Legacy Hacks

“@Tomohiro”, TAIRA

First.

Legacy System

代替すべき新しい技術などのために古くなったコンピュータのシステムや技術

Legacy Code

テストコードがないコード

思い浮かべてみよう

Scratch

リファクタリング

立ちはだかる壁

- RDB で正規化されてない
- グローバル変数の嵐
- 超巨大な関数
- テストコードがない
- 仕様書がない
- そもそも仕様があやふや

継続的改善

I.アーキテクチャ

レガシーシステム

VS

モダンフレームワーク

ミスマッチ

ライブラリ

- Zend Framework
- PEAR
- PHPUnit
- Smarty
- APC
- RequireJS with jQuery

車輪の再発明

を恐れるな

オレオレライブラリ

- Like RoR ActiveRecord
- Like RoR ActionController
- Like RoR script/generate
- Like jQuery

e.g. ActiveRecord

- 複合主キー
- データベース接続先を変更できる

Just in Time

必要なものを必要な時に必要な
量だけ生産する

依存しない

提供されているライブラリを使う場合，できるだけそのライブラリに依存しないようにする

継承せず

ゲートウェイを

2. テスト

テスト

テスト

テスト

安心感

編集して祈る

↓

保護して変更する

UI のテストは目視で

3. 開発者の士気

ツールへのこだわり

ツール

- Subversion
- Redmine
- Testing Framework
- VMware ESXi
- オレオレツール

ランチミーティング

開発者向け

ドキュメントの整備

4. 結合

ラッパーで結合

新ライブラリの機能を API として提供する(グローバル関数など)

人海戦術

全機能の
テスト仕様書

人か(ry

5. ドッグフード

感謝の気持ち

「ありがとう」

ソースコードの結晶が

きれいに！

継続的改善