Last night I drifted off while reading a Lisp book.

Huh?

Suddenly, I was bathed in a suffusion of blue.

At once, just like they said, I felt a great enlightenment. I saw the naked structure of Lisp code, unfold before me.

My God

It's full of err's

The patterns and Metapatterns danced. Syntax faded, and I swam in the purity of Quantified Conception. Of ideas manifest.

Truly, this was the language from which the Gods wrought the universe.

No, it's not:

It's not?

I mean, ostensibly, yes. Honestly, we hacked most of it together with Perl.
Prefer ‘Good Developers’ over ‘Good Languages’

“For all program aspects investigated, the performance variability that derives from differences among programmers of the same language—as described by the bad-to-good ratios—is on average as large or larger than the variability found among the different languages.”

Who am I?

Tim Bunce
Author of the Perl DBI module
Using Perl since 1991
Involved in the development of Perl 5
“Pumpkin” for 5.4.x maintenance releases
http://blog.timbunce.org
~ Myths ~

Perl is dead

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5
Perl is dead

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5
Perl 5 isn’t the new kid on the block
Perl is 21 years old
Perl 5 is 14 years old

A mature language with a mature culture
Perl 0: embryo
Perl 1: infant
Perl 2: toddler
Perl 3: child
Perl 4: preteen
Perl 5: adolescent
Buzz != Jobs

Perl5 hasn’t been generating buzz recently
It’s just getting on with the job
Lots of jobs
    - just not all in web development
Guess the Languages

Job Trends

www.indeed.com

“web developer”

Yes, Perl is growing more slowly than others but these are just “web developer” jobs
Perl is mentioned in many more software engineer/developer jobs.
Perl is the primary focus of more developer jobs.
Want a fun new job? Become a Perl developer!
Massive Module Market

Large and vibrant developer community
Over 15,000 distributions (58,000 modules)
Over 6,700 ‘authors’ (who make releases)
One quarter of all CPAN distributions have been updated in the last 4 months!
Half of all updated in the last 17 months!
Top Modules

Many gems, including...

DBI  DBD::*  DBIx::Class  Rose::DB::Object
Catalyst  Moose  DateTime
Algorithm::*  Statistics::*  Thread::*
XML::*  HTML::*  WWW::*  Parse::*
Net::*  Email::*  POE::*  Locale::*
Test::*  Devel::Cover  Perl::Critic  perltidy
Comprehensive Perl Archive Network
360 mirrors in 51 regions (TLDs)
Developer Services

Upload a perl module distribution and you automatically get...

- global *distribution and archiving*
- *namespace* ownership and management
- a *bug tracking* queue at rt.cpan.org
- a *forum* at cpanforum.com
- *smoke testing* on many platforms
search.cpan.org

For each and every distribution:
Browse well formatted inter-linked docs
Links to forums, bug tracking, ratings, annotated documentation, dependency analysis, smoke test results, and more...

620,000 unique visitors per month
140,000 page views per day
41,000 visits per day
# Dependency Analysis available for all Modules

[http://bbbike.radzeit.de/~slaven/cpantestersmatrix.cgi](http://bbbike.radzeit.de/~slaven/cpantestersmatrix.cgi)

<table>
<thead>
<tr>
<th>Module</th>
<th>Status</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moose</td>
<td></td>
<td>16 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Test::LongString</td>
<td></td>
<td>108 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Test::Builder</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Scalar::Util</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Test::Exception</td>
<td></td>
<td>28 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Test::Harness</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Sub::Uplevel</td>
<td></td>
<td>34 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Exporter</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Carp</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Filter::Simple</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Class::MOP</td>
<td></td>
<td>20 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>File::Spec</td>
<td>Core module</td>
<td></td>
</tr>
<tr>
<td>Sub::Name</td>
<td></td>
<td>16 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Sub::Exporter</td>
<td></td>
<td>81 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Params::Util</td>
<td></td>
<td>99 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Sub::Install</td>
<td></td>
<td>102 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td>Data::OptList</td>
<td></td>
<td>89 PASSes, 0 UNKNOWNs, 0 FAILs, 0 NAs</td>
</tr>
<tr>
<td><strong>Chance of success</strong></td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>
Average Uploads/Month

