

株式会社クリアコード

須藤功平

関西Ruby会議2017
2017-05-27

テーマ

コミュニティ と ビジネス

実行委員長曰く

“コミュニティとビジネスは
いい関係でまわると思うんで
そのための情報を共有する場
になるといいな

実行委員長”

内容

- ✓ 一本のストーリーではなく
テーマに関する話の詰め合わせ
- ✓ 聞く人それぞれへの期待：
 - ✓ 参考になる何かが見つかるといいな

題材

ClearCode

クリアコード

- ✓ 設立11年目の会社
- ✓ 業務：受託開発と技術サポート
- ✓ 小さな会社：9人（募集してはいる）

大事にしていること

フリーソフトウェアの推進

(Rubyも多くのgemもフリーソフトウェア)

と
稼ぐこと
の
両立

推進するだけだとダメ

- ✓ 稼がないと継続できない
- ✓ そうなると推進し続けられない

稼ぐだけだとダメ

- ✓ クリアコードをやっている
意味がない
- ✓ 他の開発会社でいいよね

両立の実現に向けて

✓ クリアコードが活用するもの

✓ フリーソフトウェア開発で学んだ事

理念：<https://www.clear-code.com/philosophy/>

✓ コミュニティーで学んだことを
ビジネスに活かしている

関西Ruby会議2017のテーマにあっていそう！

学んだこと1

問題は
upstreamで直す

考えてみよう

使用ライブラリーに問題発見！

(たとえばRailsとかActiveなんとかとか)

あなたならどうする！？

1. 手元で回避
2. 別のライブラリーを探す
3. そのライブラリーを直す

クリアコードのスタイル

3. そのライブラリーを直す (そうじゃないときもあるけど基本的にはそう)

- ✓ 自コードのメンテ性アップ
 - ✓ 回避コードの独自メンテが不要
 - ✓ 回避コードの共有が不要
(使っているプロジェクト全部で回避が必要)
- ✓ 他の人も困らなくなる

フリーソフトウェア開発の世界

- ✓ そのライブラリーを直すのは当たり前
- ✓ フリーソフトウェアが大事にしていることの1つは「直せること」！
- ✓ クリアコードはここで学んだことを実践しているだけ

コミュニティとビジネス

- ✓ ビジネスでの知見を
コミュニティへ
 - ✓ ユースケースを元に改良・修正
- ✓ なぜやるの？
 - ✓ クリアコードのポリシーだし！
 - ✓ 「メンテしやすい」は普通に
ビジネス的に割に合う

クリアコードでの実例

Rubyにメモリーリークあり

- ✓ Ruby製のデーモンで問題に
- ✓ 調査→パッチ作成→Rubyを修正
- ✓ 定期的に再起動という回避策もあるだろうけど、本家で修正

これで社員がコミット権をもらっていた

技術的な詳細：デバッグ力：よく知らないプログラムの直し方

<http://www.clear-code.com/blog/2011/12/6.html>

学んだこと2

開発を
続けられる
コードを書く

考えてみよう

ファーストリリース完了！
その後は？

1. 納品して終わり
2. 運用チームにボタンタッチ
3. 継続的に開発

クリアコードのスタイル

3. 継続的に開発

- ✓ フリーソフトウェア開発ではリリースを重ねつつ改良が普通
(Rubyも20年以上も改良を続けている)
- ✓ 最初から完璧な状態はわからない
- ✓ 使ってフィードバックを得てよりよい状態を探していく
- ✓ よい状態の維持にも継続は必要
(よい状態は時間が経つと変わる)

継続的に開発とビジネス

- ✓ 長期間一緒にやれる仕事を優先
- ✓ ビジネス的にもよい：
 - ✓ 安定収入
 - ✓ 知識が増えてよりうまく開発できる

うまく開発

開発効率が
上がる

だけじゃない！

クリアコードのうまく開発

- ✓ よりよい提案をできる
 - ✓ お客さんもうれしい
クリアコードもうれしい、そんな案
- ✓ お客さんのうれしい
 - ✓ よりよい成果（儲かるとか）
- ✓ クリアコードのうれしい
 - ✓ フリーソフトウェアの推進

