最良の砂場を目指して
プライベート開発を
支える技術

@__timakin__

自己紹介

- twitter: @__timakin__
- github: timakin
- 株式会社トランスリミット
- 主に業務ではゲームの
  クライアントサイドの開発(Cocos-2dx)
- プライベートではGo, Rubyでライブラリの開発等

自己紹介

- twitter: @__timakin__
- github: timakin
- ~~**株式会社トランスリミット**~~
- ~~主に業務ではゲームの~~

  ~~クライアントサイドの開発(Cocos-2dx)~~
- プライベートではGo, Rubyでライブラリの開発等

自己紹介

- twitter: @__timakin__
- github: timakin
- **無職**
- プライベートではGo,
  Rubyでライブラリの開発等

本日のアジェンダ

- エモい話をします！失敗すると無限に流行りの技術への入門をする羽目になる中で、有意義なプライベート開発をするには？
- 長いプライベート開発の中で見つけたコツ
- 自分はどうトライして、
  結果どういうことができたか

長いプライベート開発の中で見つけたコツ

(エモい)コツまとめ（私見）

- よりよいプライベート開発は、より良いミドルウェアから。
- 最初長く続けることを目標に、あとあと高品質なものを短時間で作るのを目標に。
- タスク管理を業務レベルで行う。
- 他人も持っている課題にアプローチする。
- それなりに大きい会で発表することを前提に。
- ネット上ではできるだけ多くの人の目に触れるように。

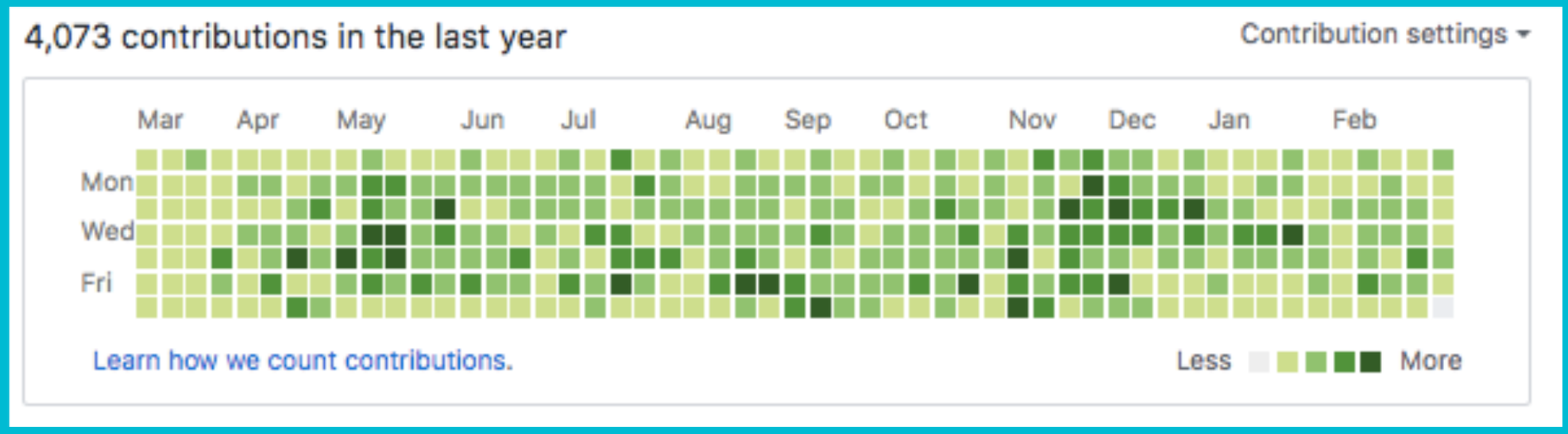
より良いプライベート開発は、より良いミドルウェアから。

- どれだけ長期間やっても、個人で開発してるものは、大規模な集団で開発しているコア機能には勝てない。

- 特定の分野だけ、高速だったり操作が簡潔なミドルウェアをつくることで、自分の個性を出すことができる。

- 自動化用のツールは個人よりも組織的大規模的な問題にフォーカスされていることが多く、微妙に手の届かない部分がプライベート開発ではでてくるので、解決が必要。

最初長く続けることを目標に、
あとあと高品質なものを短時間で
作るのを目標に。



- 短時間でガッと作った後燃焼しきって次のコードが書けない等ある。

- まず「長期間」開発することで、そもそもその機能が必要か、設計が妥当かを考える時間を用意しないと、「実際に業務でこんなコード書いたら殺される」というコードを量産する羽目に。

タスク管理を業務レベルで行う

- 見積りの経験、タスクの粒度感の整理、事前の整理による開発のスピードアップ等が目的
- めっちゃ業務に役立つ
- なんとタスク管理は電車等でも楽々できる。空き時間はタスク整理をする。
- Slackの自分用チャンネルとかも駆使。

他人も持っている課題にアプローチする。

- まず同年代の友達を持っておく。

- これはいい課題なのでは、と思った時に友達に「これは御社だとどうやって解決してる？そもそも課題になってる？こういう機能があったら便利かな？」というのを聞いてから方針を立てる

- 一人用のライブラリみたいなのを作ってもGithubのスターがつかず完全に承認欲求が満たされないので、他人に自信を持って売り込めるようなものを作るとメンタル面に良い。

