

# Methods of Attraction

---

How to bring in more contributors?

Matthias Hopf

# (Almost) All OpenSource Projects

---

- ▶ Are understaffed
- ▶ Have more open bugs and ideas than people dealing with them
- ▶ Don't have a marketing facility
- ▶ Most don't have a dedicated project manager alike position either
  
- ▶ Requires alternative structures

# X.org in Particular

---

- ▶ **Very understaffed**
  - Code size
  - Usage
- ▶ **Few new participants lately**
  - Never been a school kids' project, though

# X.org Strengths and Weaknesses

- ▶ Pro: has a dedicated board
  - X.org BoD might fill in the marketing facility position
  - Might certainly not fill in the project manager position
- ▶ Pro: many areas for potential contributors
  - Existing drivers still need a lot of polishing
  - New drivers for handheld devices required
  - Documentation still lacking
  - Mesa still way below OpenGL 4.x
- ▶ Con: complex environment, not sexy
  - Required to be more vocal, more advertizing based than some other groups
- ▶ Con: existing developers mostly employed long-term contributors
  - Can be intimidating

# Facilities

---

- ▶ X.org
  - Not driving development direction itself
- ▶ Distributions
  - Interested mostly in hardware support, directly usable features
- ▶ Users (Industry)
  - Interested mostly in required features
- ▶ Spare time contributors
  - Apart from academia the origin of open source
  - Limited possibility of support in monetary aspects
- ▶ Academia
  - Addressing OpenSource typically in a broad way, i.e. not X.org specific

# Methods

---

- ▶ Easygoing communication
- ▶ Being cool
- ▶ Working development environment
- ▶ Ease of gaining speed
- ▶ Subvention programs
- ▶ Education
  
- ▶ Gaining **and** keeping contributors is important
  - Short-term contributors with high cost prohibitive

# Communication

- ▶ Communication open, upfront
  - Often said to be the #1 contributor killer
  - Often said to be the #1 female contributor killer
  - Even if only code is criticized, not people
- ▶ People not used to push their own development
  - First-time contributions left stranded **almost** done
- ▶ Really as problematic as seen from the outside?
- ▶ Distinctive skill set required
  - Verbose, criticism tolerant, persisting, open minded
  - How to communicate this to potential contributors?
- ▶ How to improve stickiness?

# Coolness

- ▶ Some projects are „just kewl“ - e.g. the Linux kernel
  - Attraction just by being there
  - Self-accelerating effect
  - Only problematic in little used subsystems, old drivers, etc.
- ▶ Being BIG is not enough (LibreOffice)
- ▶ Being big in the media isn't enough (Firefox)
- ▶ Being technologically interesting isn't enough (Mesa)
- ▶ Being used by everybody isn't enough (Xserver)
  
- ▶ Requires *very* strong leadership
  - Difficult with projects that never (or too long ago) were a one-man show
- ▶ Don't think we will be able to change that...

# Development Environment

## ▶ Code Repository

- git: better than sliced bread
- Private repos – information difficult to find

## ▶ Acception

- Patches (answering rate, acceptance rate)
- New subprojects (drivers, subsystems)
- Non-code contributions

## ▶ Documentation

- Especially structure

## ▶ Quality and accessibility of non-code services (Mailing lists, Bugzilla, Wiki, ...)

- How to change this other than by contributors (chicken/egg)?

## ▶ License (MIT-alike vs. GPL)? Don't think so.

# Easy Startup

- ▶ Openness
  - Almost trivial with git-based projects
- ▶ Ease of development infrastructure
  - Way more difficult with git
  - Still not understood generally – needs more advertizing
- ▶ Startup documentation
  - Membership
  - Starting points
- ▶ Mentoring
- ▶ Low-hanging fruits for starters

# Employment

---

- ▶ Main source of contribution ATM
  - Distributions
  - Hardware vendors
  - Software vendors
  - High-end using industries
  
- ▶ Good for base speed in development

# Subvention Programs

- ▶ Google Summer of Code
  - Some success of long term contributors already
  - Still a low rate
- ▶ X.org's Endless Vacation of Code
  - So far only as addon for good evaluated gSoC projects with too little slots
  - No promotion beyond this original idea
  - Low success rate so far
  - Lessons learned: pay only *after* fully successful evaluation
- ▶ How to improve rate?

# Education

## ▶ Industry: Trainees

- At SuSE successful in other areas

## ▶ Academia

- Promoting OpenSource OSes and tools
- OpenSource classes (tools/environments/concepts)
- Fixing bugs as class work for getting a glimpse in these environments
  - X.org's Bugzilla probably not suited – bugs often highly hardware dependent, intermittent, difficult to reproduce, let alone fix
  - List of low- to middle-hanging fruits needed

## ▶ Can't say much about success – yet :^)