

My Favorite OSS Projects

Glenn Engstrand

October 28, 2007

Abstract

Open Source Software has become a very important and influential movement in the world of Information Technology. Software developers can gain a lot of ability by embracing OSS tools. This article lists my favorite OSS projects, it also explains why and how I feel that these projects are so relevant in the areas of requirements capture, designing, and coding software. I listed some fun projects here. Finally, I mention some projects that are breeder projects for the entire software ecosystem in that they lay a foundation for other projects to grow. Unless otherwise stated, you should assume that these tools run on both MS-Windows and Linux.

1 Why OSS Is So Important

It's one word actually, love. OSS is so relevant to software developers because it is written by software developers for software developers. The writers have a very good understanding of the target market. Furthermore, if you find a feature missing or you just don't like the way it works, then change it yourself. Sometimes that is easier said than done but it is always impossible in closed source software.

There's another word too, money. OSS is free and the upgrades are also free, forever. If you were to purchase proprietary versions of what is listed here, then it would cost you tens of thousands of dollars. Don't forget that upgrading every two years will also cost you thousands of dollars. OSS lowers the economic barriers to entry into software development for both IT and ISV shops. Using OSS gives developers the tools they need to do the job better and cheaper.

OSS is not about bloatware. In proprietary systems, the vendor has to come up with all kinds of reasons why you need to upgrade to the next release because that is how they make payroll. The affect of years of that kind of motivation is an overly complicated and resource intensive application that tries to do everything only half way. OSS comes from the Linux culture which is all about using the right tool for the job and that each tool should do one thing only and do it really well. As a software engineer, I can relate to that. I don't

mean to imply that OSS stays the same. I am writing this in the fall of 2007 so if you are reading this years later, realize that many of the issues or faults that I mention here are most probably already addressed.

Proprietary software is OK too. I'm not religious or fanatical about OSS. There are plenty of great closed source applications out there. I use plenty of such applications at work every day, no problem. They are good enough but there are also plenty of OSS applications that are just as good or better and are also available at the right price and are written by people with the right intentions. If you are considering purchasing proprietary tools in your software development shop, then you owe it to your company to consider the OSS alternatives.

2 Tools for Requirements Capture

Requirements must be captured because coding should be based on specifications. Otherwise, the project will soon be lost and QA won't know what to test for either. There are many great OSS tools for this.

- Mind mapping is a type of diagramming that focuses on presenting the order or structure of something through a radial hierarchy with pictures, text, and linking between nodes. It is highly effective for analyzing complex data. **Freemind** is a mind map diagramming tool that is great at drawing such diagrams easily. It gets out of the way and lets the creative process flow [1].
- A picture is worth a thousand words. There are two types of such pictures during requirements capture, rough, or low fidelity, mockups and finished, or high fidelity, mockups. The rough mockup is drawn quickly and is good for providing immediate feedback to ideas about the GUI. The vector illustration drawing software, **Inkscape**, is great for drawing these kind of mockups [2].
- The other type of picture is called the finished, or high fidelity mockup. This mockup is more accurately detailed so it looks very much like a screen shot of the finished product. The best type of tool for this is the raster image editor and my recommendation for drawing raster images is the GNU Image Manipulation Program or **GIMP** [3].
- Words are still needed, however. How do we put it all together and describe what needs to be written without word processing? There are two approaches to this, a document publishing system or an office productivity suite. For the former, there is **Latex** [4]. For the later, there is **Open Office** [5]. Many claim that Latex is too hard to use because you have to code all these archaic commands into your document. There is a pretty cool IDE for Latex called **TexNicCenter** that generates those codes for

you so all you have to do is click the appropriate toolbar button or menu much like MS-Word. TexNicCenter works only on MS-Windows, however [6]. Many claim that Open Office is too slow and lacks important features. That has not been my experience.