クリアコードと提案

- ✓ 既存の実現方法は
フリーソフトウェアの推進が
弱いことが多い
 - ✓ それはしょうがない
 - ✓ 推進も加えるところが
クリアコードの存在意義
- ✓ 両立する提案に必要なもの
 - ✓ お客様の目的の理解

フリーソフトウェア開発の世界

- ✓ ユーザーがパッチ提供
 - ✓ 開発者：目的（ユースケース）確認
(Rubyへの機能提案時もユースケースは？と聞かれる)
 - ✓ 開発者：提供されたパッチ以外の方法での実現もありえる
- ✓ ユーザーとプロジェクトで
メリットの両立を目指す
 - ✓ クリアコードも同様にしているだけ

クリアコードでの実例

Groongaサポート

(Groonga : フリーソフトウェアな全文検索エンジン)
(商用パッケージに組み込み)

- ✓ 改良はGroongaへマージを提案
- ✓ お客さん専用ブランチは作らない
- ✓ mrubyのバグはmruby修正を提案
(Groongaはmrubyを組み込んでいる)
- ✓ 独自パッチを抱えない

Groongaサポートと提案

- ✓ 変更はupstreamへ
 - ✓ →成果がフリーソフトウェアに
(クリアコードのメリット)
- ✓ メンテナンスコストが下がる
(クリアコードのメリット)
 - ✓ →サポート費用を抑えられる
(お客様のメリット)

学んだこと3

相手が
想像しなくても
わかるように
説明する

学んだこと4

楽しく開発する

学んだこと5

非難するよりも
手を動かす

学んだこと6

回避策よりも
根本解決

主な業務内容

✓ 受託開発

✓ FLOSSサポート

歴史的な経緯でFLOSSのまま
昔は「フリーソフトウェア」か「OSS」かで迷っていた

✓ OSS開発支援

サービス名ではフリーソフトウェアではなくOSSにしている

受託開発

✓ 優先度

a. フリーソフトウェアを 直接的に推進できる仕事

例：成果をフリーソフトウェアにできる仕事

事例：SMSさんの基盤ソフトウェア開発

b. フリーソフトウェアを 間接的に推進できる仕事

例：利用フリーソフトウェアの改良も含む仕事

事例：Groongaクライアント用gem各種を使う仕事

c. いずれ推進につながりそうな仕事

FLOSSサポート

導入支援と 障害調査

よくあるやつ

導入支援

✓ 事例：Firefox/Thunderbirdの
企業導入 (Ruby関係ないけど…)

✓ インストーラー作成キット開発
<https://github.com/clear-code/fx-meta-installer>

✓ アドオン開発 (機能制限・一括設定)
<https://github.com/clear-code/>

✓ ↑はフリーソフトウェアとして公開

✓ ノウハウはFAQとして公開
<https://github.com/mozilla-japan/enterprise/blob/master/FAQ.md>

障害調査

- ✓ "ソースがあればなんとかなる"
- ✓ フリーソフトウェアが大事にしていることの1つは「調べられること」!
- ✓ 調査対象のバグとわかったら
upstreamにフィードバック
- ✓ フリーソフトウェアの推進

OSS開発支援

- ✓ お客様のOSSの開発に参加
 - ✓ 事例：
Treasure DataさんのFluentd関連
- ✓ お客様のOSS開発を支援
 - ✓ 事例：SpeeeさんのWebapp Reviewee

お客様のOSSの開発に参加

ダイレクトに大事なことを両立

クリアコードの大事なこと：

✓ フリーソフトウェアの推進

✓ 稼ぐこと

どうして仕事になるか

- ✓ OSSで稼いでいる企業が増えた
- ✓ ちゃんとOSSを開発できることをOSSを開発することで示せている
- ✓ OSSの開発が営業活動になっている
- ✓ コミュニティー活動がビジネスへ

お客様のOSS開発を支援

ダイレクトに大事なことを両立

クリアコードの大事なこと：

✓ フリーソフトウェアの推進

✓ 稼ぐこと

どうして仕事になるか

- ✓ OSSの活用が当たり前になった
- ✓ OSSのエコシステムに参加する
メリットが認知されつつある
- ✓ 支援してくれそう感が出ている
 - ✓ OSSの開発や発表などが営業活動
 - ✓ コミュニティー活動がビジネスへ

