

git cococoとRailsアプリ開発

西田雄也

2018-03-31 kunibiki.rb#7

自己紹介

- 西田雄也
- Twitter, GitHub: @nishidayuya
- 趣味:
 - 小さいものを書く
 - スノーボード
 - なんとかGo(昨年末にTL40)

git

cococoっ

てなに？

git cococoってなに？

実行したコマンドとともにコマンドによる変更をコミットするコマンド

```
$ git cococo bundle update
```

1. コマンド実行前にコミット漏れがないか確認
2. コマンド実行
3. 変更点をコミット
 - もう一度同じことをするコミットメッセージとともに

git cococoってなに?

実行したコマンドとともにコマンドによる変更をコミットするコマンド

```
$ git cococo bundle update
```

1. コマンド実行前にコミット漏れがないか確認
2. bundle updateを実行
3. 「run: git cococo bundle update」といったコミットメッセージで変更点をコミット

もう一度同じことをするための コミットメッセージ

- コマンド実行はいいとしてコミットメッセージを書くときにコピー&ペーストするの面倒.
- 毎回「どんな風を書いてたっけ…」って思いながらgit logで調べて同じように書くの面倒.



これを解決!

- 実行時に一回だけ書けば良い
- コミットメッセージはコマンド任せ

動機

松江Ruby会議08のa_matsudaさんの話

- 「コード生成するコマンドを実行した場合は、そのままコミットするのが望ましい。」
 - 読みたいのは人間が書いた部分
 - コマンドでコード生成した部分は実行したコマンドがわかればいい

テ毛

デモ: Railsア プリケーション ン開発でよく ある使い方

構成

- git cococoはBourneシェルのシェルスクリプト
 - 当初はRubyで実装していた.
 - Windowsで動かしたくなった.
 - mruby-cliもあるけど…
 - Gitが動く環境にはBourneシェルあるいは互換のシェルがある.
- 自動試験はRubyで書いている.
 - 単体テストとインテグレーションテスト
 - test-unitが試験を書きやすい.

設計
思想

設計思想: インストールが簡単

- 1ファイルのシェルスクリプトで実装してある.
- ダウンロードして実行パーミッションを与えるだけ.

```
$ wget https://raw.githubusercontent.com/nishidayuya/git-cococo/master/exe/git-cococo
$ chmod a+x git-cococo
$ mv git-cococo move-to-PATH-env-directory/
```

設計思想: 動く環境はGit

- Gitが動作する全ての環境で動く(はず).
 - はず… 全ての環境で試験できません!
- 次の環境でCIをまわしている.
 - Ubuntu
 - Windows
 - MacOSX
- 動かないUnixあるいはUnix互換OSがあれば教えてください&プルリクエスト歓迎です.

設計思想: 余計なことはしない

- 当初はコミット漏れがあれば自動的にgit stashすることを考えていた.



- 使ってみたところコマンド実行に失敗したときにstashしたままになるが、自動的にやっちゃうとstashしたものが消えたように見える.
 - 焦っているときは特に困る.



- --autostashオプションで明示的に指示する形とした.

設計思想: Gitと合わせる(1)

--autostashオプションについて

- 当初は--auto-stashというオプション名で考えていた.



- git rebaseに--autostashオプションがあることに気がついて揃えた.

設計思想: Gitと合わせる(2)

git cococoという名前について

- Gitにはgit rerereコマンドがある。
 - REuse REcorded RESolutionの略



- git cococo
 - COmmit COmpletely COmmand outputの略
 - 意味が合っている。
 - 短い。

ご静聴ありがとうございました

インストールは

<https://github.com/nishidayuya/git-cococo>
より

気に入ってくださったらスターとかいただけると
~~調子に乗ります~~ 励みになります。