

ソフトウェア開発における再利用のための バッドノウハウパターン

Tatsuhiko Miyagawa

miyagawa@bulknews.net

Livedoor, Co.,Ltd. / Shibuya Perl Mongers
Bad Knowhow Conference 2004

アジェンダ

- **バッドノウハウパターンとは？**
 - **なぜ必要か**
- **バッドノウハウパターン紹介**
 - **具体例**
 - **傾向と対策**
- **まとめ**

バッドノウハウへの心構え

- **バッドノウハウは必要悪**
 - どんなソフトウェアにもノウハウはある
 - 「近頃の新人はなんでもバッドノウハウだといって覚えようとしなない」
- **エンジニアとしての心構え**
 - <http://blog.bulknews.net/mt/archives/000835.html>
 - 「エンジニアの価値はBKの数で決まる」
 - 誤解されているようだ

バッドノウハウの数がエンジニアの価値!?

- 1個1個の BK に大した価値はない
 - 深追いした過程こそが重要
 - なぜそのノウハウは「バッド」なのか？
 - 大局的視点を身につける
 - メタバグデータベースとしての利用
- BK を有効活用する
 - バッドノウハウの「匂い」
 - パターン分類

バッドノウハウパターン

- 「不吉な匂い」を嗅ぎわける
 - この道はいつか来た道
 - 君子バッドノウハウに近寄らず
- コミュニケーション(愚痴)を円滑に！
 - 「今日△△でハマってさ～」
 - 「あ、それ○○パターン！」
- アンチパターン
 - ソフトウェア開発者として

実践バッドノウハウパターンカタログ

実例協力: Satoru Takabayashi

Historical Reason パターン

■ 症状

- 歴史的経緯により放置されたバグ、わかりにくい仕様
 - バグでもそのほうが便利なので仕様になったり

■ 実例

- mod_rewrite: RewriteRule と RewriteCond の解釈順は設定ファイルの下から上
- Apache モジュールの MODULE_VAR_EXPORT 構造体 → 順番ぐちゃぐちゃ、マクロの嵐

■ 対策

- 枯れたソフトウェアはドキュメントが豊富
- 深追いせず Google に聞く

互換性ないがしろパターン

■ 症状

- ころころ仕様が変わるため、バージョン間の微妙な差異に悩まされる。
 - Historical Reason パターンの逆
 - 当然、枯れていないソフトウェアに多い

■ 実例

- autotools: バージョン上げると必ずはまる
- Perl DBI: Tim Bunce 仕様変えすぎ

■ 対策

- 深追いする前にバージョン確認

フルスクラッチパターン

■ 症状

- 下位互換性がないメジャーバージョンアップ
 - 開発者の気まぐれ
 - 過去のバージョンを捨てられないユーザ
 - 2個の系列が存在するハメに

■ 実例

- Apache2: モジュールに互換性ないので移行できない
- Perl6: vaporware

■ 対策

- 仕事で使うなら枯れてる方、Hack するなら新しい方

Undocumented Feature パターン

■ 症状

- ドキュメントに書いていない仕様、バグ
 - 実装の都合上の制約
 - ドキュメントにあるが見つけにくいものも含まれる

■ 実例

- Apache: AddModule の順番により予期したとおり動かないことがある (Documentated)
- Perl: `/.{1,$n}/` $n > 32767$ で落ちる (Documentated)

■ 対策

- 対策しようがない (ソースを読むぐらいか?)
- 見つけたらドキュメントパッチを送る

柔軟すぎる設定ファイルパターン

■ 症状

- 設定ファイルが柔軟すぎて何でもできる
 - 設定ファイルで解決するとスマートな気がする（幻覚）
 - 奥が深い症候群

■ 実例

- `mod_rewrite: [P] [C] [L]` 後から見ても全然わからん
- `procmail: “:0” $MATCH` 意味わからん

■ 対策

- 後から見る人のこと(自分含む)を考えよう

mod_rewrite のすばらしいところは、Sendmail のような設定性と柔軟性を与えてくれるところだろう。また、mod_rewrite のよくないところは、Sendmail のような設定性と柔軟性を与えてくれるところだろう。 -- Apache URL Rewriting Guide

不親切なデフォルトパターン

■ 症状

- デフォルト設定が不親切: 使い始めにいきなりつまずく
 - セキュリティとかが理由であることが多い

■ 実例

- cvs: **初回**の cvs login で .cvspass: No such file or directory → touch ~/.cvspass **で解決**
- Emacs: **バックスペースでヘルプが表示される**
- screen: C-a **が使えない**

■ 対策

- **設定ファイルを人からもらう**

センスレスパターン

■ 症状

- API にセンスがないため直感的でない、覚えられない

■ 実例

- ImageMagick: **画像のサイズ取得** image->columns, image->rows → **ダサー**
- Perl 5.8 Encode.pm: encode, decode と from_to で引数の順番が違ふ → **覚えられんよ**
- gnuplot: **独自ミニ言語最悪**

■ 対策

- **ラッパー関数をつくる(グッドラッパー!)**
- **捨てる**

バッドラッパーパターン

■ 症状

- 他のソフトのラッパーをつくったつもりが、自身がバッドノウハウになってしまった

■ 実例

- Webmin
- linuxconf
- autotools
- CF

■ 対策

- 捨てられるなら、捨てる
- グルに聞く

同じ名前で非互換パターン

■ 症状

- 同じ名前なのに環境によって仕様が異なる
 - OS 付属のライブラリ
 - GNU もの
 - 紛らわしい

■ 実例

- libc
- iconv
- gettext

■ 対策

- すべての環境をあわせる(無理)

不親切なメッセージパターン

■ 症状

- エラーメッセージが不親切で何が間違っているのかわからない

■ 実例

- SuExec のエラー: premature end of headers
- rpm: imlib2-devel-1.0.6-1.i386.rpm: read manifest failed: **成功です**

■ 対策

- そのエラーメッセージでググってみる
- 冗長オプション、デバッグオプションを探す

I18Nパターン

■ 症状

- 他の言語で動かない、マルチバイトが通らない

■ 実例

- zphoto: スウェーデンではガンマ値を 1.0 ではなく 1,0 と書くそう。locale 厄介
- Namazu: LANGUAGE が設定されていないとデフォルトでわかち書きしない
- SJIS: Shift_JIS, CP932, Windows-31J

■ 対策

- 全部ASCIIでやる + 国際化オタクに援軍を頼む

バグの濫用パターン（ユーザ）

■ 症状

- 実装のバグを利用してあやしいTipsを生み出す
 - いつ変わるかわからないのに ...

■ 実例

- IE が CSS のアンダースコアを認識しないことを利用してクロスブラウザなCSSを実現
- Perl: `my $foo = 1 if 0;` コンパイラの最適化バグを利用してクロージャ作成

■ 対策

- 自己責任

まとめ

- **まとめてはみたが ...**
 - あるあるネタになってしまった
 - あてはまらないものも結構多い
- **パターンの功罪**
 - パターンにまとめて悦に入りがち
 - 新たな BK を見つけたときに分類する楽しみ
(本末転倒！)
- **Let's face it, Bad Knowhow is fun.**