OSSのエコシステムに参加

どうということ
だろうか？

OSSのエコシステムに参加

“自分たちのソフトウェアと
OSSを同じように扱う

須藤功平”

同じように扱う

- ✓ 問題があれば直す
- ✓ 気になるところがあれば共有

いつも通りでいい！
OSSは身近なもの

仕事の作り方

お客さんに見つけてもらう

お客さん探しを頑張らない

- ✓ クリアコードは大衆受けすることをしていない
- ✓ マッチするお客さんは少ない
- ✓ 少ないお客さんを探し回るパワーはない
 - ✓ そこに力を入れるよりも得意なことを伸ばしたい

諦めることは大事

- ✓ クリアコードは小さな会社
- ✓ すべてに全力投球はできない
- ✓ なにを諦めるかを決断
 - ✓ なにかに力を入れるということは
なにかを諦めるということでもある

決断

- ✓ 社長職の面白いところでもあり
難しいところでもある
- ✓ 決断する基準があると楽しめる
(自分の中で基準がはっきりしていないうちは大変だった)
- ✓ クリアコードの基準
 - ✓ フリーソフトウェアの推進と
 - ✓ 稼ぐことの両立

お客さん探しと稼ぐこと

- ✓ 単に探す事をやめては稼げない
 - ✓ 探さないで稼げないか考える
- ✓ 基本的にまずは両取りを狙う
 - ✓ さっきは諦めるっていったけどさー
 - ✓ 探す事は諦めるが稼ぐ事は諦めない
 - ✓ トレードオフっぽく見えるやつでも
トレードオフじゃないかも！？
と考えてみる

推進と稼ぐことの両取り

- ✓ フリーソフトウェアの推進を
がんばる
 - ✓ 開発・パッチ提供・レビュー
 - ✓ 情報公開：ブログ・発表
- ✓ ↑で見つけてもらいやすくなる
 - ✓ 見つけてもらおう→仕事→稼げる！
 - ✓ 両取り！

推進→見つけてもらった例

- ✓ milter manager開発・発表
(Rubyを組み込んだ迷惑メール対策ツール)
- ✓ →問い合わせ
- ✓ →大規模システムへの導入支援
(Rubyで独自メールフィルターも開発)

推進→見つけてもらう

- ✓ フリーソフトウェア推進自体が営業活動
- ✓ 推進と稼ぐの両取りの一例
- ✓ コミュニティー活動がビジネスにつながる例

最近の推進兼営業活動

Apache Arrow

Apache Arrow

- ✓ データ分析システム用
次世代データフォーマット
- ✓ RubyもApache Arrowを扱えると
データ分析できそう！
 - ✓ なので扱えるように開発に参加
- ✓ Rubyでデータ分析する仕事に
つながるといいな

お知らせ

✓ Apache Arrow勉強会開催！

- ✓ Apache Arrowが早くデファクトになるとRubyでデータ分析しやすくなる日も近づくはずなので普及活動中

✓ 明日の午前に大阪で

[https://classmethod.connpass.com/
event/56478/](https://classmethod.connpass.com/event/56478/)

採用

- ✓ マッチする人は少なそう
(仕事と同じ傾向な気がしている)
- ✓ RailsでWebアプリ！とか
自社プロダクト〇〇を開発！とか
とは違いそう
- ✓ 業務内容ではなくポリシーでマッチ

採用活動

- ✓ フリーソフトウェアの推進
 - ✓ 見つけてもらう
 - ✓ 一緒にフリーソフトウェアを開発
→ よさそうな人を誘う
- ✓ RubyKaigiにスポンサー
 - ✓ Rubyを応援したので始めた
 - ✓ 採用は期待していなかった

まとめ1

“コミュニティとビジネスは
いい関係でまわると思うんで
そのための情報を共有する場
になるといいな

実行委員長”

まとめ2

- ✓ 一本のストーリーではなく
テーマに関する話の詰め合わせ
- ✓ 聞く人それぞれへの期待：
 - ✓ 参考になる何かがあったならいいな

コミュニティとビジネス

この話をしたことが
仕事につながると
すごくキレイなのでぜひ！

ClearCode