

Michael Schwern is one of the top Perl developers in the world today, working on the language itself and maintaining critical portions of the module build system. He has developed SQL applications responsible for hundreds of millions of dollars in financial transactions. He has written, or maintained, such popular Perl modules as `Class::DBI`, `Test::More` and `ExtUtils::MakeMaker`.

Paid Work

Independent Consultant
Elemental Clinic

Summer 2010

- Maintaining a large Perl application that has been through many hands
- Catalyst, Template Toolkit and mod_perl
- Repairing the test suite
- Writing new tests of a web application
- Gather requirements and bug reports from the client

Optimization Consultant
Social Concepts -- [Wordace](#)

Summer 2009

- In two weeks and on short notice...
- Took a PHP application from handling 50 concurrent users to 3000
- Rapidly learned new technologies and techniques
- Applied live code profiling to direct aggressive optimization
- Parallelized a serial process
- Wrote a basic automated test suite to cover critical functionality
- Wrote automation tools to simulate human players

Senior Developer
Open Sourcing

Spring 2009 (*laid off*)

- Learned Drupal module development
- Wrote Drupal tests using the SimpleTest framework
- Instituted stand up meetings to get people communicating
- Wrote automated tests with Selenium
- Worked on the QA process
- Worked on inter- and intra- team and project communications

Grant Recipient
The Perl Foundation -- [Test::Builder2](#)

Fall 2008 - Present

- Grant to rewrite [Test::Builder](#), the foundation upon which hundreds of Perl testing modules are written.
- Allow customizable test outputs: [TAP](#), POSIX, XML or xUnit style.
- Use modern Perl object oriented techniques including **Moose**, traits and object composition.
- Retain **backwards compatibility**.
- Test on various operating systems and versions of Perl.

Grant Recipient
The Perl Foundation -- [y2038 Project](#)

Fall 2008

- Wrote a **grant proposal**, schedule and deliverables.
- Developed a method to work around the **year 2038 bugs** in time.h on 32 bit machines.
- Implemented this method in **ANSI C 89** for maximum compatibility.
- Tested on various operating systems, compilers and environments including various **Unixen, Windows, Cygwin and VMS**.
- Adapted **perl** to use this method for it's own internal date functions.
- Produced a **Perl module**, Time::y2038, to allow earlier versions of Perl access to the y2038 fix.
- **Worked with a grant manager** to keep them updated on progress.
- **Grant completed**. The code is in Perl 5.12.

QA Automation Consultant
RIPE NCC -- whois database group

Spring/Summer 2008

- Working for RIPE NCC on their whois server responsible for the domain registration service for much of Europe, the Middle East and parts of Asia.
- Worked on a large, existing project of largely untested C, shell and Perl.
- Standardized their existing, custom test framework **using TAP** to allow **automated analysis of test results**.
- Wrote simple **TAP framework** and basic **unit tests for C**.
- Developed an API to start, stop, configure and query their whois server as well as interpret and test responses and log entries.
- Developed a system to rapidly **setup and teardown a test environment** for their whois server including fresh databases and log files.
- Developed a series of high level whois **unit test functions** to reduce the effort involved in writing realistic whois tests.
- Developed an **acceptance test framework** for their RPSL parser to enable easy, quick writing of tests for reported bugs.

Senior Programmer Analyst
Rentrak Corp -- [Video On Demand](#)

2005 - 2006

- Worked with over **half a million lines** of **object-oriented Perl** code and **Mason**.
- Developed for a **large PostgreSQL database** (in excess of billions of rows).
- Worked with **large volumes of customer data** (two billion transactions).
- Worked independently and collaboratively using **test-driven development**, pair programming, task/time tracking systems.
- Responsible for all aspects of software development from specification to installation and bug fixing.
- Converted a large **CVS** repository to **SVN** and retrained staff.
- Performed **profiling and optimization** of both SQL and Perl.
- Worked with a **SQL code generation** system.
- Used DBI, PostgreSQL, Perl, SVN, Apache and FastCGI

QA Automation, Refactoring and Agile Methodologies Consultant 2004 - Present
Grant Street Group -- [LienAuction](#) and [TaxSys](#)

In addition to the normal duties of a QA Automation, Refactoring and Agile Methodologies Consultant mentioned below...

- Helped to develop and test LienAuction, the first online tax lien auction.
- Monitored and hot-fixed LienAuction while auctions were ongoing.
- More than \$150 million in transactions.
- No misplaced bids.

