User Respecting Software

Software that respects users.

How is user respecting software different from freedom respecting software?

It isn’t.

User respecting software is freedom respecting software that is about user needs (and not just developers, or the need of someone to make money.)

It is about bottom up rather than traditional top down development.

Cathedral vs Bazaar Model vs User Respecting Software Model

- **Cathedral, monolithic model of software development inflicted on users.**
- **Bazaar, model. A bunch of developers working together to make some software, without much input from users.**
- **User Respecting Software model, software developed with user requirements and desires, not because of what makes money for someone or what “developers feel like doing today.” (this of course does not in anyway preclude users paying developers to write it and support it, but instead that developers have no need to add “anti-features” such as proprietary versions or unwanted advertising.)**

Why is Freedom Respecting Software Playing Catchup?

- **Some Free Software projects created before their proprietary equivalents have gotten “behind” in terms of number of users.**
- **We can’t keep using the “entrenched monopoly” excuse**
- **If freedom respecting software is inherently better, because it has the four freedoms, why don’t all users see that?**

Excuses for Proprietary Software

- **Meets users current requirements.**
- **Awareness of the software**
- **Peer Pressure**
- **Required to use it by schools, work, or government.**
- **Software solves a users problem. (That they didn’t already know they had.)**

What do users want out of software?

- We should ask them.
- Works with existing software.
- Ease of Use
- Features.
- Performance
- Fills an existing need

Awareness of the Software

- **Also known as “marketing” (a term I know people hate).**
- **How do users find out about software?**
- **From other users, internet search, existing repositories.**
- **Who tells them about the software and how do they know what search terms to use?**
- **Do users even realize if it is a problem that software could help solve?**

The Network Effect

- More or less important depending on the type of software (software that is about connecting people has more “network effect” than solo programs.) (for example video chat)
- Everyone all using the same formats (I’m still messing with old pdfs made in a certain ancient proprietary office suite that don’t quite come out in Libre Office)
- Telling people about the software so people in your network use it too.
- Can be a negative thing with proprietary software, particularly for certain multi-user programs that are hard to avoid.

The Usage of existing power structures to spread software use

- **School usage of software has been covered in several other talks at this conference and several ways to deal with that.**
- **Workplace usage of software, can often be resolved by talking to others, finding another job, freelancing, and starting a business.**
- **Government, the difficulty of finding who the decision makers are, lobbying and getting laws passed. (covered in several other talks as well.)**

Solving a Users Problem, Software you didn’t know you needed.

- You don’t need software, so why even have it, the world ran just fine before computers.
- Proprietary companies are always coming up with this stuff and getting a first to market advantage.
- Try to stay ahead of the curve. It is better to guess what users will want in the future and create it rather than trying to replace proprietary software later.
- Software can’t solve everything.
- This isn’t just about making the software, but being up to date with user trends and demanded features (see my next slide about IRC vs Discord)

IRC vs Discord

- **IRC came first and is better because it is freedom respecting software.**
- **People started using discord because it saves the chat logs for when you are offline (do not have to be logged in 24/7 to read the chat logs)**
- **Discord got used for open source projects(and became known as the “discord problem.”**
- **Not to be confused with Discourse forum software(Discourse is freedom respecting software.)**
- **Users and system admins don’t fix this problem because “why bother?”**
- **This reminds me of the old “Bitkeeper problem” (Bitkeeper, an old piece of proprietary software no one misses now because we all use git.)**

Minetest vs Minecraft

- **Minetest came first.**
- **Not really sure why Minecraft surpassed Minetest in terms of users and mods.**
- **Possibly has to do with “marketing” rather than any inherent “betterness”**
- **The Minecraft engine was written in Java for a true cross-platform experience, while Minetest was written in C++.**
- **C++ is considered a hard language that few people like to hack on.**
- **Minetest had/has trouble with getting mods checked out so they fit into the game and are “relativity bug free” (yeah right)**
- **Last time I played it, (a few months ago) the minetest game had trouble connecting over a local area network. (mineclone worked ok though)**

Proprietary Social Media Vs Mastadon

- **Proprietary Social Media came first, but old social networks have long since been replaced (who even remembers myspace or that google thing, I forgot what it even was.)**
- **New kinds of social media all the time, some kinds only work on cell phones (as if proprietary sass platforms weren't bad enough, now you have to use a cell phone to use it.)**
- **Federated social media, sounds like a better way.**
- **In practice, a lack of features users care about such as the “groups” feature.**
- **Recent political unrest in the U.S. and some prominent “canceling” of formerly acceptable speech on certain social media platforms had people scrambling for alternatives.**
- **Free software was unable to take advantage of a lot of this because we refused to add the necessary “groups feature” that users requested and developers did not want to implement.**
- **Gab added the feature, but not in a way that users or developers were entirely happy with for mastadon (groups were limited to the particular instance, and not universal, not a problem if you don’t need to federate.)**
- **Should you reintegrate code from hard forks?**
- **A fundamental need for semi-private spaces, things that are not “public” but also not “completely unsearchable and private.”**

How can we apply User Respecting Software Model to freedom respecting software projects?

- **Ask for user input before even starting.**
- **Users should directly fund development (not have to go through anti-feature business models, business models are not more important than users.)**
- **Developers should anticipate user needs in advance. (so we don’t have to play the catch up game)**
- **“Marketing” and “Collecting user requirements” is just as important as coders and other types of contributors**
- **Don’t be afraid of the “hard fork” if project leaders are not being responsive enough to user needs.**