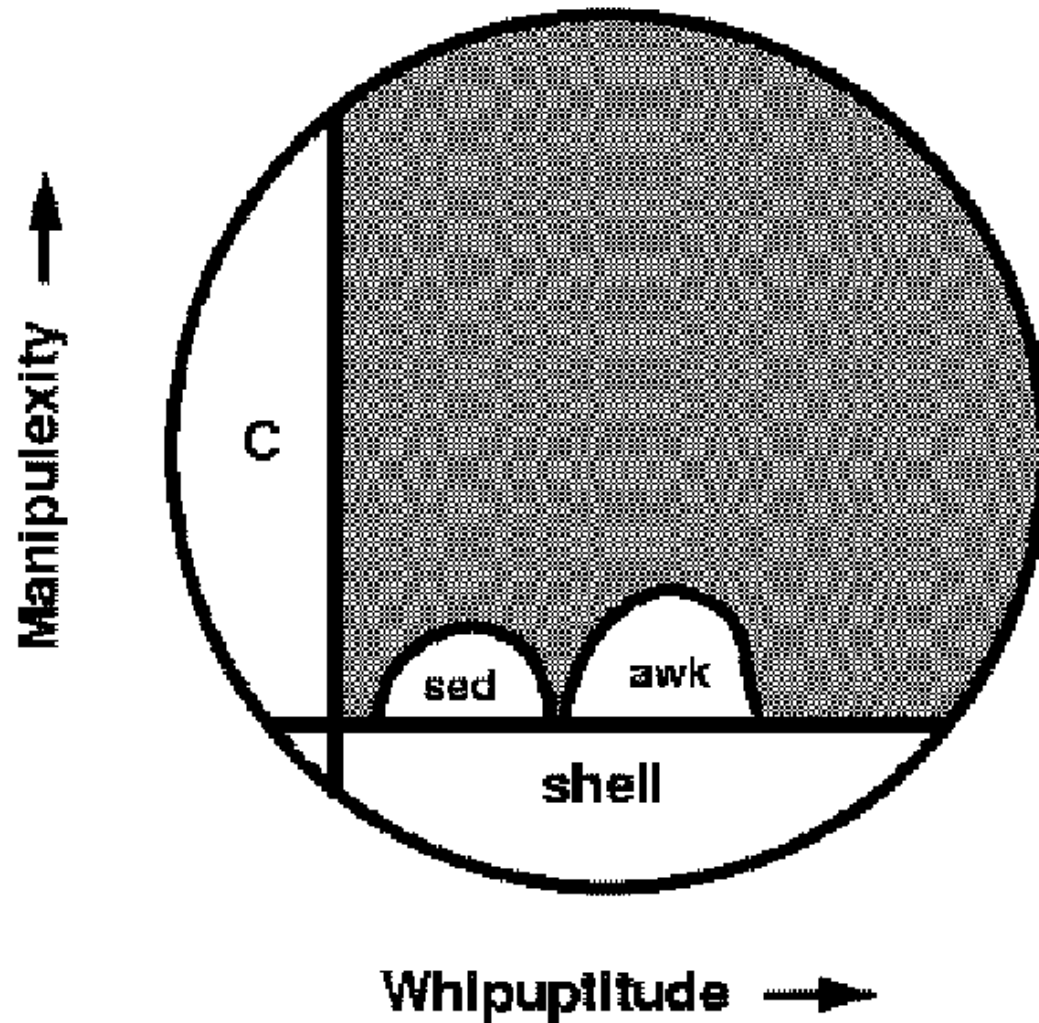


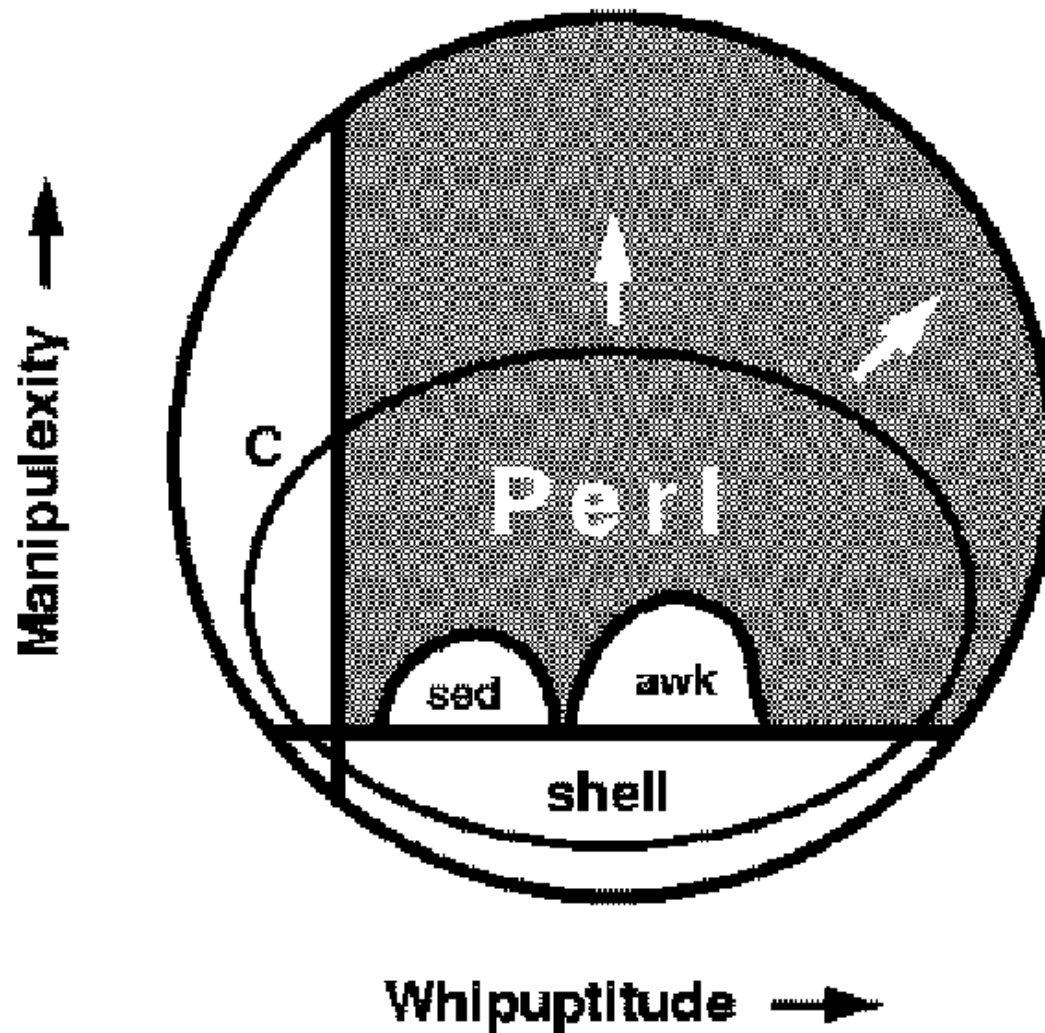
Modern Perl

Florian Ragwitz
Maik Hentsche

Es war ein mal



Es war ein mal



Es war ein mal

- Web Boom
- Dynamische Webseiten
- Baby Perl

HI!

