Task category / Task	Value	% of total
Remaining	0	0%

Instructions

The goal of this stage of the funding allocation procedure is to estimate "value" of individual tasks.
Value shall measure how useful each task would be to our users, to developers, to health of the project, etc.
Every participant gets 1000 units of value to allocate between the tasks, starting from an equal distribution of units.
So the only thing you should edit is to allocate those 1000 units in the "Value" column of each task. Everything else will be computed automatically.
LibreOffice / OpenOffice can highlight the cells you should fill in: Go to "View" and select "Value highlighting" (CTRL+F8).

The tasks are grouped into different areas such as "Speed" and "User experience" to ensure we think about all areas.
Some tasks might be helpful to not only one but multiple areas. Nevertheless, they are only enlisted in one area.
Thus you do not need to feel obliged to cast votes in every area, you might as well instead vote for the tasks which cover more than just the area they are enlisted at. When deciding about the value of a task, you might want to consider how many areas you think it covers.

A few important things here to ensure we get the true benefit of "wisdom of the crowd":
- Don't be strategic (e.g. don't over-allocate to one area because you assume others might under-allocate to it).
- Don't collude.
- Don't consider development cost in your value allocation, this will be done in the next stage.
- If you aren't sure, leave stuff at an even allocation, don't assign 0 value.

Once everyone has allocated, we will take the average for each task to be our group estimate for that task.

Speed - Improve performance of Freenet or Freenet applications

Web of Trust: Finish first iteration of most critical speed fixes (1 bugtracker entry). Was subject of previous 2 years of paid work. Ensures this work is not left unfinished. Needed for Sone / Freetalk / filesharing / … (see all other sections except Technical Debt)	15	1%
Freetalk: Make Freetalk usable again by using the new WoT event-notifications API to fix its most severe performance issues. The API had been implemented as previous paid work to specifically allow repairing Freetalk. Thus we'd reap what we sowed there.	15	1%
WoT: Second iteration of less critical performance fixes: Reduce transaction size from O(512) to O(1): Split up "import trustlist" transaction into 1 transaction for each trust.	15	1%
WoT: Second iteration of less critical performance fixes: Identity garbage collection: Stats show that less than 1000 of the 14000 WoT identities are still active (thanks to ArneBab). Thus, we need to GC abandoned identities.	15	1%
WoT: Second iteration of less critical performance fixes: Process results of peer review of xor's Web of Trust bachelor's thesis (which was about last year's performance improvements).	15	1%
Fred: does it provide sufficient benefits to re-add native acceleration for FEC, or crypto? Especially JVMs without gmp acceleration - Oracle on Windows.	15	1%

Task category / Task Remaining	Value 0	% of total 0%
User Experience - make things easier and more enjoyable to use		
Single use node references with authentication token: Currently, two users need to add the node reference of each other to create a connection. With this, only one of the two would have to add the other. Might benefit from FOAF (see 3 tasks below) for connectivity if Opennet is off.	15	1%
Darknet invitation bundles: Feature for adding a single use node reference to an installer executable. People could hand out the installer executable to their friends to allow them to connect by Darknet instantly.	15	1%
Short node references: Currently, references fill almost half a sheet of paper. Improve this by uploading node references to Freenet itself as a random KSK, with for example 128 bit entropy to be ~ 25 letters. (Requires having peers already, but without the UI could just show the long one)	15	1%
Friend-of-a-friend connection suggestions ("FOAF"): Like the Facebook friend finder, Freenet could be improved to tell you about Darknet peers of your peers. You could then chose to add them as your peers. Part of the code exists.	15	1%
Friend requests, like in Facebook: With primitive FOAF, both peers would still have to add each other. With friend requests, peers of your peers could just request to connect to you. Could use similar UI to Facebook friend requests.	15	1%
Something like Icicle to ease setting up Darknet connections - perhaps add support for connecting nodes over text message or Signal?	15	1%
Darknet chat improvements: Freenet allows sending messages to Darknet peers. The UI for that is very primitive. It could be improved, for example to be similar to Facebook chat. Needs fixes for high message loss probability.	15	1%
Maybe FLIRCP can add darknet N2NTMs as a backend?	15	1%
FlogHelper (WoT-based blogging app): Accumulate flogs of other users. List them nicely. Add a search function. Use that to replace or amend the current concept of FProxy's default bookmarks as entry point for Freenet. Hundreds of blogs instead of ~10 bookmarks.	15	1%
Better feedback on progress: currently in FProxy progress is displayed in terms of completed blocks. Information on how many fetch / insert requests are running for each one, and the progress on them, could help make the default mode of operation seem like hanging.	15	1%
Make robust search functionality part of the default installation [task is possibly not specific enough: Complexity depends a lot on search target: Freesites? FlogHelper flogs? Freetalk messages? WoT identities? Sones? Filesharing?]	15	1%
Add ability to comment on Freesites (fulfilled by babcom (is opt-in) / Freetalk / Sone?)	15	1%
Frost-alike communication system supporting easy usability, encrypted and read-only boards and hidden instant messaging between 2 or more ID's, integrated file transfer agent, mixed bulletproof ID'd and unsigned messaging. [= e.g. Freetalk. Likely requires the WoT speed tasks.]	15	1%
Improve the vulnerability and functionality of Flip [One anonymous user commented that you may as well re-invent FLIP]	15	1%
Improve FProxy CSS3 support to allow better Freesite UI	15	1%
Document the content filter	15	1%
Improve Shoeshop/blob handling [how?]	15	1%
p2p file exchange by time-set expiring keys [Imprecise task? Also violates main goal of Freenet of preventing censorship!?]	15	1%

