Public Administrations Partnering with Free Software/ Open Source Developers

IDABC Workshop on Public Administrations and OSS

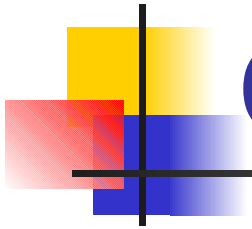
LinuxTag, Karlsruhe, Germany, June 23, 2005

Rishab Aiyer Ghosh (rishab@dxm.org)

MERIT, University of Maastricht

IDABC Open Source Observatory

http://www.europa.eu.int/idabc/oso



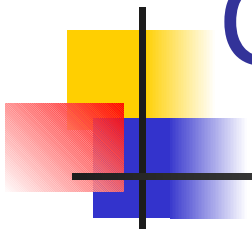
Outline

- What is the “Community”?
- External incentives
- 10 steps for interaction
- Overview of legal frameworks

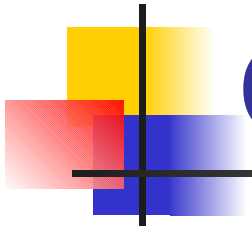
Community

- “There is no such thing as society”
- No single community with one coherent voice; overlapping heterogenous groups

Community

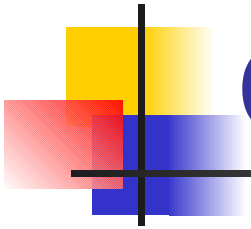


- By supporting free software, you (Pas) are part of the community
- You may not get extensive support from the mainstream – “the community”
- You may not need that support; the “land management software community” is likely to be other PAs, collaborators



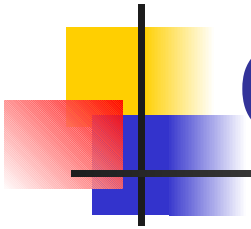
Community organisation

- Software projects differ in organisation structures among themselves
- “benevolent” or rotating dictatorships
- Semi-democracies, foundations
- Corporation-led
- ...all essentially self-organising



Community motives

- Monetary motives are not common...
- ...but earning money is. Most productive developers do not work “for free”, though many work for “freedom”
- No single, overriding motive
- Most interesting to PAs is the motive to improve software of personal interest

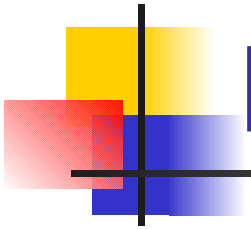


Community motives

- The most successful projects have a significant number of User-Developers
- A disadvantage for PAs building PA-specific software?
- An advantage for PAs building general-interest software?
- Opportunity for niche community interested in PA software

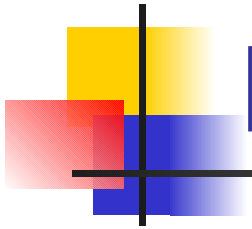
External incentives (money!)

- Development is often paid for!
- PAs with specific needs will probably have to pay for them



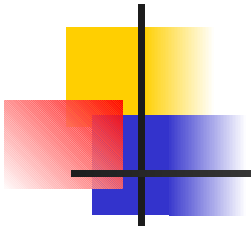
External incentives (money!)

- Hire an entire team?
- Hire people (or contribute staff) as team leaders?
- Contracting for customisation, services and support (esp. individuals, SMEs)?
- Bounties (e.g. Shuttleworth)



Examples

- VistA
- Extremadura (LinEx)
- BSI (Ägypten)
- Camden Town
- SPIP

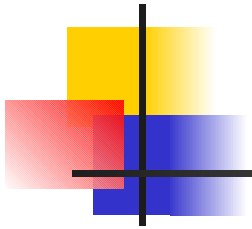


10 steps for interaction

- 1. Identify the “seed”
  - Free software is not developed in a vacuum: it starts with an impetus – usually a kernel of the desired software
  - “Please community, write us software for our needs” will not get much support
  - PAs can develop a core of software, alone or (ideally) in a pool with other PAs, then release it to a wider community

