

Going Open Source

The 20 Most Important Things To Do

open **YOUR** mind



Martin Aschoff, AGNITAS AG

Before deciding on going OS

- ▶ **1. Analyze the open source competition in your space**
 - ▶ Mine SourceForge for competition
 - ▶ Are you the first in your segment?
 - ▶ Do you want to re-invent the wheel?
 - ▶ How do you differ from your competitors?
- ▶ **2. Collect info on successful open source projects**
 - ▶ SourceForge: Rankings, download numbers, bug tracker activity, ...
 - ▶ Enterprise Open Source Directory (www.eosdirectory.com)
 - ▶ Look at the project websites, visit their community sites
 - ▶ Can you match that or, at least come close?

- ▶ **3. Identify your business model(s)**
 - ▶ Open source is **NOT** a business model
 - ▶ Open source is "just" a development, distribution and marketing model
 - ▶ This step is the most important one to undertake if you want to succeed!
 - ▶ My experience: at least 50% of the ideas will not work
 - ▶ Do not rely on the wrong assumption that, if your user base is big enough, money will follow automatically
 - ▶ My experience: users become very creative to avoid paying you
 - ▶ You probably have to tweak your business model several times
 - ▶ My advice: Do not fall in love with your business model, but be flexible enough to adapt to reality

Before deciding on going OS

- ▶ **4. Calculate how going open source will lower your revenues from existing clients**
 - ▶ Clients will demand a better price since you give your software away for free
 - ▶ Clients will want to switch to your open source product
 - ▶ Core question: how quickly will your open source revenues make up the loss of revenues from existing clients?
- ▶ **5. Talk to your investors / shareholders about your idea of going open source**
 - ▶ How familiar is your board with the term "open source"?
 - ▶ How short term / long term oriented is their investment?
 - ▶ Are they ready and do they think big enough?

Before deciding on going OS

- ▶ **6. Develop a set of arguments why going open source is good for your existing clients to minimize loss of revenues**
 - ▶ The real argument: "The software you use will continue to exist even if our company fails or gets bought."
 - ▶ Admittedly an argument difficult to communicate
 - ▶ Other arguments: free code contributions from community, better interoperability with third party software, improved security, ...
- ▶ **7. Define the initial feature set and capabilities of the open source version** (especially if a closed source version will continue to exist)
 - ▶ Your open source version must be a working product and attractive enough for users to download, test, install and configure
 - ▶ But some attractive premium features should be reserved for your commercial product

Before deciding on going OS

- ▶ **8. Develop a business and a financial plan for your investors / shareholders to get their approval**
 - ▶ You can only convince them if you are convinced about your plan!
 - ▶ Plan carefully and rather for the worst case than the real case
 - ▶ My experience: Like software development, everything takes twice as much time as planned initially

(number of downloads, growth of community, ramp-up of revenues, etc.)

- ▶ **9. Choose the project name and logo carefully**
- ▶ If you have to invent a new name:
 - ▶ Choose a name that not only native English speakers are able to pronounce ("Massachusetts")
 - ▶ Choose a name that has no bad meaning in other languages ("Pajero", "Mist", etc.)
- ▶ Check, whether your name is available as product name, domain name and SourceForge project name
- ▶ Design a logo that looks good even if printed in b/w and small size

- ▶ **10. Define the granularity of developer, admin and user documentation**
 - ▶ Quality and quantity of documentation defines initial acceptance of your project
 - ▶ This can be a critical factor for the timing of the initial release of your project
- ▶ **11. Choose a suitable open source license**
 - ▶ Most users don't care about the license
 - ▶ But it is important for use in commercial environments
 - ▶ Select a OSI-approved license (= "real open source")
 - ▶ If possible, select a well-known license (Apache, BSD, CPAL, Eclipse, GPL, LGPL, MPL)

After deciding on going OS

- ▶ **12. If the software stack below your project would contain closed source components or components incompatible with your open source license, replace them**
- ▶ **13. Refactor your source code**
 - ▶ Refactoring = close security holes, update frameworks and libraries, restructure and comment source code, use english words only, add license header
 - ▶ Your code will be public and you don't want to be ashamed of it
 - ▶ If the community does not understand your code, nobody will contribute new code
 - ▶ If documentation is not the deciding factor for your timing, then refactoring is

- ▶ **14. Write the documentation for developers, admins and users**
 - ▶ Developers: build guide, list of files and their purpose, database schema, signature and description of web service methods, etc.
 - ▶ Admins: install guide, configuration files and parameters, etc.
 - ▶ Users: manual, online help, etc.
 - ▶ Tip: developers should not write the user documentation
- ▶ **15. Launch a project website (*.org) with community modules**
 - ▶ Recommended components of website: news, FAQ database, forums, bug tracker, public roadmap, wiki, text of license, contact
- ▶ **16. Don't push commercial offers on your project website**
 - ▶ For those services provide links to your company website (*.com)

After deciding on going OS

- ▶ **17. Write public announcements to communicate the launch of your open source project**
 - ▶ Targets: your project website, your company website, SourceForge, freshmeat.net, social networks, blogs
 - ▶ If you have the money, do PR: publish a press release, write a whitepaper, talk to the press, call editors, etc.
- ▶ **18. Publish the binary and the source code on SourceForge**
 - ▶ Or use an alternative platform with comparable publicity and reach ...
- ▶ **19. Set up a demo server on the Internet for public testing** (if your project is web-based)
- ▶ **20. Put all unfinished features and capabilities on the public roadmap for your next release**

After going open source

- ▶ **The point of no return has been passed - now the real work starts!**
 - ▶ Keep track of the SourceForge activity index and influence it to stay near the top of the lists
 - ▶ Check and moderate forum and wiki posts
 - ▶ Write FAQ entries for the most popular questions
 - ▶ Respond to e-mails asking for (free) support
 - ▶ Collect and analyze statistical data such as website visits, download numbers, forum entries, e-mail requests, etc.
 - ▶ But do not confuse downloads with users and users with customers
- ▶ **Book tip: "Producing open source software" by Karl Fogel, online available free at www.producingoss.com**

My experience:

Going open source is
not a short-lived adventure holiday,
but more of a permanent relocation
to a different country

Example: OpenEMM

- ▶ **In July 2006 we launched the first release candidate of OpenEMM, our e-mail marketing solution, as open source software**
- ▶ **Two years on, what we have achieved:**
 - ▶ > 60.000 downloads, > 1.300 active users, 10 to 20 new users/week
 - ▶ A growing and vibrant community that tests new features, provides feedback, improves the documentation, reports bugs, suggests bug fixes, translates the GUI, etc.
 - ▶ Code contributions by the community (Typo3 extension, SugarCRM plug-in, ...)
 - ▶ Growing international revenues (best source of income: SaaS, second best: customization & development)
- ▶ **Conclusion: It was worth the effort and we would do it again anytime!**

Want to know more?

My contact information:

Martin Aschoff, AGNITAS AG
maschoff@agnitas.com
www.agnitas.com
www.openemm.org

++49 89/55 29 08-68

Werner-Eckert-Str. 6
81829 Munich
Germany

open **YOUR** mind

