



プログラミングを
簡単にするために
プログラミングする

3種類のプログラム

- 納品するプログラム（商品）
- テストプログラム
- 開発者（自分）のためのプログラム

納品するプログラム

- 要求された要件を満たすプログラム
- 普通はプログラム全体を自分が作ることはほとんどなくて、一部の開発をしたりするはず
- 小規模な製品・システムならば全体を作る場合もあるかもしれないけど
- 製品全体としての開発管理をなされていて、すべてのソースコードのバージョン管理は、全体方針に従う必要があったりする

テストプログラム

- 納品するプログラムは要件を満たしていることを示すために、テスト内容やテスト結果も、商品の一部として納品したりする
- 実際の運用上で動作するわけでないのでメモリ容量を気にせず比較的に雑に作ったりしてもあまり問題にならなかつたりする
- テストプログラムは開発者が自己管理しなくてはならないことが多いと思われる

開発者のためのプログラム

- 開発における定型的な作業を自動化したり高速化したりするためのプログラム
- 手作業でやった場合とプログラムを実行した場合で、結果は同じになることもあるが、効率が全然違ったりする
- 書き捨てだったり、適当に管理されていたり（管理されていなかったり）する
 - 使う人と作る人が別ならば、ちゃんと管理したほうがトラブルにならずにすむかもしれない

カイゼンのために開発

- 注目したいのは, **3** 番目の「開発者のためのプログラム」について
- たとえばどんな言語を使っていたりするのか
 - **DOS** バッチファイル (**Windows** バッチファイル)
 - **PowerShell**
 - **shell (bsh, csh, bash)**
 - **Makefile**
 - **perl, python, ruby**
 - **Excel VBA**
 - **C** 言語

開発??

- 作業の自動化をするために必要なのは、プログラム言語とは限らない
 - 秀丸マクロ
 - キーボードマクロ
 - マウス自動操作ソフトウェア
- 自動化するための、自動実行スケジューリング
 - **cron**
 - **Windows** のタスク機能

オススメの言語

- 覚えておいて損がないのは、多分 **perl**,
python, **ruby**
 - (ある程度) クロスプラットフォームだと思う
 - 使える幅が広い
- でも、必ずそれがベストの方法であるというものではないため、状況にあわせて、もっとも適したものを使うように、使い分けるべき
 - **C** 言語だって結構使えるよ？

一応、薦める理由とか

- 自分は **Perl** が好きなのです.
- ちゃんとしたプログラムも書ける.
- (でも、人の読めないようなちゃんとしてないプログラムも書けてしまっただけ)
- 開発環境を整えるのがわりと簡単
- ライブラリ管理 **"CPAN"** がすげーいい
 - **rpm** とか **apt** みたいな感じで **perl** のモジュールをインストールできる
 - ライブラリの **man** を見たりソースを見たりもできる

最近の自分の実用例とか

• テストプログラム

- ハードウェアの動作確認のため、テスト用プログラムを作成したりした
- テストプログラムとテスト結果 (テスト動作時のシリアルログ) を自分用の **SVN** で管理
- 開発中のデバイスドライバに、テスト用の非公開 **API** を追加して、そこからテスト向けのコンパイルオプションを有効にしたときだけコンパイルされるテスト関数をたたくようにした
- テストプログラムの設計はわりと適当. コピペしまくりだし, つくりも粗雑. まあ, テストできれば問題ない.

実例その2

- 実装中のソースコードがどれくらいの量なのか
いちいち数えたくない
 - 自分専用の **SVN** リポジトリの **svn log** から、変更の
あったリビジョンを抽出して、**svn diff** コマンドで変更
差分を参照して、各リビジョンの追加行数、削除行数を
自動で数えるようなプログラムを **Perl** で作成した
 - **Perl** は **Windows** でも **ActivePerl** が使えて便利
 - **4** 半期報告とかしなきゃいかんはずなので、それに備えて
一応用意してみた。（まだどこにもデータとかは出して
ないけど）