Created on: Wednesday January 30, 7:33 PM
~ Myths ~

Perl is dead

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5

BUSTED!
~ Myths ~

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5
“True greatness is measured by how much freedom you give to others, not by how much you can coerce others to do what you want.”
—Larry Wall
Guidelines and Tools

Perl Best Practices
Perl::Tidy
Perl::Critic
Test::*
Devel::Cover
Perl::Tidy

Perl code beautifier
Works beautifully - can be trusted
Supports many options for personal styles
Perl Best Practices recommended options
Perl::Critic

Static Code Analysis for Perl
Includes over 120 policies
Most based on Perl Best Practices
Grouped into levels and themes
Configurable and extensible
for local policies and styles
Perl culture takes testing seriously

Excellent mature tools for testing

Test Anything Protocol - producers/consumers

Test::* modules make it easy to write tests

Over 200 Test::* distributions on CPAN
Perl Testing
A Developer's Notebook
Ian Langworth & chromatic

- Testing strategies
- Testing whole applications
- Testing Perl & non-Perl code
- Helpful testing libraries
Devel::Cover

Coverage Analysis for Perl
Tells you what code has been executed
Statement, branch, condition, subroutine, pod
Produces drill-down reports in HTML
# Devel::Cover Reports

http://pjcj.sytes.net/cover/latest/

<table>
<thead>
<tr>
<th>File</th>
<th>stmt</th>
<th>bran</th>
<th>cond</th>
<th>sub</th>
<th>pod</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcePerl-1.91</td>
<td>72.48</td>
<td>52.74</td>
<td>43.18</td>
<td>74.66</td>
<td>56.36</td>
<td>64.15</td>
</tr>
<tr>
<td>Algorithm-Annote-0.10</td>
<td>100.0</td>
<td>50.00</td>
<td>n/a</td>
<td>100.0</td>
<td>0.00</td>
<td>87.80</td>
</tr>
<tr>
<td>Algorithm-C3-0.06</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Algorithm-Dependency-1.102</td>
<td>75.13</td>
<td>41.60</td>
<td>31.43</td>
<td>85.00</td>
<td>92.11</td>
<td>67.14</td>
</tr>
<tr>
<td>Algorithm-Diff-1.1902</td>
<td>87.47</td>
<td>78.68</td>
<td>77.66</td>
<td>80.77</td>
<td>100.0</td>
<td>83.85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>file</th>
<th>stmt</th>
<th>bran</th>
<th>cond</th>
<th>sub</th>
<th>pod</th>
<th>time</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>lib/Algorithm/Dependency.pm</td>
<td>95.2</td>
<td>83.3</td>
<td>n/a</td>
<td>93.3</td>
<td>100.0</td>
<td>34.4</td>
<td>92.7</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Item.pm</td>
<td>100.0</td>
<td>50.0</td>
<td>33.3</td>
<td>100.0</td>
<td>100.0</td>
<td>8.2</td>
<td>86.8</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Ordered.pm</td>
<td>100.0</td>
<td>75.0</td>
<td>n/a</td>
<td>100.0</td>
<td>100.0</td>
<td>9.3</td>
<td>93.7</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Source.pm</td>
<td>86.3</td>
<td>55.0</td>
<td>46.7</td>
<td>90.9</td>
<td>100.0</td>
<td>22.2</td>
<td>75.5</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Source/File.pm</td>
<td>100.0</td>
<td>50.0</td>
<td>n/a</td>
<td>100.0</td>
<td>100.0</td>
<td>7.8</td>
<td>85.2</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Source/HoA.pm</td>
<td>100.0</td>
<td>50.0</td>
<td>n/a</td>
<td>100.0</td>
<td>100.0</td>
<td>0.5</td>
<td>90.5</td>
</tr>
<tr>
<td>lib/Algorithm/Dependency/Weight.pm</td>
<td>97.7</td>
<td>44.4</td>
<td>n/a</td>
<td>100.0</td>
<td>100.0</td>
<td>13.6</td>
<td>85.9</td>
</tr>
<tr>
<td>inc/Class/Inspector.pm</td>
<td>41.1</td>
<td>19.6</td>
<td>0.0</td>
<td>43.5</td>
<td>83.3</td>
<td>2.1</td>
<td>35.6</td>
</tr>
<tr>
<td>inc/Test/ClassAPI.pm</td>
<td>77.2</td>
<td>40.5</td>
<td>28.6</td>
<td>100.0</td>
<td>50.0</td>
<td>1.9</td>
<td>68.4</td>
</tr>
<tr>
<td>Total</td>
<td>75.1</td>
<td>41.6</td>
<td>31.4</td>
<td>85.0</td>
<td>92.1</td>
<td>100.0</td>
<td>67.1</td>
</tr>
</tbody>
</table>
package Algorithm::Dependency::Weight;