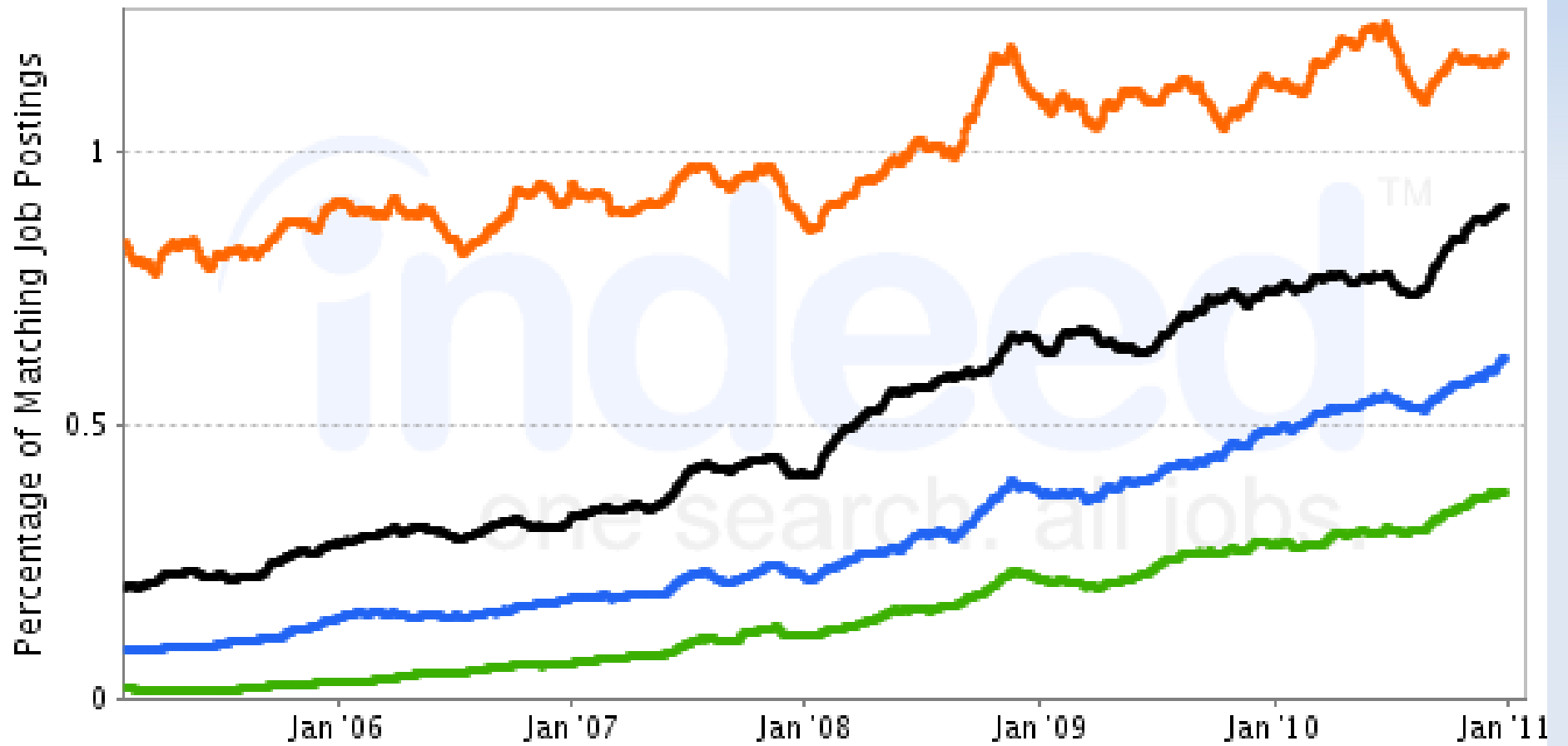
WE ARE THE PERL
COMMUNITY!

WE SUCK AT
MARKETING!



Job Trends from Indeed.com

— perl — python — ruby — php



Neues in Perl 5.10 (2007)

- Aktivieren mit "use 5.010", perl -E oder Boilerplate-Modulen
- kleine Helferlein
 - say - print mit Newline
 - // - defined or
 - given/when - switch-Statement mit smartmatch
- smartmatch
 - Do what I mean-Vergleicher
 - überladen oder Tabelle siehe perlsyn

Neues in Perl 5.12 (2010)

- regelmäßige Releases
 - jeweils im Frühling
- extensible Keywords
 - neue Keywords einfügen und Hooks für Parser definieren
- komplettester Unicode-Support
 - Unicode-Standard 5.6
 - "use utf8" erlaubt Unicode im Quelltext
 - schon länger implementiert"

Neues in Perl 5.14 (2011)

- Performance
- Komplettesterer Unicode (6.0) Support
- Weniger cruft
- Non-destructive substitutions und transliterations
- Extensibility

