Upstream, How?

Daniel Vetter, Intel SATG
OSP 2023
Recap

- upstream open source = efficient, maximally competitive market
- understand your upstream business case:
  - know which software market to rid of margins & profits
  - have a complementary product to sell
Individual Engineers

- no market value, no salary
- just write proprietary code
- open source, as a hobby
- for the stubborn: this talk
- https://xkcd.com/2347/
Individual Engineers

- have a union, for collective bargaining power
  - maybe call it upstream community, foundation, ...
- make sure your union is big enough
  - can be really small in a niche project
- union dues = networking across companies
- union negotiators = upstream maintainers
- so important, companies might found your union for you
  - e.g. Linux Foundation Core Infrastructure Initiative
Upstream Community: Goals

• defend the competitive market
• tie-in with upstream business cases
  • or upstream engineers wont earn money
  • this is hard
  • broadest adoption to spread the costs
• radical customer orientation for maximal adoption
  • radical = it will hurt
Community Power Structure

- competitive market = level playing field
- linux kernel biggest employer of contributors: Intel, ~10%
- linux kernel biggest employer of maintainers: RedHat, ~13%
- key people employed by the foundation directly
- no CLA, special rights, controlling or majority votes, …
- if not: open source R&D becomes a donation to the special company, not a write-off
Community: Barriers to Contribute

- efficient market = no barriers to entry/exit
- clear&accessible contribution process
- ... across all levels, from first bug report/small patch to becoming a maintainer
- public CI, test suites, ..., open everything including planning
- clear decision and escalation structure
- code of conduct and globally welcoming culture
- ...
- if not: R&D cost shouldered by too few players, impossible to achieve the efficient market
Community Leadership

- building a good community is hard work
- and it requires skills & experience
- libraries of books
- entire conferences or conference tracks dedicated to community leadership topics
- personally like unconference style meet-ups
- beware of trying to transfer management skills, the power structure is fundamentally different
Engineering Org

- sell complementary product
- reward sales
  - both direct sales
  - and marketing, brand recognition, goodwill, ...
- do not incentivize production of «valuable IP»
  - open source has no economic value!
- radical customer orientation
  - direct connection to customer's engineering org
Engineering Org: Sunk Cost

- open source has at most zero book value
- downstream forks have less than that: maintenance, support, …
- many engineering orgs in denial, still reward «valuable IP»
- upstreaming has immediate additional costs, while downstream fork costs long term
- employ upstream maintainers&engineers to stay relevant&competitive with evolving upstream code base
  - or can't sell your complementary product anymore
- this is a hard sell, needs all the help from the upstream community
Upstream Engineering Org

- stay relevant with evolving upstream
- protect the upstream business case over the long term
- upstream engineers & maintainers must defend the competitive market, or they become irrelevant themselves
- direct conflict with short term sales department goals
- maybe fix your conflict of interest policy, e.g. RedHat
- fundamental challenge of upstream open source engineering
Conflict of Interest: The Bad

- downstream first for maximal sales
- upstream eventually
- upstream engineers misincetivized, will quit
  → defacto upstream never
- can last for a very long time due to denial about negative book value of produced software
Conflict of Interest: The Ugly

- upstream first, port to downstream
- downstream features only as tactical exceptions
- downstream features only if *additional* sales beats downstream fork costs
  - adjust your contracts for upstream rejections, require upstream when buying
- reward long term, sustainable sales
Conflict of Interest: The Good

• ...
Engineering Org Culture

- upstream open source culture shock
  - sales instead of R&D
  - brutally competitive market
  - fundamental conflict of interests
- a few years for individual engineers to adapt, if they do
- decade plus for entire engineering org, if successful
  - e.g. Microsoft
- easy to dismiss early signs as «arrogant» customers
- be first, or you're guaranteed to be too late
Summary & Questions

- individual engineer: have a union
- upstream community defends competitive market
  - level playing field and no barriers to contribute
- engineering org: sales department
  - fix rewards and conflict of interest policies
- business strategy
  - execution: upstream first and be first, or bust
  - upstream business case: destroy a market and sell a complementary product