- It is also helpful to be able to manage the timeline and delivery schedule of the software. This is best done with PERT charts and Gantt charts. **GanttProject** is a nice project management tool for creating tasks, resources, PERT charts, and Gantt charts. It is a Java swing application that has come a long way in the many years that I have used it [7]. In addition to the price advantage, I would pick this over its proprietary contender, MS-Project, because MS-Project has the annoying feature that tasks in the past are no longer easily editable. Everything in GanttProject is easily editable.
- All of these OSS tools are used to author files that can be exported in what is known as PDF or Portable Document Format. One such open source viewer for PDF files is **GSView** [8]. The Adobe reader is not OSS but it is also free; however, I prefer the OSS software as it is also free of advertising whereas the Adobe reader is not. There are also fewer security problems with the OSS version and it takes up less system resources than the Adobe version.
- If you are stuck with MS-Word (which cannot create PDFs), you can still export to PDF using **PDFCreator** which is a printer driver that saves whatever is printed to a PDF file. This works for anything that is printable and not just MS-Word [9].

3 Design Tools

Design must precede coding for the same reason that blueprints are needed before building construction can begin. Without some sort of design, what you code may very well fall apart before you are finished. The linear parts of design can be captured the same way as what is discussed in the requirements area of this article using either a document publishing system or an office productivity suite. Diagrams are a very powerful way of capturing the non-linear parts. Here are some of my favorite OSS diagramming tools.

- Unified Modeling Language, or **UML**, is a notation system for capturing the design of software. To this affect, it introduces nine different diagram types [10]. I have used a lot of different UML diagramming tools over the years, both proprietary and OSS. By far, my favorite is **Umlet**. Most of the other UML diagramming tools try to get in the way of the design process, either trying to have you enter in all of the information for the full UML specification or trying to generate code from the design or trying to validate the design. Umlet does none of that. It is all about letting you

very easily and effectively draw the diagram that you are trying to draw [11].

- An **ERD**, or Entity Relationship Diagram, is used to capture and design the data model for a software project. The data model is usually implemented using a relational database [12]. I have used a lot of different ERD diagramming tools over the years. Most of them are really expensive. My favorite tool for drawing these types of diagrams is called **DDT** which is short for Database Design Tool. I like this tool because it is a pure diagramming tool. The other tools also get into having you completely specify the database design. There is usually a differentiation between logical design and physical design. They usually also generate create SQL statements for various relational database vendor products. DDT is just about quickly and easily drawing ERD without getting in the way of the creative process. This tool is only for MS-Windows [13].
- Another pure diagramming tool is **dia**. It can also draw UML and ERD but it is not limited to that. Other types of diagrams that it can draw are flowcharts, networking, and electric circuits [14].
- It is sometimes useful when designing software to capture ontologies. You may be designing a Resource Description Framework Schema, or RDFS, for the purposes of developing a wire protocol for inter-application communication. Or you may be attempting to create a conceptual class diagram for your main entity abstractions that you would like to convert to a data dictionary. Either way, **Protege** is the tool for you. It is not a pure diagramming tool. You are supposed to be able to create diagrams with it but I have rarely seen it work that way. Normally, I just fill out text fields, check boxes, and select lists on GUI screens. It is also very much **Semantic Web** oriented using such languaging as slots, roles, domains, constraints, and facets [15]. In spite of all that, I find myself using it over and over, usually during the high-level design phase. Protege can also be used to generate Java code from a defined ontology [16].

4 Coding

Here we get to the actual construction phase of software development. This is really the only deliverable that customers care about. It is also the part where OSS really shines. The first section is about some tools that developers use to write code. The other sections detail various application programming stacks that developers can write to in order to accelerate the speed of development because they have less that they need to write.

4.1 Cool Tools

What are my favorite tools for editing source code? Here is the OSS IDE, or Integrated Development Environment, that I recommend.

