

IBM Open Source Competition 2011

UCD

Brendan Arthurs – Advisory Software Engineer, IBM Dublin Software Lab

What is Open Source

- Open Source Initiative (OSI)
<http://www.opensource.org>
- Open Source Definition
 - Free Redistribution
 - Source Code
 - Derivative Works
 - Integrity of The Author's Source Code
 - No Discrimination
 - No Discrimination Against Fields of Endeavor
 - Distribution of License
 - License Must Not Be Specific to a Product
 - License Must Not Restrict Other Software
 - License Must Be Technology Neutral

The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.

Open Source Development Methodology

- Virtual community of programmers, leveraging the Internet for communication, who create / debug / maintain / evolve a source code base
- OSS projects often self-organizing:
- Someone determines a need and communicates that need to others on the Internet
- If the project generates interest, one or more programmers begin writing code
- Someone takes a leadership role and begins to map out a project road map
- Interested programmers join the project to contribute new code or fine-tune existing code
- A network of participants, linked via the Internet
- Tiered participation levels emerge

Example: The Linux Kernel

- 1991: Linus Torvalds starts to develop a free OS “just like a hobby, won't be big and professional like GNU”
 - Posts details on Usenet, asks for contributions
 - Lots of enthusiasm, since there was no other free unix-like OS available at the time
 - 1994 - Linux 1.0 (1996 - Linux 2.0, 2003 - Linux 2.6)
 - 1999 – IBM zSeries support
 - 2010 – Linux runs on everything from phones to supercomputers
-
- Major commercial interest (Novell, Redhat, Canonical, IBM and many others)
 - Unquantifiable development value



Open Source Development Methodology, Take 2...

- Company recognizes that a core piece of software has potential to grow further than they can take it
- Source code released under Open Source license
- Three possible paths:
 - Anyone can “fork” the code, and customize it for their own use
 - Company takes on a “gatekeeper” role, accepting submissions and releasing new versions
 - Independent foundation set up with community involvement to “own” the main development stream

Example: Eclipse

- Developed by IBM, initially as an IDE
- 2001: Released under an Open Source License
- Huge level of user interest, thousands of contributions integrated into main product.
- 2004: Eclipse Foundation created to manage the Eclipse project

- 2010: Used as the client-side foundation for everything from email clients to stock trading platforms, to city modeling tools



Eclipse Project Aims

- Provide open platform for application development tools
 - ▶ Run on a wide range of operating systems
 - Windows, Linux, AIX, MacOS X, etc.
 - GUI and non-GUI components
- Language-neutral
 - ▶ Able to handle different content types
 - ▶ Java, HTML, C/C++, JSP, EJB, XML, GIF, etc.
- Facilitate seamless tool integration
 - ▶ At UI and lower level
 - ▶ Add new tools easily
- Attract community of tool developers
 - ▶ Including independent software vendors (ISVs)
 - ▶ Capitalize on popularity of Java for writing tools

Why develop plugins that extend Eclipse ?

- Infrastructure
 - You get a workbench created for free – just add what you need
 - No need to create an environment from scratch
 - Build on Eclipse Modelling Framework (EMF) for model support
 - Build on Graphical Editing Framework (GEF) for graphics support
- Interoperability
 - Use any of the existing tools in your ‘end-product’
 - Java Editor, outliner . . .
 - Explorer, Text editor, search . . . etc.
- *Examples*
 - Java (or any language) code analysis/transformation plugins
 - Any UML related tooling (using EMF)
 - Any Java graphical projects (using GEF)

IBM and Open Source

- IBM uses open source in its products
 - Building products on top of open source projects e.g. Rational Application Developer
 - Apache webserver to support and bundle with its WebSphere suite
 - Blue Gene – Linux used on all IO Nodes

- IBM promotes open source development
 - Externally
 - AlphaWorks: alphaworks.ibm.com
 - DeveloperWorks: www.ibm.com/developerworks/opensource
 - Internally
 - Using open source intranet site to promote software componentization and reuse

IBM Open Source Software Competition

- The competition is based on fourth year CS projects in each university
- An IBM Mentor who is a senior software architect from Dublin Software Lab has been assigned to each university
- To be eligible for participation there must be a minimum of five qualifying projects
- At the end of the academic year, the mentor will evaluate the projects based on the level of Open Source content and on the overall project results and will select a short list of projects which will in turn be evaluated by a panel at the Software Lab
- Individual projects, not joint projects, will be evaluated
- The student with the best project will win a high spec IBM Think Pad

Opensource Projects to Consider

- Frameworks / Languages

- Ruby / Rails
- PHP / CakePHP
- Perl
- Python

- Cloud Computing

- Linux Kernel
- OpenMPI
- Xen hypervisor

- Web 2.0

- LAMP
- Dojo
- Firefox + XUL Runner
- Websphere Application Server Community Edition

- Client Environments & Libraries

- Eclipse / OSGI
- Apache libraries (for everything!)

Questions



Contact: Brendan Arthurs - brendan.arthurs@ie.ibm.com