How to tell if a FLOSS project is doomed to FAIL

CS 401

|  |  |  |
| --- | --- | --- |
| **Aspect/Range** | **Rating** | **Comments** |
| **Size** |  |  |
| Source code > 100 MB. [ +5 points ] | 5 | 121.7 MB total when uncompressed |
| source code also exceeds 100 MB when it is compressed [ +5 points of FAIL ] | 5 | 102 MB total when compressed |
| **Source Control** |  |  |
| no publicly available source control (e.g. cvs, svn, bzr, git) [ +10 points ] | 0 | Use bazaar, some parts use git |
| **Publicly available source control, but** |  |  |
| no web viewer for it [ +5 points} | 5 | I couldn’t find a web viewer |
| no documentation on how to use it for new users [ +5 points] | 0 | Documentation is okay as long as you know your way around the terminal |
| You've written your own source control for this code [ +30 points] | 0 | Git works for our purposes (Euca2ools) |
| You don't actually use the existing source control [ +50 points] | 0 | ---- |
| **Building From Source** |  |  |
| no documentation on how to build from source [+20 points] | 0 | Install guide – from what I can tell seems pretty in-depth |
| Documentation exists but it doesn't work [ +10 points] | 0 | ---- |
| **The source is configured:** |  |  |
| handwritten shell script [ +10 points] | 10 | Need to configure from terminal – not necessarily a shell script though |
| editing flat text config files [ +20 points] | 0 | ---- |
| editing code header files [ +30 points] | 0 | ---- |
| source isn't configurable [ +50 points] | 0 | Seems to be pretty configurable from what I’ve heard from Dave’s efforts |
| **The source builds:** |  |  |
| Not using GNU Make [ +10 points] | 0 |  |
| third-party tools [ +50 points] | 0 | ---- |
| your own build tool [ +100 points] | 0 | Not recommended |
| The build has no tests [ +5 points] | 0 | Seems to have tests since it was failing while Dave was having trouble last week. |
| **Java** |  |  |
| build process doesn't use standard build tool (Ant, Maven, etc) [ +10 points] | 0 | Java is not a big factor in this project |
| runtime artifacts rely on native libraries [ +5 points] | 0 | ---- |
| build has no tests [ +20 points] |  | ---- |
| **Bundling** |  |  |
| source only comes with other code projects that it depends on [ +20 points] | 20 | Dependencies are available on the website, but are not bundled together with the source |
| cannot be built without first building the bundled code bits [ +10 points] | 0 | ---- |
| you have modified those other bundled code bits [ +40 points] | 0 | ---- |
| **Libraries** |  |  |
| only builds static libraries [ +20 points] | 0 | ---- |
| can build shared libraries, but only un-versioned ones [ +20 points] | 0 | ---- |
| does not try to use system libraries if present [ +20 points] | 0 | ---- |
| **Java** |  |  |
| code builds native deliverables [ +5 ] |  | ---- |
| **System Install** |  |  |
| tries to install into /opt or /usr/local [ +10 points] | 10 | Recommended to install in /opt |
| has no "make install" [ +20 points] | 0 | ---- |
| doesn't work outside of the source directory [ +30 points] | 0 | ---- |
| **Code Oddities** |  |  |
| uses Windows line breaks [ +5 points] | 0 | Not an issue |
| depends on specific compiler features [20 points] | 0 | Uses GNU complier |
| depends on specific compiler bugs [20 | 0 | ---- |
| depends on Microsoft Visual ?? [ +100] | 0 | Nope |
| **Communication** |  |  |
| does not announce releases on a mailing list [ +5 points] | 0 | Mailing lists seem to be very active from what I have experienced. |
| does not have a mailing list [ +10 points | 0 | (see above) |
| does not have a bug tracker [ +20 ] | 0 | There is a web-based tracking system – but seems overly complicated |
| does not have a website [ +50 points] | 0 | ---- |
| sourceforge vaporware [ +100 points] | 0 | ---- |
| **Releases** |  |  |
| does not do sanely versioned releases (Major, Minor) [ +10 points] | 0 | Seems like they do a good job at this (not to much, no to little). |
| does not do versioned releases [ +20 | 0 | ---- |
| does not do releases [ +50 points] | 0 | ---- |
| only does releases as attachments in web forum posts [ +100 points] | 0 | ---- |
| releases are only in .zip format [ +5 ] | 0 | tar file |
| releases only in OSX.zip format [ +10 ] | 0 | ---- |
| releases are only in .rar format [ +20 ] | 0 | ---- |
| releases are only in .arj format [ +50 ] | 0 | ---- |
| releases in invented format [ +100 ] | 0 | ---- |
| does not unpack into a versioned top-level directory (e.g. glibc-2.4.2/ ) [ +10 | 0 | ---- |
| does not unpack into a top-level directory (e.g. glibc/ ) [ +25] | 0 | ---- |
| unpacks into an absurd # of directories [ +50 points] | 0 | ---- |
|  |  |  |
|  |  |  |
| **History** |  |  |
| fork of another project [ +10] | 0 | Started at UC Santa Barbara |
| primary developers were not involved with the parent project [ +50 points] | 0 | ---- |
| Until open sourcing, your code was proprietary for: |  | Started as Open Source I believe |
| 1-2 years [ +10 points of FAIL ] | 0 | ---- |
| 3-5 years [ +20 points of FAIL ] | 0 | ---- |
| 6-10 years [ +30 points of FAIL ] | 0 | ---- |
| 10+ years [ +50 points of FAIL ] | 0 | ---- |
| **Licensing** |  |  |
| does not have per-file licensing [ +10] | 0 | GNU General Public **License** |
| contains license incompatibilities [ +20] | 0 | ---- |
| No notice of licensing intent [ +30] | 0 | ---- |
| No copy of the license text [ +50] | 0 | ---- |
| No license [ +100] | 0 | (see above) |
| **Documentation** |  |  |
| Code doesn't have a changelog [+10] | 0 | Available on their wiki page |
| doesn't have documentation [+10] | 0 | Documentation is also available on their website |
| website has no examples of use [ +20] | 0 | ---- |
| website has no documentation [ +30] | 0 | (see above) |
|  |  |  |
| **FAIL METER TOTAL** | 55 | FAIL: Babies cry when your code is downloaded |

After analyzing the Eucalyptus FOSS project, I found that there is room for improvement. Based on Tom Callaway’s system, the project received a score of “FAIL: Babies cry when your code is downloaded”. I think this may be a little extreme, as I would say that the project seems more like a rating of “FAIL: You're probably doing okay, but you could be better.” I say this mainly because some of the points that I marked down were recommended, but NOT forced upon the users. For example “tries to install into /opt or /usr/” it is recommended to install in the /opt directory, but it allows you to choose another directory if you wish. The most points came from the dependencies being a separate download outside of the source. If they were to bundle them together it would drastically improve the score (since they already violate the over 100MB rule it shouldn’t be that big of an issue). Overall I would give them a letter grade of B - doing well but still room for improvement.