- There are many IDEs, both proprietary and OSS, that I have used over the years. The one that prevails the best over time is **Eclipse** by IBM. It's got all the cool features such as color syntax highlighting, statement completion, easy compile and debug, and automated refactoring. Some complain that it is slow with large projects; however, I find it to be no slower than its proprietary counterpart which is VS.NET by Microsoft. The best feature is its plug-in architecture and there is a very large and active community of plug-in writers out there [17]. Let's discuss two of them now.
- The default Eclipse experience is all about coding in Java but maybe that is not your preference. There is a great plug-in for python developers called **pydev** [18]. This is what I use on Linux machines but my favorite python IDE runs only on MS-Windows. It is called **pythonwin**. It does debugging and code navigation better than pydev [19].
- The other Eclipse plug-in that I want to mention is **RadRails**. This gives you a pretty quick and easy way to develop Ruby on Rails web applications [20].
- An IDE by itself is OK for a single developer but today's modern software projects require potentially large teams of developers who are most probably geographically disbursed. How do you keep track of all the changes or just try to keep developers from overwriting each others code? That is where a source code concurrency and control repository comes in. With such a system, changes to any of the source code files are never lost and are also kept in a way such that the history of changes is never lost. I wouldn't even think of doing modern development without one. I have used plenty of such systems, both proprietary and OSS, over the years and my favorite is called **Subversion**. It is well suited for highly geographically dispersed groups of developers. It supports multiple concurrent writers with a merge style commit. It also has a way for you to write your own hooks into it [21]. Because of that, it integrates very nicely with SDLC tools such as **GForge** [22] or continuous integration tools such as **CruiseControl** [23].
- Eclipse integrates very nicely with Subversion but if you are not using Eclipse and you don't want to use the command line, there is **Tortoise**. It is an MS-Windows only Subversion client that runs as a windows shell extension so all you need to do is right click on the top level source code folder in windows explorer and select the Commit menu [24].

4.2 Apache Foundation

This group has donated so much to J2EE based web application development that it is safe to say that without them, J2EE would have failed. All of the projects that I am about to mention are OSS and are great for excellerating development by adding important features to any sophisticated J2EE web application.

- The namesake for Apache, the flagship application and what they are most known for is their web server. As of the time of this writing, it continues to be the most popular web server on the Internet [25]. It is a very advanced, pluggable web server. It integrates with subversion, tomcat, and plone. It supports SSL, WebDAV, caching, smart filtering, IPv6, you name it [26]. There is also a thriving plug in community for Apache. Here are some examples. **Lingerd** is an accelerator for Apache that works by taking over the job of lingering on a connection before closing it [27]. **Pound** is reverse proxy and load balancer for Apache [28].
- A J2EE web application has to run within an application container. **Tomcat** is, by far, the most popular full featured, fully J2EE compliant container. Most J2EE shops use Tomcat for development and many shops use it for production too. Recent versions have exhibited very respectable scalability and performance behavior [29].
- For J2EE shops wishing to architect an MVC application with an IoC framework, take a look at Struts. It features an extensive tag library, forms validation, and support for i18n [30].
- If you have a web application that has to do more with email than just sending it, then I recommend you integrate with **James** which is an advanced, full featured email server that is easy to integrate with. Your app can be notified when email arrives at a certain account. It also features sophisticated filtering and relay features and authenticated SMTP [31].
- If you wish to allow your GUI to be customized with a template generation system, then consider embedding **Velocity** in your application. That is a very effective way to allow individual shops to customize an installation's GUI without having to junk up the main code base [32].
- If you need XML parsing, then **Xerces** is the way to go. It is a W3C DOM compliant XML parser for both Java and C++ [33].
- Sometimes, you want to generate the GUI using XSLT where the template for the GUI is an XSL file and the data is an XML document. The best way to handle this is with **Xalan** which is an extension to Xerces [34].
- If you need to algorithmically generate PDF documents from data in your database, then you will most probably find the best way to do that is by combining Xalan with **FOP** which takes an XML file and creates a PDF from it [35].
- If there is something you want to do that you can't find in the J2EE or the other Apache libraries, then it's worth a look into the **Jakarta Commons** project which has a very large library of routines to balance out anything that might be missing elsewhere [36].

- An easy way to build complex Java applications is to use a humble tool called **Ant**. Many IDEs use Ant and it is incredibly easy to write and maintain an Ant build file [37].

4.3 Java

There is more to OSS J2EE than Apache. In the past, there have been numerous complaints that Java itself was not OSS. What was the point of building an OSS application on top of a proprietary foundation? Sun Microsystems has begun releasing Java as OSS [38]. Here are some more OSS Java offerings that are neither a part of Apache nor Sun.

- The increased popularity of eXtreme Programming, also known as XP, has heightened the awareness of the importance of automated unit testing. **JUnit** is a java based testing framework for this style of testing [39].
- A very popular and powerful framework is Spring which features lightweight IoC/DI, ORM, and support for AOP [40]. CruiseControl, Spring and Junit combined do a great job of advancing the continuous integration features of XP.
- If you ever need to parse anything other than XML, you should take a serious look at **Antlr** which is a LALR(k) parser for any language that can be specified in BNF. This is in the Java section but Antlr can also work in python, C++, and .NET environments [41].
- If you ever need an easy OLAP capability for cubes that are not too big nor complex, then consider giving **Mondrian** a try. It has a web based analytical browser that is pretty cool. This is not a contender with the proprietary offerings of Microsoft and Hyperion but if you just need OLAP as a checkmark on the feature list of your app, then consider it [42].

