

Quality Hacking in Birmingham

A photograph of a white sign with black text and a small circular arrow icon. The sign is flanked by two vertical yellow bars. The text on the sign reads: "QA HAKAKTHON", "2009", and "TEA & COFFEE".

QA HAKAKTHON
2009
TEA & COFFEE

David Golden
Perl Seminar NY
May 2009

Second annual Perl QA Hackathon

- March 28-30, 2009
Birmingham, UK
- Brought together teams for 3 days of face-to-face coding
- Targets specific, realistic projects for a long weekend
- Organized by Birmingham.pm
- Full/partial funding for 10+ people
- Sponsors
 - Birmingham.pm, Dijkmat BV, Digital Craftsmen, \$foo Magazin, Nestoria, booking.com, PROCURA B.V.



Timeline

- August 2008: Birmingham.pm announces intention to host the second QA hackathon
- December 2008: Specific dates announced & website launched (<http://qa-hackathon.org/>)
- January 2009: Sponsored invitations sent to specific developers
- February 2009: Venue announced
- Continuous: recruitment of sponsors and attendees
- March 2009: Hackathon



Why a hackathon?

- Code
 - Time-boxed, team motivation
 - Faster iteration of radical design exploration
 - A sketch is worth 1000 words in email/IRC
 - Cross-fertilization of ideas between teams
- Social
 - Get to meet people that you only know online
 - Feature requests carry more weight in person
 - Arguments less toxic in person
 - Consensus around 'best practices'
- Bonus
 - Learn new tools or techniques
 - Explore a new city

A day in the life at the hackathon

- Breakfast (at hotel)
- Morning summary meeting – goals or accomplishments
- Hack, hack, hack
- Lunch (pre-arranged)
- Hack, hack, hack
- Dinner (pre-arranged)
- Bar (pre-arranged)
- Return to hotel and (maybe) hack until bed



Projects this year

- Config::Perl::V
- configure_requires signoff for 5.10.1
- CPANPLUS + CPANPLUS::Dist::Build
- CPAN Testers + Metabase
- CPXXXAN
- MYMETA.yml
- Nested TAP
- Perldoc::Server
- TAP IETF Draft Standard
- Test::Class::Sugar
- Test::Builder2
- Test::Regexp
- Smolder
- Superfast CPAN mirroring



CPAN-ish stuff

- META.yml 'configure_requires' support confirmed
 `configure_requires:`
 `ExtUtils::MakeMaker: 6.50`
- CPANPLUS + CPANPLUS::Dist::Build
 - Synchronized and prepped for 5.10.1 core
 - Fixes long-standing problems of Module::Build upgrading Module::Build
- CPXXXAN or CP5.8.8AN
 - BackPAN with only things that PASS on 5.X.Y
 - <http://cp5.6.2an.barnyard.co.uk/>
 - <http://cp5.8.8an.barnyard.co.uk/>
- Superfast CPAN mirroring direct from PAUSE (~ 30 seconds)
 - File::Rsync::Mirror::Recent
 - Working towards CPAN 'strata' (like NTP)

Testing tools

- Test::Regexp

```
use Test::Regexp 'no_plan';
```

```
match    subject    => "Foo",  
        pattern    => qr /\w+/;
```

```
match  
  subject    => "Foo bar",  
  keep_pattern => qr /(?<first_word>\w+)\s+(\w+)/,  
  captures   => [[first_word => 'Foo'], ['bar']];
```

- Smolder

- Web-based continuous integration smoker server
- Now installable from CPAN
- <http://www.slideshare.net/mpeters/smolder-introduction>

Experimental nested TAP for Test::More

```
plan tests => 2;  
  ok 1, 'some test';  
  subtest "this is a  
description" => sub {  
    plan tests => 2;  
    ok 1, 'uno';  
    ok 1, 'dos';  
  };
```

Comes out as:

```
1..2  
ok 1 - 'some test'  
  1..2  
  ok 1 - uno  
  ok 2 - dos  
ok 2 - [subtest] this is  
a description
```



Experimental testing tools

- Test::Builder2
 - Abstract recording tests from outputting TAP
 - Attach diagnostics to tests

```
ok($test)->diag($stuff);
```

- Test::Class::Sugar (available on CPAN)

```
use Test::Class::Sugar;
testclass exercises Person {
    startup >> 1 { use_ok $test->subject; }
    test autonaming { is ref($test), 'Test::Person'; }
    test multiple assertions >> 2 {
        is ref($test), 'Test::Person';
        is $test->current_method, 'test_multiple_assertions';
    }
}
Test::Class->runtests;
```

Perldoc::Server

- Like perldoc.perl.org but local on **your** computer

Perldoc::Server - Mozilla Firefox

File Edit View History Delicious Tools Help

http://localhost:7375/

Perldoc::Server on NYC-L3AM346

Manual: [Overview](#) [Tutorials](#) [FAQs](#) [History / Changes](#) [Licence](#)

Modules: [Pragmas](#) [Installed modules](#)

Reference: [Language](#) [Functions](#) [Operators](#) [Special variables](#) [Internals](#)

-- Perl 5.10.0 documentation --
[Home](#) > [About this site](#)

Perldoc::Server - local Perl documentation

This site contains local Perl documentation for **C:\strawberry\perl\bin\perl.exe**

Search path:

- C:/strawberry/perl/lib
- C:/strawberry/perl/site/lib

Powered by Perldoc::Server version 0.05

My project: CPAN Testers + Metabase

- CPAN Testers 2.0 – structured reports transmitted via web to a scalable repository
- Metabase – backend repository for CT 2.0
- Team
 - Me – Metabase fact definition for CT Reports
 - Ricardo – Metabase internals
 - Leon – Database backend for Metabase storage
 - Rich – Test::Reporter::Transport::Metabase
 - Barbie – Metabase search API requirements
 - Andreas and Tux – Config::Perl::V
 - Jos and Chris – CT Client API design sketch
- POC on my flight home: Injected saved CPAN Testers reports into a local metabase

How to make the most of only 3 days

- Focus on small things you can get done
- Measure yourself with working code
- Optimize your face-to-face time not your code
- Skimp on testing if needed to get spike solutions done
- Agree on a few simple standards for a team to follow
- Take the time to educate your team and structure tasks
- Review your code and goals before the hackathon
- Use 'git' for source control if you can



Notes for organizers

- Essentials
 - Space to break out into teams
 - Network that can handle a crowd
 - Coffee, drinks, snacks
 - Morning summary sessions
- Good to have
 - All meals/entertainment pre-arranged
 - White boards and/or flip-charts
 - Free/cheap wireless in hotel for after hours
- Nice touches
 - Locked room for laptops during lunch
 - Rides from airport
 - Local transportation guides
- Watch out for
 - Flaky network access



Questions?