5.16 and beyond

- Neue features kommen rein
 - Extensibility
- Non-features fliegen raus

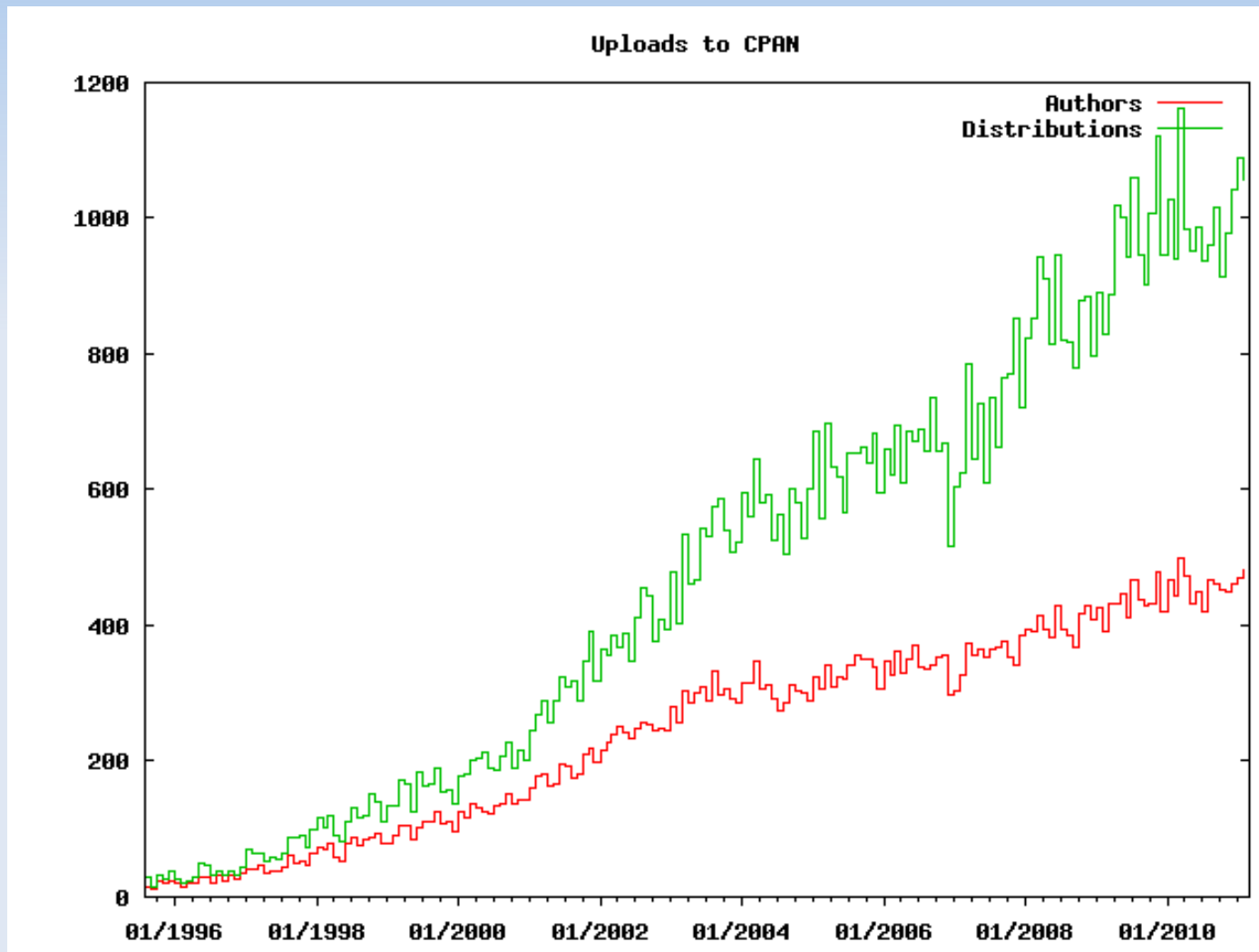
Perl was always designed
to be an evolving language.