4.4 Dynamic Scripting

Open source has always been at the vanguard of dynamic scripting. Programming languages can be either static or dynamic. In a static programming language, there is both a compile time and a run time. Variables are assigned a type at compile time and remain that type throughout the entire run time period. In a dynamic language, there is only a run time. Variables can change type during run time. Classes can change their interfaces during run time. Even meta-programming is possible in some dynamic languages. The obvious advantage to this is the incredible flexibility that these kinds of languages provide. The disadvantage to this approach is that dynamic scripting languages tend to run slower than static ones. Also, the flexibility gives the developer a lot more opportunity to introduce bugs or defects into the system. Open source developers tend to be OK with this because the culture of open source is one that would choose freedom over safety [43]. Dynamic Scripting Languages, also known as

DSL, have support for data structures built into the language instead of accessible from a library. This usually tends to make heavy algorithmic coding easier to read. Here are some of the more popular OSS DSL.

- **Perl** is perhaps the first, original OSS DSL. Originally, it was used by system administrators to perform routine, repetitive maintenance tasks. Apache can now integrate with the perl interpreter to make huge web applications written entirely in perl. Although it preceeded object oriented programming, perl has been retrofitted to be somewhat object oriented. Perl style object oriented programming is a bit cumbersome in my humble opinion. Perl has both been criticised and praised for its extremely terse and archaic syntax [44]. There are two big advantages to developing in perl these days. Almost every ISP allows you to develop and run perl programs cheaply and there exists an incredibly huge and stable library of perl routines that allow you to add just about any feature to your web application that you can imagine [45].
- **Python** is an object oriented DSL with great support for string generation and parsing. Indentation of the code is what controls scoping which, I've been told, is a good thing. Adding specially named methods (e.g. `__getattr__`) to each class is how you can dynamically control its interface. Python isn't truly object oriented because you don't really have control over the visibility of members in an interface. It's all public all the time [46].
- **Zope** is a python web application container with a very flexible tag based, pluggable templating system and a pure object oriented database called ZODB [47].
- **Plone** is a highly customizable content management system and rudimentary portal site built on zope. Another very cool feature is a way to quickly build custom document types, called archetypes, and a configurable way to choose between storing the underlying data in the ZODB or a relational database [48].
- **Ruby** is the most recent newcomer to the OSS DSL world. It is an uber pure object oriented DSL with support for closures and meta-programming. It doesn't provide support for multiple inheritance but it does provide support for mixins which is the one part of multiple inheritance that every developer wants. Support for regular expressions is built into the language itself [49].
- **Rails** is a web application container and framework for Ruby using the ActiveRecord design pattern. It is also great for RAD style prototyping because it can generate a lot of the scaffolding code based on a database schema but I don't recommend going that approach when writing the real application [50].

- I've got to mention **Java Script** when it comes to DSL because that is clearly what most developers code in when implementing dynamic GUI behavior in a web application. Java Script is faux object oriented at best because, under the covers, there is no difference between a class and a hash table. Because of this, embarrassing things happen such as the semantics of the **self** keyword change depending on how a method in a class is called. That's not my idea of a good time.
- My favorite OSS Java Script library is **Prototype** because it mitigates all of Java Script's little shortcomings. Prototype has good support for Asynchronous Javascript and XML, also known as AJAX, which is compelling because this is the technology that frees the surfer from having to suffer the flicker of the screen refresh when a round trip is made between the web browser and the web server [51].
- Java Script is used mostly in the web browser to dynamically manipulate an HTML DOM. There are now a lot of high-end GUI widget toolkit libraries to jump start your development in this area. This is a very good thing because the overly forgiving nature of Java Script makes it notoriously hard to debug. **Scriptaculous** is a library, built on Prototype, that is known for its flashy affects and sweet eye candy [52]. The **Yahoo User Interface**, or YUI, is another Java Script library widget toolset that is more focused on writing serious business apps [53].
- The **Google Web Toolkit**, or GWT, is yet another popular Java Script library. It's different in that you create an object hierarchy in Java and it uses that to generate the Java Script. The developer really doesn't code in Java Script at all using this library [54].