10 steps for interaction

- 2. Reach out to the community
  - If nobody knows you, they cannot respond to your needs
  - Disseminate your “seed” software widely to the target communities of user-developers
  - Channels from PA-specific (IDABC OSO), regional associations (France: AFUL), general (SourceForge, Berlios, etc)



10 steps for interaction

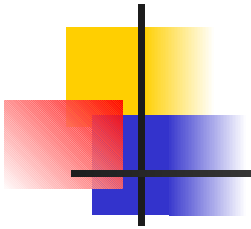
- 3. Attract a community to solve problem
  - Obvious, but requires that PAs show that they have tried to solve it themselves
  - Building upon a community's work may make this easier. Extremadura chose Debian to create GNU/LinEx
  - Easier when a large PA user base exists, related communities form (e.g. MMBase)

10 steps for interaction

- 3b. Attract a community to solution
  - Alternative to (3), if a partial solution exists, attract a community to work on it (planting a “tree” rather than “seed”!)
  - Possible when PA develops a full application, but would like to share, or get additional support from a community
  - e.g. VistA

10 steps for interaction

- 4. Provide a platform (?)
  - Creating a development platform can create an interaction channel, but hard to exploit
  - e.g. BerliOS, successful in attracting a large European community, not exploited by PAs
  - Schoolforge.org – made by the community!
  - Platform supported by coalition of PAs?

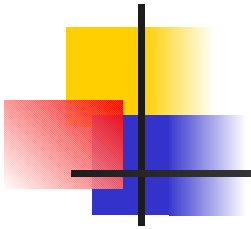


10 steps for interaction

- 5. Proactively cooperate, feedback
  - Main problem in PA interaction with free software is often lack of communication between users, IT staff, developers
  - Free software works on a “user-pull” rather than “provider-push” development model
  - Users must contribute, but don't have to write code. “Active” vs “passive” use is valued by the community. (Linus)

10 steps for interaction

- 6. Identify community leaders
  - In self-organising communities, leaders are self-selected, by showing the most initiative
  - Identifying, recognising and interacting with leaders is helpful in influencing output
  - Creating leaders is even better (initiative from PA staff, or hire leaders)



10 steps for interaction

- 7. Identify competition model
  - Free software uses Darwinian competition
  - PAs need to make their preferences heard, to ensure that their criteria select “winners” among competing development choices
  - Companies effectively do this by hiring people with influential views in the community's technical selection process

10 steps for interaction

- 8. Identify funding methods
  - Funding may (will?) be required
  - Small (potential) user base may require more direct funding for community
  - Or – creation of (sub)-community of PA staff and PA contractors, with an automatic self-interest in PA-specific software with a small potential user base

10 steps for interaction

- 9. Monitor and evaluate
  - Continuously monitor extent of community, level of community interest and level of PA-community interaction
  - Cycle back to feedback (5), leadership (6) and financial support (8) when interest is flagging

10 steps for interaction

- 10. Involve other PAs!
  - Usually, single-PA efforts can succeed in attracting a wider community only if they are high-profile, general interest projects (Extremadura)
  - PAs should work in groups. This is most effective because they are the community.
  - Community of PAs can reach out to the wider community more naturally (Adullact)

Legal frameworks: overview

- Matrix of three factors:
  - Type of software solution
  - Type of service
  - Type of partnership

Legal frameworks: overview

- Type of software solution
  - General interest (office suite...)
  - Specialised (CMS, workflow, collaboration tools used by but not specific to PAs)
  - PA-specific (land record management; in Europe, usually also health, education)

Legal frameworks: overview

- Type of service
  - Development from scratch
  - Customisation: adaptation of externally developed solution to local or PA-specific needs
  - Attracting external support and improvement for software developed by PA

Legal frameworks: overview

- Not all 3x3x3 combinations are common
- Need awareness of legal framework before starting initiative
- 5 common cases are described in detail in report, available on OSO

More information

- "Guidelines for Public Administrations on Partnering with Free Software Developers" - OSO study: europa.eu.int/idabc/en/document/3879/471
- FLOSS project & related research www.flossproject.org
- FLOSSPOLS – govt use: flosspols.org
- **Feedback: gposs@cec.eu.int**