QA Automation, Refactoring and Agile Methodologies Consultant 2001 - 2005
O'Reilly and Associates, Mitel Networks, XS4ALL

Hired as an independent contractor to:

- Establish a **culture of automated testing** by programmers.
- Work with existing Perl-based web services products.
- Convince management that automated testing is worthwhile.
- **Develop the scope of work and requirements** of the project.
- Quickly learn large, complex code bases and begin cleanup.
- Work with developers to find out their needs.
- Discuss design issues with developers and advise on improvements.
- **Document** large amounts of undocumented code.
- Rapidly develop a **testing harness** and test suite.
- Teach automated testing, **refactoring**, documentation, version control, **release management**, advanced OO and **agile methodologies**.
- Overcome language barriers (developers for whom English is a second language).
- Travel to various locales and countries.
- Use Perl, DBI, MySQL, Apache, CVS, SVN and mod_pe

Non-Commercial Projects

Conference Speaker 1999 - Present

Spoke at various conferences around the world on topics such as advanced Perl, industrial design and software engineering

- [O'Reilly Open Source Conference](#)
- [Yet Another Perl Conference \(YAPC\)](#)
- [YAPC::Europe](#)
- [Scandinavian Conference on Java and Object Orientation \(JAOO\)](#)
- [Nordic Perl Workshop](#)
- [Reed College](#)
- [Pittsburgh Perl Workshop](#)
- [Friprog/Go Open](#)

CPAN

1997 - Present

Author

Wrote over 50 Perl modules including such ground breaking modules as [Class::Accessor](#), [Class::DBI](#) and [Test::More](#).

- Writes **tests** and **documentation** for all.
- Perform **coverage analysis** to ensure good test coverage.
- Maintain **compatibility** with many versions of Perl and all operating systems Perl runs on.
- Separate larger systems into small, maintainable, reusable, independent modules.
- Extract and release modules from larger systems ([ExtUtils::MakeMaker](#) and [Test::Harness](#)).
- Use **version control** for all projects (CVS, SVN, git).
- Accept **bug reports** and **integrate patches** from users.
- Design **APIs** which are easy to use, understand and do not encourage mistakes.
- Manage a **community** of users for each module.

ExtUtils::MakeMaker

2002 - 2010

Maintainer

MakeMaker is the Perl module which builds and installs all other Perl modules.

- Took over maintenance after the project had been neglected for several years.
- Large, old, overly-complex, cross-platform Perl module.
- Added automated tests, improved documentation and user tutorials.
- Improved internal design through refactoring.
- Made decisions about design goals and release dates.
- Accepted and rejected patches and bug reports.
- Coordinated code and design input of dozens of programmers.
- All communication done electronically.

Test::More

2001 - Present

Author

Test::More is the most popular Perl testing module today and ships with Perl itself. It provides a well-documented, easy to use, unambiguous module for programmers to write their own tests for their code.

Also includes Test::Builder, a library to allow programmers to write their own testing functions specific to their needs.

Perl 5 Porters

1997 - 2008

Perl Core Developer

Participating in the development of the Perl language and interpreter.

- Large scale, mission critical project with an international team of developers.
- Personally responsible for maintaining several core libraries
- Wrote many bug, documentation and testing patches.
- All communication done electronically.

Perl 5 Porters

2001 - 2005

Perl Quality Assurance Chair

Took on the responsibility of improving Perl's automated testing suite for the 5.8.0 release.

- Convinced the project manager and key developers to support testing.
- Established a culture of testing.
- Wrote new suite of testing tools (see Test::More below).
- Reorganized and improved existing tests and added new ones.
- Taught and encouraged others about QA.
- Advised project manager on risk management.
- Repaired the test suite for Windows, VMS and commercial UNIX variants.
- Established the perl-qa@perl.org mailing list.
- All communication done electronically.

Class::DBI

1999 - 2001

Author

Developed the Class::DBI object relational mapper to turn using an SQL database into an Object Oriented process. Revolutionized the state of Perl object relational mappers and strongly influenced the design of all new ORDMs.

- Provided a simpler API than contemporary systems.
- Wrote strong documentation for easy setup and use.
- Works on most any SQL database.
- One of the most popular Perl DBI modules.
- Handed off maintenance to a new developer.
- All communication done electronically.