of governance.

Either one, they both lead to habits and procedures that are understood and everyone can participate. **Self Organizing Systems are best b/c everyone is conscious that a few bad apples can spoil the whole barrel.**

Beware of Forkability

- Forking is when someone makes a copy of your project
- Forking isn't necessarily a bad thing but if you want **your** project to be **successful** you would like people to continue on the path of assisting and making your **project great**, you are willing to **COMPROMISE!**
- The potential of Forking is always there…
- If a leader that everyone defers to starts making bad decisions -> restlessness follows by a revolt and then a fork will happen

Compromise and BD

- To avoid this from happening, you will compromise.
- Because people should be willing to compromise there aren't dictators for a project but rather **Benevolent Dictators.**
- **How do you Define a BD?**
  - Imaging you are a king whose subjects could copy the entire kingdom at any time and move to the copy to rule as they see fit. Would such a king govern differently from one whose subjects were bound to stay under his/her run no matter what s/he did?

Deferred Leader

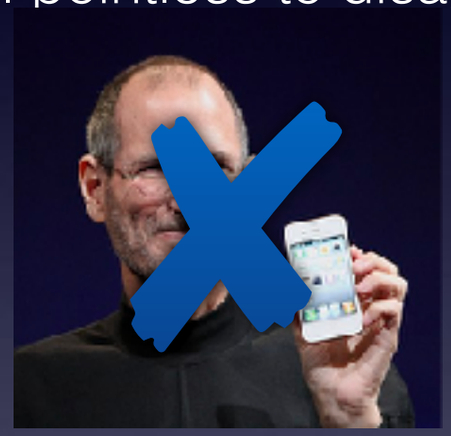
- They may not be formal democracies but in practice, they act as if they are when it comes to important decisions.
  - The “dictator” has no magical hold over a project.
  - Sometime a person who everyone defers to is b/c they choose to do so. Think Linus or Stallman.
  - Compromise first!
  - Usually is one of the founding members (correlation than cause)

BD is a Deferred Leader

- **Benevolent Dictator Model** - Final decision-making authority rests with one person, who, by virtue of personality and experience is expected to use it wisely.
- They can also be called “judge” or “community approved arbitrator”
- They actually do not make all or most decisions, they are more like a last resort when people can’t agree. They guide!
- They make directional decision more so than coding ones.
- They let things work themselves out through discussion and experimentation whenever possible.

Who can be a BD?

- Well honed sensitivity to one's own influence (self-restraint)
- Don't express opinions and conclusions that others feel pointless to disagree.
- Be open to others ideas, even if stupid.
  - BDs may do this as well, but acknowledge it!
  - BDs can afford to slip not devs with less seniority.



- Be sensitive as their words can carry weight in both technical and psychologically
- BD doesn't have to be sharpest tech skills, but skilled enough to code themselves, understand and comment on changes that are presented.
- Experience in overall design sense (recognize and endorse good design from whoever)

Forking goes both ways

- BDs can for a project if they wan to take it on a different route than what the majority of developers want to go.

- If for some reason that happens and you have more centralized run projects, it may be obvious who will be next BD.

- If decentralized, might have to have a Consensus-based Democracy

Consensus-Based Democracy

- Eventually become democratic systems.
    - Evolutionarily Stable
- If BD steps down or wants to spread work out , a non dictorial systems get created
- **Common Elements:**

    1. Group works by consensus most of the time

    2. There is a formal voting mechanism to fall back on when consensus cannot be reached

- **Consensus - agreement that everyone is willing to live with.**
    - Basically you discuss how to fix something for example, have a general agreement. However one should reiterate what was the conclusion of the consensus.

If decisions need to be undone..

- Thank your lucky stars for Version Control!
    - Decision can me unmade
        - If you think everyone was happy with it, but with objections later, due to missing discussion, you can revert.
            - Helps with bad or hasty judgement
- VC makes consensus need not to be formal
- Minor changes go in w/o discussion or min nods of agreement.
- Big changes should wait and have a discussion, check email and chat groups for such changes for talking.
- This apple to not only code, but website updates, documentations and anything else like that.

Can’t Agree then lastly Vote

1. Need Clear choices on the ballot.
2. Need a **honest broker - someone who post periodic summaries of the various arguments and keeping trac of where the core points of disagreement (& agreement) lie.**

- Honest brokers do job well, credibly call for a vote when the time comes and the group will have a ballot sheet based on the summary.
- Brokers can participation the debate.
- They shouldn’t let partisan sentiments prevent the from summarizing accurately

Voting Cont...

3. Adjust Ballot if Necessary - When ballot presented, someone might object due to misinformation or needing more, make sure to address appropiately.

4, Also What Voting system should you use?

- https://en.wikipedia.org/wiki/Voting_system
- There are many ways but for FOSS usually **APPROVAL VOTING.**
- **Approval Voting - each voter can vote for as many choices as possible.**

5. Conduct vote in public as much as possible.

- Each voter post on mailing list or online. So it can be kept on record
- FOOS voting system - https://vote.heliosvoting.org/

When to Vote

- Should be a last resort
- Not a great way to resolve debates b/c it ends discussion and creativity to solve problem; try to compromise as much as possible.
- Better to broker compromise than to vote.
- Vote has many people or a group unhappy, compromise everyone is a little bit unhappy.
-

Prevent PreMature Voting

1. Most obvious: “I don’t think we’re ready for a vote yet.”

2. You can ask for an informal (non binding) show of hands

    - It may make others more willing to compromise if the response is more towards one side over another.