それなりに大きい会で発表することを前提に

- 自分の開発は所詮入門程度だし...と
  思うと辛い。さらにはそこに甘んじる。

- その技術の先駆者とつながりを持って、質問・宣伝できる状態じゃないと、改善作業が上手くいかない。

- **承認を得る**

- だいたい先駆者のブログ記事とか読んで開発するんだから、その記事でわからないことあったときに質問できないと、どこかで限界が生じる。

ネット上では多くの人の目に触れるように

- 宣伝活動は海外から。githubに置くのだけではなく、Hackernews, Reddit, 言語別のニュースサイトに掲載する。
- 海外からの方がスターもIssueももらいやすい。市場調査みたいなものがしやすい。
- 発表する時に「なんと海外で人気」というと、フォローしてくれる人が増える。

(エモい)コツまとめ（私見）

- よりよいプライベート開発は、より良いミドルウェアから。
- 最初長く続けることを目標に、あとあと高品質なものを短時間で作るのを目標に。
- タスク管理を業務レベルで行う。
- 他人も持っている課題にアプローチする。
- それなりに大きい会で発表することを前提に。
- ネット上ではできるだけ多くの人の目に触れるように。

自分はどうトライして、
結果どういうことができたか

timakin's 砂場

- バックエンドはRails、フロントは微妙にReact。ミドルウェアや外部のクローラーサーバーでGo
- プロビジョニングツールとしてItamae。
- サーバーはAWS(EC2, S3, CloudFront)上に。Circle CIで継続的インテグレーション。
- NewRelic、Errbitでエラー監視。
- Dockerビルド。
- 検索、ログ解析でElasticSearchを利用。
- ステージング/プロダクションが分かれていて、ちゃんと検証できる環境を用意。

timakin's 砂場

- 中規模以上の環境（業務レベル）を想定したクオリティで、一個のアプリの開発環境を用意。
- コアな機能は保守的に、周辺機能は前衛的に。
- モデルが多くなってきたので、Serviceレイヤを挟んで
  管理を行う等の工夫をしている。たまにActionCableとか使ってRails5らしくやってます。

```
+----------------------+-------+-------+---------+---------+-----+-------+
| Name                 | Lines |   LOC | Classes | Methods | M/C | LOC/M |
+----------------------+-------+-------+---------+---------+-----+-------+
| Controllers          |  2539 |  2063 |      43 |     291 |   6 |     5 |
| Helpers              |   274 |   159 |       0 |      17 |   0 |     7 |
| Jobs                 |     9 |     8 |       2 |       1 |   0 |     6 |
| Models               |  2519 |  1529 |      74 |     143 |   1 |     8 |
| Mailers              |    20 |    18 |       3 |       2 |   0 |     7 |
| Channels             |    37 |    28 |       3 |       5 |   1 |     3 |
+----------------------+-------+-------+---------+---------+-----+-------+
| Total                |  6853 |  4908 |     125 |     568 |   4 |     6 |
+----------------------+-------+-------+---------+---------+-----+-------+
```

砂場で開発した周辺物

- Railsアプリ用のgemとGoのミドルウェアがメイン
- capistrano3-ridgepole
  - capistrano経由でridgepole(Migrationツール)を叩く
- itamae-plugin-resource-npm
  - itamaeでnpm installを行えるプラグイン。
- jf
  - 便利jsonフォーマッター
- gonvert
  - クローラーサーバー用に作った文字コード変換ライブラリ
- gopli
  - ローカル、ステージング、本番間のDBレプリ用のCLIツール
- autocompl
  - Rails5でのオートコンプリート実装用プラグイン

プライベート開発で出会う危険な場面

- 最初から大規模の設計をしようとする。
    - 必要になったタイミングで修正しない限り、無駄な設計のせいで書き直しコストがかさむ
- メインのフレームワークで個性を出そうとする。
    - だいたいすでに誰かが通った道を通ることになるのが、大きめのフレームワークにありがちなこと。個性を出そうとして車輪の再発明をしがちだが、最初気分が良くてもすぐ虚しくなる。
- なんでも新規
    - Go、Ruby以外に手を出したら多分失敗していた。自分以外でも開発できるようなレベルに新規性を止めるには？と考えて開発しないと、実際の現場での技術選定とかに役立つ知見は多分得られない。

宣伝活動

- Goのミドルウェアを書くようになってから、周りにあんまりなさそうなツールというのが書けるようになってきた。
- ライブラリの宣伝の結果、勉強会で発表して4stars、海外のhackernews、redditで宣伝すると、10~260stars。

結果得られたもの

- 職
- Issueを通じた意見やエラー報告
- 設計への関心、学習機会
- 新規性のある技術を採用する場面の見極め方
- 規模に比例して徐々に現場レベルの知見が得られる

現在やっていること

- Goでクローラーサーバー書いてます。
  https://github.com/timakin/grawler
- DDDベースの設計、GAEに乗っけてCron実行もGAE提供の機能を使う。
- AWSとはだいぶ勝手が違うが、薄いフレームワークでGoのサーバーを書く新規性のある体験を実現できる。CLIとかだとまだ現場に通用しないんじゃないかという焦りがつきまとう。

まとめ

- より良いプライベート開発を得るには？
  - 周辺機能で個性を出しつつ、適度な新規性を。
  - 中規模開発を行い、現場に通用するノウハウを得る。
  - 設計はするべきだが、その都度必要な範囲で設計を。
  - 最低限勉強会やカンファレンスで発表するところを目指す。