



Perl

Improving Quality in Perl.

Adam John Trickett

www.iredale.net
adam.trickett@iredale.net
PGP Key: 0x166C4BF0



Quality via Consistency

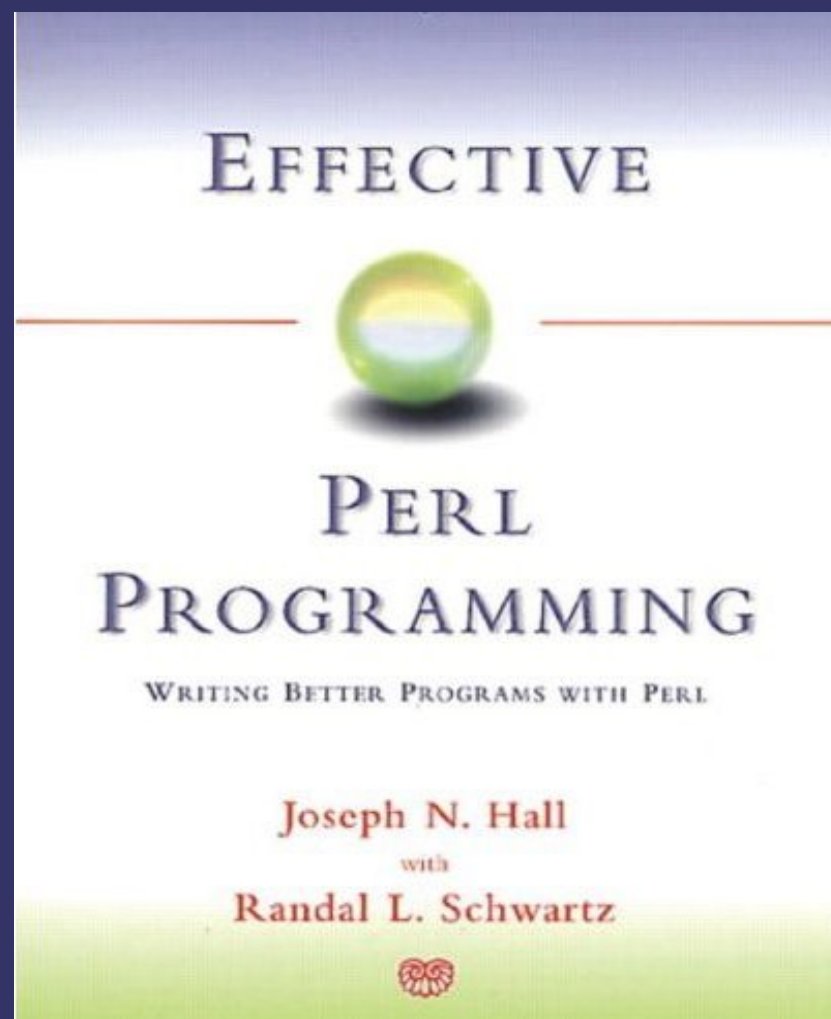
```
#!/usr/bin/perl  
our $global_standards;  
my $local_standards;  
use Test::More;  
use Software::Version::Control;
```

Write a coding standard and stick to it!



use Perl;

- Write your Perl programs in Perl, NOT another programming language
- Write in idiomatic Perl
- Use built-ins and modules not external calls





Low Hanging Fruit...

- There are things we can all agree that we should always do
- Always do the things everyone agrees we should do
- Aim to be 100% Perl Kwalitee compliant



Quality and Kwalitee

- Quality is easy to understand... but...
 - Hard to measure
 - Hard to compare from one program to another
 - Therefore very hard to measure in an automated way
- Kwalitee
 - Easy to measure
 - Easy to compare
 - Easy to compare automatically



Kwalitee

- Certain measurable features are correlated to good quality
- Therefore measure what can be measured and report kwalitee instead
- Most kwalitee markers are known good practices
- Calculated automatically when modules are uploaded to CPAN:
 - <http://cpants.perl.org/>



Some Kwalitee Features

- extractable, extracts_nicely
- has_readme, has_manifest, has_buildtool
- no_symlinks
- has_tests
- has_version, has_proper_version
- proper_libs
- no_pod_errors, has_test_pod, has_test_pod_coverage
- manifest_matches_dist