sub weight_all {
    my $self = shift;
    my @items = $self->source->items;
    defined @items[0] or return undef;
    $self->weight_hash( map { $_[0] } @items );
}
<table>
<thead>
<tr>
<th>line</th>
<th>l</th>
<th>l&amp;&amp;l&amp;</th>
<th>l&amp;l&amp;l</th>
<th>l&amp;l&amp;l&amp;</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>139</td>
<td>0</td>
<td>0</td>
<td>1037</td>
<td></td>
<td>defined $_[0] &amp;&amp; !ref($[0])</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1037</td>
<td></td>
<td>defined $<em>[0] &amp;&amp; !ref($[0]) &amp;&amp; $</em>[0] ne ''</td>
</tr>
<tr>
<td>140</td>
<td>1037</td>
<td>0</td>
<td>0</td>
<td></td>
<td>$$self{'loaded'} or $self-&gt;load</td>
</tr>
<tr>
<td>161</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td></td>
<td>$$self{'loaded'} or $self-&gt;load</td>
</tr>
<tr>
<td>182</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td></td>
<td>$$self{'loaded'} or $self-&gt;load</td>
</tr>
</tbody>
</table>
~ Myths ~

Perl is dead

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5
~ Myths ~

Perl is dead

BUSTED!

Perl is hard to read / test / maintain

BUSTED!

Perl 6 is killing Perl 5
“Perl 5 had already started dying, because people were starting to see it as a dead-end language.

It seemed odd at the time, but when we announced Perl 6, Perl 5 suddenly took on a new life.”

—Larry Wall, 2002

Perl 6 saved Perl 5!
Perl 6 saved Perl 5!

In 2000 perl was dying from the inside out
The Perl 6 RFC process “vented speen”
Perl 5 development has gone smoothly since
Much refactoring driven by Perl5-on-Parrot
Many new features inspired by Perl 6 work
Perl 5.10

Perl 5.10 was released in December 2007
Five years after 5.8.0, two years after 5.8.8

Refactored internals
   many fixes, more speed, less memory

Switch statement, smart matching, named captures, state variables, defined-or, say, field hashes, pluggable regex engines, trie-based non-recursive pattern matching, and more...

http://www.slideshare.net/rjbs/perl-510-for-people-who-arent-totally-insane
A Culture of Testing

Another bonus from Perl 6:

*Strong test suites for Perl 5 code are needed to ensure backwards compatibility*

Strengthened culture of testing
Perl Test Suite

2002: Perl 5.8.0 had 26,725 core tests
     +41,666 more for bundled libraries etc.