3. Most effective way is simply offer a new solution or a new viewpoint on an old suggestion to show engagement with the issue.

Who can Vote?

Kinda low key points out who is more involved or has better judgement than others.

- You can look at who commits the most and attach voting privileges to it.
- There is full and partial commit access
- Question is: do partial committer can they vote? It should be spelled out that PC is about committing not voting, if you choose to not have them vote.
- If a person show disruptive or obstructionist tendencies on mailing list, should be very cautious making him a commiter, even if technically skilled.’

Who can Vote cont..

- Coders are not the only ones that should vote, maintainers should have a right as well.
- There are people who are **maintainers (sometimes called member) - Responsible of handling other parts of project besides code, like the website, documentation, donations, legal help, organize events, manage bug tracker, etc etc.**
    - Maintainers may have a level of influence in the community or familiar with dynamics that exceeds the code committers.

Speaking of Maintainers

- Adding new Maintainers
    - When choosing new people, this is **secrecy is appropriate.** You need to take that persons feeling in consideration.
    - Should be done in a private mail ing list for example and propose in to group. In order to have others speak their mind freely knowing its private.
        - Make sure everyone making the decisions has had an opportunity to respond. Discussions will happen and may result in voting.
        - Be open and frank, the mere fact this discussion is happening should be secret.
            - If someone asked to be considered, make sure to be as polite as possible with a clear explanation of why they were not picked. Also say that there will be other opportunities in the future if there is any.
- Make sure to have a voting system I'm. place, like majority vote or you may have to have at least N votes, or whatever you prefer. But make sure the process is in place. This may be also used to remove a maintainer, hopefully you will never have to do that.

Polls vs Votes

- Sometimes you poll because:
    - It is useful to expand the electorate. For example, can't figure out whether a given interface choice matches the way people actually use the software.

1. Make sure to is clear to participants that theres' a write in options

2. Make sure it is clear what are the options

Vetoes

- **Veto-is a way for a developer to put a halt to a hasty or ill considered change @ least long enough for everyone to discuss it more.**
  - Strong object and a filibuster

- Some projects make it difficult to override others let it be overridden by majority vote.

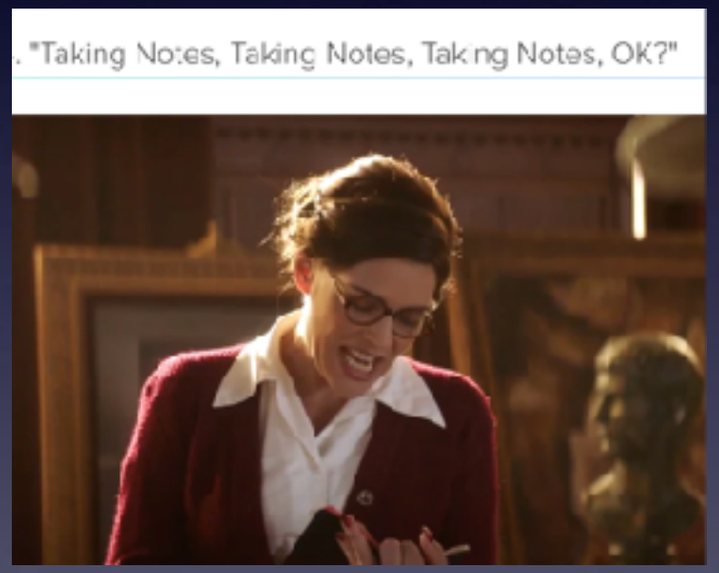
- Veto should come with a thorough explanation; a veto without such an explanation should be considered invalid on arrival.

Vetoes cont..

- Veto abuse can occur and be problematic.
    - Don’t be hasty to veto, continue to have discussion.
- You can prevent abuse by being reluctant to veto yourself.
- Call out gently when someone else uses their veto too often.
- Remind vetoes are binding for only as long as the group agrees they are.
- To express a veto people may write “-1”. Comes from Apache software foundation —> https://www.apache.org/foundation/voting.html
    - -1 can also be a very strong objects
- Vetoes apply retroactively… don’t object a veto on ground that the question has already been committed or action taken. Unless it was weeks or months later.

Writing it Down “Taking Notes”

- You need to record things somewhere, in order to give decisions legitimacy and make it clear that is it based on discussions and agreements.
- Link to relevant threads, mailing list archives, and make sure to ask again if you can’t find a point that was discussion.
- Its the description of the agreements that people made, no surprises should come of it.
- Don’t be comprehensive, no doc cant capture everything.
    - Don’t mention obvious things
    - Don’t need to re-write guidelines
    - Good way to determine what to include on docs think of newcomers ask most often.’
    - Don’t make it a FAQ sheet, just a narrative structure of what is asked.
- Rules and documentation should have conventions and living reflections of the projects/groups intentions.
    - It should not be a source of frustration or blockage.
    - If many people disagree with rules, then something needs to change and reevaluated.



Good Examples of Guidelines

- https://wiki.documentfoundation.org/Development
- https://subversion.apache.org/docs/community-guide/
- https://www.apache.org/foundation/how-it-works.html
- https://www.apache.org/foundation/voting.html.