Task category / Task Remaining	Value 0	% of total 0%
Freenet key links directed to a dedicated Freenet browser, opennet [= non-Freenet Internet?] links to the system default like in the AutoHotKey script "Freenet Browser Director”	15	1%
Warn users not to use Chrome, but instead an open source/free software browser	15	1%
FLIP or FLIRCP could use work to become official plugins, maybe with a webclient instead of having to install a potentially privacy leaking 3rd party one. [Likely requires the WoT performance work in the “Speed” section.]	15	1%

Task category / Task	Value	% of total
Remaining	0	0%

Security - Make Freenet more secure against attacks

Fix plugin data leak: None of our official plugins deletes its database when the user uninstalls it. They also do not obey the "PANIC" button of Freenet which should delete all client data.	15	1%
Fix lack of plugin data encryption: WoT, Freetalk, and potentially other plugins do not encrypt their databases when the user has configured Freenet to encrypt user data. Should be changed to obtain an encryption key from Freenet to encrypt their stuff.	15	1%
Add simple tunnels. The easiest approach here may be to have some automatically added temporary time limited or data throughput limited FOAF connections that form simple tunnels. Tunnels should have a probability to use for high HTL inserts and requests based on security settings for the node.	15	1%

Task category / Task	Value	% of total
Remaining	0	0%

Technical debt - Stuff that will make future development faster

fred: Documentation is said to be poor. Improve it. Would help people other than Matthew to understand the code. Good to do this now while Matthew is still available as a volunteer and we can thus maybe ask him questions.	15	1%
fred unit test coverage is 20%. Improve it. Would help people other than Matthew to understand the code. Good to do this now while Matthew is still available as a volunteer and we can thus maybe ask him questions.	15	1%
WoT unit test coverage is 45%. Given the lower total line count, it would be possible to complete it to 100%. That would be a preparation of the major “refactoring” of replacing db4o with a different database. May become necessary: db4o’s vendor has abandoned it.	15	1%
Alternate task instead of the above tests/doc: Permanently use 1 day per workweek (= 20% of total time) to deal with any kind of technical debt whatsoever. Allow the developer to chose freely which issues to deal with so even unimportant technical debt will be dealt with eventually.	15	1%
Alternate task instead of the above technical debt day: 1 day/week as “do whatever you want to”-day like Google does. Example activities: Generic education (new Java language features, programming patterns, algorithms, papers); Wiki; Drafting new apps; Tests; Doc; Mailing lists	15	1%
Fred: Migrate to the Gradle build tool - Gradle can provide dependency fetching and verification support and then defer to the existing ant build. (Or we can move the project to fit standard directory structure and then the Gradle build becomes a lot easier to write.)	15	1%