2007: Perl 5.10.0 has 78,883 core tests
     +109,427 more for bundled libraries etc.
Module Test Suites

CPAN Testers Network:

*Automated smoke testing of CPAN uploads*

Runs the test suite included in distribution

over 60 different platforms

over 20 different perl versions

Immediate feedback for developers
Platform/Version analysis available for all Modules

http://cpandeps.cantrell.org

<table>
<thead>
<tr>
<th></th>
<th>MSWin32</th>
<th>VMS</th>
<th>cygwin</th>
<th>darwin</th>
<th>freebsd</th>
<th>linux</th>
<th>netbsd</th>
<th>openbsd</th>
<th>solaris</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.11.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.9.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.9.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.9.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.6.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.6.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.5.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
~90,000 Reports/Month

Monthly - August 1999 to December 2007
~ Myths ~

Perl is dead
BUSTED!

Perl is hard to read / test / maintain
BUSTED!

Perl 6 is killing Perl 5
BUSTED!
~ Myths ~

Perl is dead

Perl is hard to read / test / maintain

Perl 6 is killing Perl 5

BUSTED!
Perl 6 Myths
“Perl 6 is not Perl”

“It'll never be finished”

“There's no code written in Perl 6”
“Perl 6 is not Perl”
“Perl 6 [...] is clearer, more direct, more expressive, and without many of the old false leads and rough edges in Perl 5.

So, Perl 6 is still Perl in that a programmer looking at a Perl 6 program will instantly recognize that it is "Perl", even if some of the details are different.”

—Larry Wall
“It’ll never be finished”
“If we'd done Perl 6 on a schedule, you'd have it by now. And it would be crap.”
—Larry Wall

"do it right" and "it's ready when it's ready"

“Truly radical and far-reaching improvements over the past few years.”
When will be be be done?

“We're not doing the Waterfall [model of development] we're doing the Whirlpool, where the strange attractor whirls around with feedback at many levels but eventually converges on something in the middle.”
—Larry Wall

in 'What criteria mark the closure of perl6 specification'

Specification is now relatively stable
“feedback at many levels”
Multiple implementations

Perl 6 compilers
Pugs - initial experimentation in Haskell
KindaPerl6 - perl5 - aiming to self-host
SMOP - Simple Meta Object Protocol
Rakudo - built on Parrot Compiler Toolchain
Sharing a common test suite
Pugs - 850,000 Lines of Code in 2 Years
“There's no Perl 6 code”
> 35,000 lines of tests written in Perl 6
> 10,000 lines of examples written in Perl 6
parrot
An advanced virtual machine for dynamic languages

Advanced capabilities with JIT compiling

Already supports over 50 languages about 20 of which are non-toy

Python, Ruby, PHP, Lua, Lisp, TCL, ...
Parrot development at r24049

revision number

0 5000 10000 15000 20000 25000

01/01/01 01/01/02 01/01/03 01/01/04 01/01/05 01/01/06 01/01/07 01/01/08
date

commits
releases
Parrot Test Suite

11,086 tests in 557 files
53,884 tests if run for all 7 runcores
(in Jan 2008, many more now)
Parrot Compiler Toolkit

Write a compiler in an afternoon!

“Parrot is really quite wonderful. [...] Parrot lets you implement your own languages using Perl 6 rules for the grammar and Perl 6 for the compiler.”

- Simon Cozens

http://blog.simon-cozens.org/post/view/1323
Perl 6 on Parrot

"Rakudo" short for "rakuda-do"
Japanese for "Way of the Camel"

"rakudo" also means "paradise"
Community Resources

“Awesome community. Perl people tend to be laid-back and friendly.”

perlmonks.org - Meditations and wisdom
use.perl.org - News and blogs
perlbuzz.com - Headlines and articles
comp.lang.perl.* - Newsgroups
Any questions?

http://perlmonks.org
http://search.cpan.org
http://blog.timbuunce.org