- Larry Wall

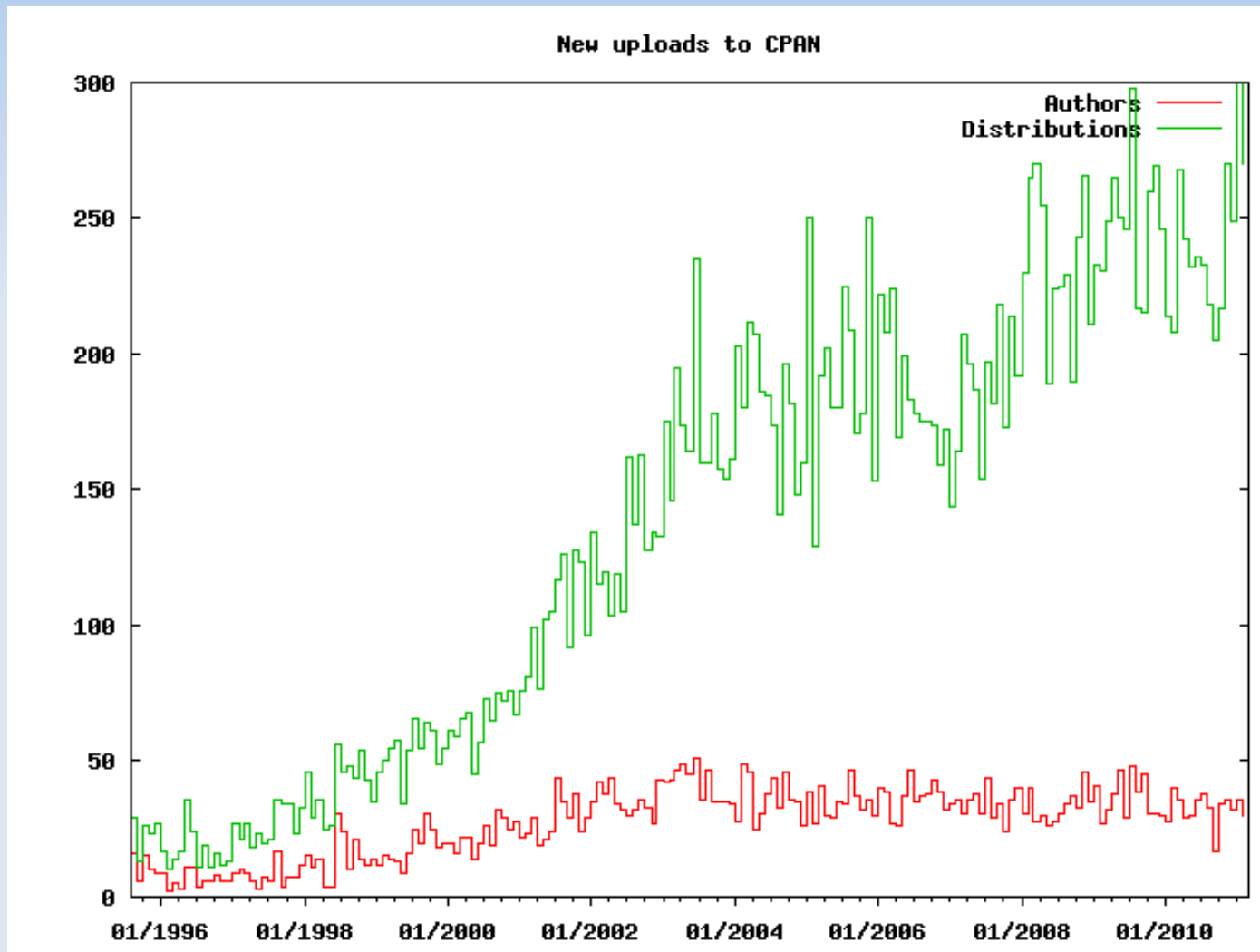
CPAN

- DAS zentrale Archiv für Perlmodule
- inspiriert von CTAN
- Konvention
 - Code, Tests, Binaries, Metadaten, Installation, ...
- Infrastruktur
 - Installer, Smokers, Bug reports, Reviews, ...

CPAN Entwicklung



CPAN Entwicklung



Testing-Kultur

- Testing the `***t` out of software since 1988
- beim Installieren von Modulen automatisch ausgeführt
- CPAN-Testers Infrastruktur
 - erlaubt Tests auf "fremden" Plattformen

TAP

- Test Anything Protocol
- einfache, mächtige, maschinenlesbare Testergebnisse

1..3

ok 1 - erster Subtest Zusammenfassung

not ok 2 - zweiter Subtest Zusammenfassung

ok 3 - Test 3 # SKIP Begründung

ok 4 - Test 4 # TODO Kommentar

not ok 5 - Testbeschreibung

Fehlerbeschreibung: In YAML-Form

...

