Perl in The Enterprise

Dave Cross
dave@mag-sol.com
Magnum Solutions Ltd
http://mag-sol.com/

What We Will Cover

- What is Enterprise Computing?
- Why Perl is good in the Enterprise
- Why Perl is bad in the Enterprise
- The Perl Foundation
- What is coming with Perl 6

What is Enterprise Computing?

- Lets ask Google
- define:"enterprise computing"
- *“A vague term used generally to refer to networks and other computing needs implemented throughout a large, often widely dispersed corporation. Microsoft and IBM, for example, are enterprises; a small business is not.”*
- www.microsoft.com

Enterprise Computing

- Large scale hardware and software implementations
- Interfaces to other systems
- High availability
- Many developers working simultaneously
- Long lifespan
- High visibility

Why Perl is Good in the Enterprise

- Who uses it?
- Why do they use it?
- Web Application examples

Who Uses Perl?

- Everyone
- No really
- Well, almost everyone
- Even Microsoft shops
- Banks see it as a “competitive advantage”
- People regularly build and deploy applications containing 10,000s lines of Perl code

Why Do They Use It?

- Very High Level
- Dynamic Language
- Ready Written Modules
- Quality Assurance
- Multiple Programming Paradigms

Very High Level Language

- Typically one line of Perl to ten lines of C
- Number of bugs is proportional to lines of code
- Much code in off the shelf modules

Dynamic Language

- Code is data
- Like Python and Ruby
- Unlike Java, C and C#
- Programs that write programs
  - Are the happiest programs of all

Ready Written Modules

- Perl has a culture of sharing code
- Comprehensive Perl Archive Network
- http://search.cpan.org/
- Thousands of pre-built modules
- Standard installation and testing procedures
- Plug and Play Programming

Quality Assurance

- Very powerful testing framework
- Encourages module authors to include test suites with their code
- Lots of work in this area over the last five years
- “Perl Testing: A Developers Notebook”
- CPAN Testers
  - http://testers.cpan.org/
- CPAN Ratings
  - Like Amazon ratings
  - http://cpanratings.perl.org/

Quality Assurance

- Perl QA works on two levels
- Makes it easy for module authors to write test plans
  - Code you bring in from CPAN is well tested
- Makes it easy for your developers to write test plans
  - Smoke testing
  - Test first development

Multiple Programming Paradigms

- Procedural Programming
- Object Oriented Programming
- Functional Programming
  - “Higher Order Perl”
- Other, more esoteric, paradigms too
- Pick whatever is best for your application

Web Application Examples

- Everyone is talking about Ruby on Rails
- Example of an MVC framework
- Many MVC frameworks on the CPAN
- Or build your own from components

MVC Components

- Model
  - Class::DBI (object-relational mapper)
  - Also Tangram, DBIx::Class, Class::Persist
- View
  - Template Toolkit
  - Also Mason, HTML::Template
- Controller
  - Your code goes here
- Complete Frameworks
  - Maypole
  - Catalyst

Why Perl is Bad in the Enterprise

- It’s all a matter of perception
- Seen as a “hackers” language
  - Bad CGI programs
  - Two audiences
    - Enterprise vs Hackers
    - Open Source vs Free Software
- TMTOWTDI
  - There’s More Than One Way To Do It
  - Everyone has their own favourite
  - Coding standards and code reviews
  - “Perl Best Practices”

The Perl Foundation

- http://foundation.perl.org/
- “dedicated to the advancement of the Perl programming language through open discussion, collaboration, design, and code.”
- Co-ordinates the efforts of a number of grass-roots Perl groups
  - Perl Mongers
  - perl.org
  - Perl Monks
  - YAPC

The Perl Foundation

- Holding a number of “Perl Days”
- Meeting large companies to discuss what they need from Perl
- Meeting in London August 2005
- Setting up an “Enterprise Perl Working Group”
  - Recommended modules
  - Pre-built libraries
  - Certification

Perl 6

- "Perl 5 was my rewrite of Perl. I want Perl 6 to be the community's rewrite of Perl and of the community."
  - Larry Wall

Perl 6 History

- 18th July 2000, OSCON
  - Perl 5 Porters meeting
  - Jon Orwant
  - Major change needed
- 19th July 2000, OSCON
  - Larry Wall's "State of the Onion"
  - Announces development of Perl 6
- It's been a long time coming

Perl 6 Design

- Massive RFC Process
- Canvassing ideas from community
- 361 RFCs received
- http://dev.perl.org/perl6/rfc/
- Larry (with help from others) merges RFCs with his ideas
- Much discussion on mailing lists
- Designs published as a series of Apocalyses and Exegeses
    - http://dev.perl.org/perl6/

Perl 6 Features

- Many new features
  - explicit strong typing
  - proper parameter lists
  - active metadata on values, variables, subroutines, and types
  - declarative classes with strong encapsulation
  - full OO exception handling
  - support for the concurrent use of multiple versions of a module
  - extensive introspection facilities
  - LL and LR grammars (including a built-in grammar for Perl 6 itself)

Perl 6 Features

- More new features
  - subroutine overloading
  - multiple dispatch
  - named arguments
  - a built-in switch statement
  - hierarchical construction and destruction
  - distributive method dispatch
  - method delegation
  - named regexes
  - overlapping and exhaustive regex matches within a string
  - named captures

Perl 6 Features

- Yet more new features
  - parse-tree pruning
  - incremental regex matching against input streams
  - macros (that are implemented in Perl itself)
  - user-definable operators (from the full Unicode set)
  - chained comparisons
  - a universally accessible aliasing mechanism
  - lexical exporting (via a cleaner, declarative syntax)
  - multimorphic equality tests

Perl 6 Features

- A few more new features
  - state variables
  - hypothetical variables
  - hyperoperators (i.e. vector processing)
  - function currying
  - junctions (i.e. superpositional values, subroutines, and types)
  - coroutines

Perl 6 Related Projects

- Parrot
- PONIE
- Pugs

Parrot

- Virtual machine used to efficiently execute bytecode for dynamic languages
- Core of the Perl 6 compiler
- Target platform for Perl 6
- Other languages will be able to target Parrot too
- Use libraries from other Parrot languages
- http://www.parrotcode.org/

PONIE

- Perl On New Internal Engine
- Perl 5 implemented on top of Parrot
- Implementing a real language in Parrot
- A real test that Parrot is up to the job
- http://www.poniecode.org/

Pugs

- Prototype Perl 6 compiler
- Written in Haskell
- Good way to try out Perl 6 now
- Limited but getting better very quickly
- http://pugscode.org/

Conclusion

- Perl is currently very suitable as an Enterprise level language
- Many large companies rely on Perl to drive their business
- Perl 6 will be even better

Questions

- Any questions?
  - Except “when will Perl 6 be out?”