Best Practices

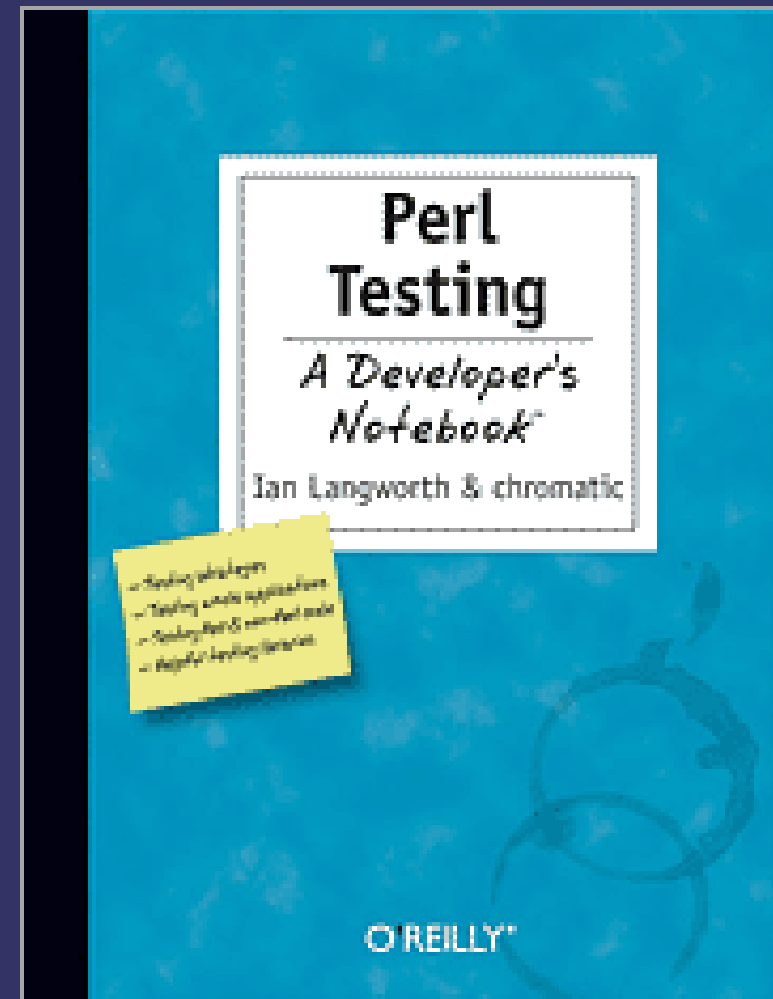
- Create a local list of coding rules AND stick to them
- Start with PBP but don't follow slavishly
- Use Perl Tidy to keep code looking standardised
- Use Perl::Critic to scan for deviations from PBP





Include a Test Plan

- Test everything!
- Unit tests
- Integration tests
- Regression tests
- Test your tests
- Be “Test Driven”
- A **day** in development,
a **week** in testing,
years in production!





Refactor

- Refactor old code
- Update to best practices
- Use tests to ensure functional compliance
- Remove dead and non functional code
- Remove **Red Flags**
- Improve documentation



Use Source::Control

- Use revision control: CVS; SVN or whatever
- Always run your tests before committing
- Back-up your repository
- Use issue tracking software, e.g. RT



Sites

- Red Flags Repair Shop
 - <http://www.perl.com/pub/a/2000/04/raceinfo.html>
 - <http://www.perl.com/pub/a/2000/11/repair3.html>
 - <http://www.perl.com/pub/a/2000/06/commify.html>
 - <http://perl.plover.com/flagbook/>
- The CPAN Testing Service
 - <http://cpants.perl.org/>
 - http://schwern.org/~schwern/talks/CPANTS/full_slides/



Books

- Effective Perl Programming
 - Hall & Schwartz, Addison-Wesley, 0-201-41975-0
- Perl Best Practices
 - Conway, O'Reilly, 0-596-00173-8
- Perl Testing: A Developer's Notebook
 - Langworth & chromatic, O'Reilly, 0-596-10092-2
- *Perl Medic*
 - *Scott, Addison-Wesley, 0-201-79526-4*



Thank You.

Any Questions?