

オープンソース開発でのスキル

Linus 君と仕事をしてボクが習ったこと

Junio C Hamano
<junio@kernel.org>

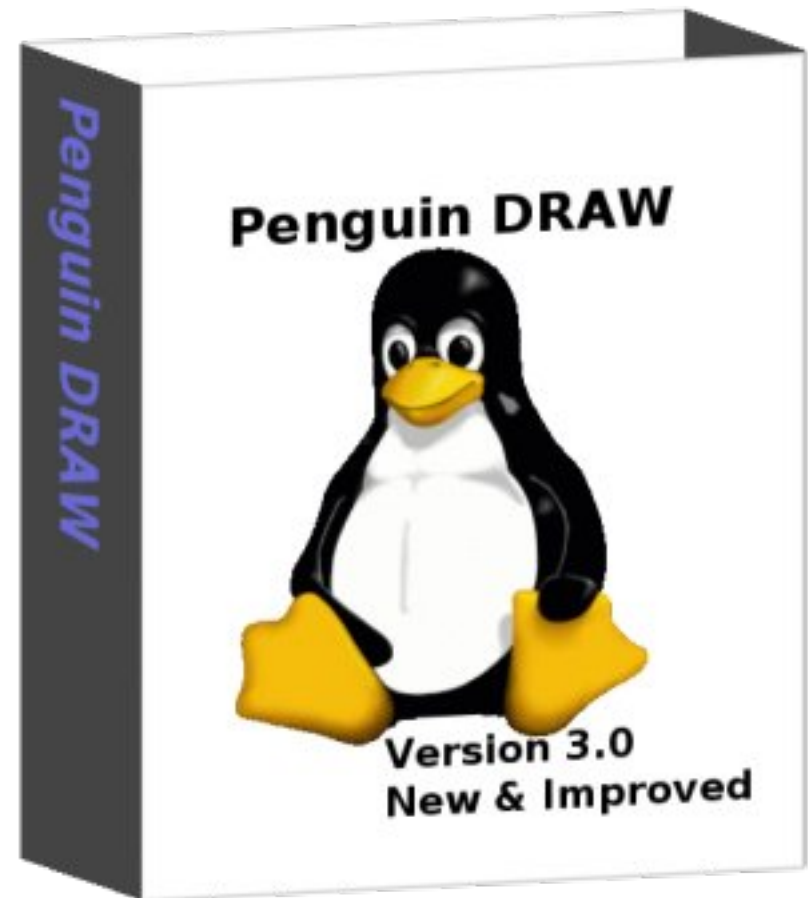
開発者になること

オープンソースの開発者って
どうやってなるの？



開発者になること

- (クローズド)
A社の製品ソフト “X”
- (オープン)
???



オープンソース

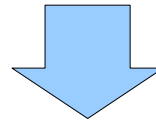
- エンドユーザとして使用
 - 「何か使い勝手悪いなあ」
 - 「こういうの、どうやってやるんだろ」
- メーリングリスト・ニュースグループに参加
 - 「あ、他の人があんな事やってる」
 - 「こうすれば良いんだよ」
- 他ユーザのヘルプ・バグ報告・パッチ投稿

オープンソース

- ソースが公開されているから、単にユーザで居つづけるだけではなく、自分で改良できる。

オープンソース

- ソースが公開されているから、単にユーザで居つづけるだけではなく、自分で改良できる。



- その変更点が「みんなに評価されて使って貰える」のはまた別の話。

「認められる」のは、積み上げ

- 他ユーザの手助け、文書化。
- 競争ではない。
 - 急がず、正確に。
- 下手な英語を恐れない。
 - でも丁寧に推敲。
- 間違っただことを言わない。

Engrish

上手に説明する

- 「どうやってほしいか」、という内容より、
 - ◆ 「何を目的に」、
 - ◆ 「なぜ」、の方が重要。
- メーリングリストでは誤解されるのが当たり前。
 - ◆ 感情的になるのは禁物

プロジェクト全体をまず考える

- プロジェクト・リーダーを困らせない
- 自分にしか使えない変更？ 十分に一般的？
- 他の変更との競合？
- 互換性？
- リリース・スケジュール？

他人を上手に使う

- バグ発見と報告
 - ついでにおねだり
- やらせてみて
 - 失敗させて、気付かせる
- プロトタイプして
 - 助けを求める
 - 穴埋めさせる



コミュニティを大切に

- 貢献者を粗末にしない
 - 「無駄な仕事」の代案は、より秀れていることが明白でないといけない
 - 的確な説明
 - 人の仕事を全く無駄にはしない
- 貢献者を退屈させない

コミュニティを大切に

- ユーザを粗末にしない
 - ◆ エンドユーザの声を拾う
 - ◆ トラブルシューティングに付き合う
 - ◆ 互換性のない変更
 - 避けられないなら仕方ない
 - 逃げ道を残す

最後は人間同士のつながり

- 幼稚園で教わった常識
 - ◆ 話せばわかる。わかるまで話す。
 - ◆ けんかをしない。
 - ◆ 思いやりを持つ。
- 信用第一。