Testmodule

- simpel - Test::Simple, Test::More
 - is, isnt, like, ..
- fortgeschritten z.B. Test::Deep
 - `cmp_deeply(\@result, [all, any, bag, subhashof, ..])`
- eigene Tests mit Test::Builder
- Defaulttest-Bundle - Test::Kwalitee

Testunterstützung

- Unterschiede erkennen - `Test::Differences`
- Testcoverage - `Devel::Cover`
- Funktionen für Test überschreiben (mocking)
 - `Test::MockModule` (`MockClass`, ...)
- Datenbankschema + Inhalt als YAML Fixtures
 - `Test::Fixture::*`



- Minimalistisch
 - Ueberlaesst das meiste dem Programmierer
 - Sinnvoll in 1994
- Common patters + Perls Flexibilitaet
 - OO Module Oekosysteme
 - Moose, Mouse, Class::Accessor, Object::Tiny, Role::Tiny

Moose

- Postmodern Object System for Perl 5
- CLOS, Smalltalk, ...
- Deklarative Syntax
- Roles
- Type System
- Introspection & Metaprogramming
- MooseX::*

MooseX::*

- Patterns, Helpers, Opinions
 - Singleton, ClassAttribute, Types, Role::Parameterized, FollowPBP, SemiAffordanceAccessor, ...
- Validation
 - StrictConstructor, Params::Validate, Method::Signatures
- Integration
 - Getopt, App::Cmd, Daemonize, POE, NonMoose

MooseX::*

- Syntax Sugar
 - MooseX::Declare
 - MooseX::MultiMethods

```
class BankAccount {
  has balance => (
    is      => 'rw',
    isa     => 'Num',
    default => 0,
  );

  method deposit (Num $amount) {
    $self->balance( $self->balance + $amount );
  }

  method withdraw (Num $amount) {
    my $current_balance = $self->balance;
    confess 'Account overdrawn'
      if $current_balance < $amount;

    $self->balance( $current_balance - $amount );
  }
}
```

DBIx::Class

- erweiterbarer OR-Mapper
- SQL::Abstract
- unterstützt kompletten Lebenszyklus
 - Schema definieren
 - Deployment in beliebige DB
 - Nutzung
 - Wechsel DB
- **jeder** Aspekt des DB-Handlings erweiterbar

DBIx::Class::*

- DBIx::Class::Schema::Loader
- DBIx::Class::Deploymenthandler
- DBIx::Class::Candy
- DBIx::Class::DateTime
- DBIx::Class::Tree*


```
package MyApp::Schema::Result::Artist;

use DBIx::Class::Candy;

table 'artists';

column id => {
    data_type          => 'int',
    is_auto_increment => 1,
};

column name => {
    data_type    => 'varchar',
    size        => 25,
    is_nullable => 1,
};

primary_key 'id';

has_many albums => 'A::Schema::Result::Album',
    { 'foreign.artist_id' => 'self.id' };

```

Perl 6

- z.Z. Experimentalstadium
- wird Perl5 nicht ersetzen
 - nebenher von Perl5/Perl6 ähnlich C/C++
- viele Ideen nach Perl5 rückportiert/beeinflusst
 - Moose
 - Perl6::Junctions
- Compiler Rakudo
 - Rakudo Star veröffentlicht

Community

- Perlmonks <http://perlmonks.org>
 - Forum für Fragen
 - 12/1999, älteste Programmierercommunitywebseite
- <http://perl.org>
 - Zentrale Stelle für Doku, Code, Mailinglisten, ...
- IRC <irc://irc.perl.org>
 - Channels für Hilfe und Subprojekte
- Events
 - Hackathon, YAPC

Deutsche Community

- #perlde auf irc.perl.org
- <http://perl-community.de>
- German Perl Workshop
- auf den CLT :-)

Referenzen

http://testanything.org/wiki/index.php/TAP_History

<http://rakudo.org/how-to-get-rakudo>

Reguläre Ausdrücke

- Perl regex in anderen Sprachen, trotzdem noch hinter Perl hinterher
- named Backreferenzen
 - `/(?<name>regex)/;`
- Look ahead, look behind (kompletteste Regexp)
 - `(?!negative lookahead)`, `(?=positive lookahead)`,
`(?<!negative lookbehind)`, `(?<=positive lookbehind)`
- `/x` erlaubt Whitespace und Kommentare
- `Regexp::Grammars`