Task category / Task Remaining	Value 0	% of total 0%
Outreach - Stuff that will help attract users, developers, and donors to Freenet		
Language: Lower threshold of entry by working on how to install Freenet	15	1%
Language: Add these strings for install to translation resource. Bonus for looking at this.	15	1%
Language: Simplify website language. Boil this into this for all pages.	15	1%
Language: Fix current low-hanging problems with UI language, as per issues on transifex.	15	1%
Language: Add Transifex issues to main bugtracker	15	1%
Language: Establish overview of, and do away with needlessly difficult words	15	1%
Language: Document how Freenet works, with a infographic, on the Wiki, detailing the remainder of speciality words.	15	1%
Website redesign	15	1%
Website redesign: Migration to static Github Pages	15	1%
Website redesign: New CSS redesign	15	1%
Website redesign: Migration of more detailed documentation to Github Wiki, allowing website to focus on onboarding new users [Notice: we have 3 non-Github Wikis currently, 2 of those have already been waiting for migration into the main one for years.]	15	1%
'How to' YouTube video series: A couple of (maximal)-5-minute videos, each explaining one important practical aspect of using Freenet. Do this because currently we demand users to read far too much (FAQ, wiki, etc.). Example of ZeroNet.	15	1%
Update FSNG: Show web interface usage rather than Thunderbird to match modern user expectations	15	1%
Update FSNG: Use a modern Windows installation with the default theme for screenshots	15	1%
Legal docs: Have resources akin to the Tor legal FAQ for exit node operators, created by a legal expert on the official project page, that can be shown to LEA or users. Explain what it is used for, how to use it safely, best practices, plausible deniability, legal status, etc.	15	1%
Legal docs: Sample response letters for the potential arrests or raids, especially for the recent events as some users were completely innocent. We need to educate LEA, more than ever, that it in itself is not a crime nor suspicious to run Freenet.	15	1%
Legal docs: Instead of the above suggestions, just link the Tor legal FAQ. Tor is pretty similar and thus we shouldn't need to create our own one. Notice especially that there are 200 countries in the world and 200 lawyers would be pretty expensive.	15	1%
Fundraising: While volunteers did help with fundraising, they don't seem to enjoy it very much – we've contacted much fewer entities than those listed in the Wiki. So also have Freenet's employee do it. Important as Matthew will require much higher level of funding to come back.	15	1%

Task category / Task Remaining	Value 0	% of total 0%
New core features - Enable using Freenet for new kinds of tasks/usecases		
Filesharing (#1 request on uservoice with 700 votes, not yet implemented but could easily be built upon Freetalk)	15	1%
Keepalive: This is very popular on uservoice. It could be improved significantly with some integration into the web interface and minor backend changes.	15	1%
Social networking (Sone, not enabled by default due to WoT performance issues and issues of its own)	15	1%
Blogging (FlogHelper, not enabled by default due to WoT performance issues)	15	1%
Forums (Freetalk, not enabled by default due to WoT performance issues and issues of its own: Freetalk itself needs its performance fixes in the “Speed” section. Also it needs general usability / security / bugfixing / polishing work which this entry shall be about.)	15	1%
Mail (Freemail, not enabled by default due to WoT performance issue)	15	1%
Distributed version control (various tools for Git / Mercurial over Freenet, not finished yet and lack code review)	15	1%
Transport plugins are a very popular feature request. There is code but it needs a lot of work, and it may be better to start from scratch. In any case it will involve breaking up a lot of monster classes (good for code quality), and a lot of knock-on effects on concurrency (less good!).	15	1%
Make dynamic web pages possible [likely bogus suggestion: We must not allow full JavaScript for the security issue of preventing tracking / de-anonymization. We usually use plugins instead such as Sone, FlogHelper, Freetalk, Freemail, etc. So you might vote for those above instead]	15	1%
Add FCP calls to Freemail to setup a user and to change user passwords	15	1%
Do not reject Freemail messages when an ID is not known, queue them for a period and attempt to find the identity	15	1%
New keytype: PSK (programmable subkeys - shared ownership)	15	1%
New keytype: RSK (revocable)	15	1%