4.5 Databases

The relational database is both the workhorse and the crown jewels of any enterprise application. OSS offers two for you to choose from.

- **MySql** is most probably the more popular one. Originally, it was a very very bare bones RDBMS that was fast and was most typically used for simple CRUD (Create, Read, Update, Delete) style access. The modern version comes with ANSI support, stored procedures, and multiple backends including a transactionally aware ACID compliant one [55].
- There was a time when **PostGreSql** was known as the slower OSS RDBMS but time has pretty much equalized that metric. Many would disagree with me and your results may vary. They both have ACID compliance, stored procedures, good performance. If you are into object relational mappings within the data tier, then PostGreSql might be the better choice [56].

Which one should you choose? Well, if you are running a web application through an ISP where the application is hosted on their machines, then your choice will most likely be MySQL. Oracle keeps trying to kill MySQL so that would lead me to believe that MySQL is the better choice [57].

4.6 The Other Java

Sun Microsystems and Microsoft have been aggressively competing for developer mindshare for a long time. What does this have to do with OSS? Microsoft has been studying its enemy and has come up with what it believes is the Java killer. They call it .NET and provide two languages for it, VB.NET and C#. It has much of the same features as Java including an extensive class library, a memory managed virtual machine, generic programming, and autoboxing. Many of the technologies mentioned during the Java section also have .NET versions (e.g. NUnit and Spring.NET). It seems that Novell now owns, maintains, and releases an OSS version of .NET called **Mono** [58]. Many claim this to be a poison pill for submarine patents. Even Microsoft itself is starting to pretend that it may open source .NET too [59] but many claim this also to be a poison pill that, when taken by a developer, forever legally taints that developer from OSS .NET coding. Those who are against .NET in OSS believe that Microsoft will start issuing C&D letters followed by patent infringement suits once there is enough traction in .NET for OSS. I have no legal training so I cannot make any viable claims regarding the legal status of .NET in OSS. I can say this. I have worked extensively in both Java and .NET and found both platforms to be equally meritorious.

5 Fun Projects

Programmers have to let off some steam too, you know. Which is why I have listed some fun projects that I like. I am, by no means, the great expert in this area so please don't take any offense if I missed your project.

5.1 Arts

Coding is one way to artistically express yourself but there are other ways too. This section is devoted to some tools that allow you to compose and create other than software.

- The **Text Adventure Development System**, or TADS, is a player and IDE that allows you to write Interactive Fiction literature. I prefer TADS over its rival Inform because TADS is more developer friendly whereas Inform is more author friendly [60].
- **Lilypond** is a typesetting system for printing written musical compositions. The commands are arcane but highly advanced and functional. Lilypond can render very professional looking scores [61].

- Of all the OSS 3D rendering and modeling systems out there, **Blender** is the most functional but it is also the hardest to learn how to use. There are a lot of key commands that have no GUI representation and no easy mnemonic device [62].
- Audacity is the best of class audio recording and editing software [63].

5.2 Games

Sometimes destroying can be more fun than creating. If running around and shooting up people sounds more appealing than writing or drawing, then it's time to get your game on. Here is my current crop of favorites in the world of OSS FPS or First Person Shooter games. There's nothing more satisfying than a half hour of gaming as cathartic release after a stressful day of coding on a project with shifting requirements, expanding scope, and too many meetings on improving productivity.

- **Nexuiz** is a great frag-fest networked FPS. I particularly like the colors of the sets, the lighting and mood, and the weapons [64].
- **Sauerbraten** is a pretty mindlessly entertaining single player arena style FPS. It supports network play too [65].
- **Tremulous** is an innovative asymmetric team based network FPS. If you join the human team, then you get range weapons and *level up* (i.e. get more powerful) by purchasing more equipment. If you join the alien team, then you have melee capability only but you're faster than humans. You level up by evolving [66].

6 Influential Projects

Apache isn't the only foundation who has contributed much to OSS. There are two other organizations whose influence have been critical to the open source movement, GNU and Mozilla.

6.1 Operating Systems

No article on OSS is complete without talking about Linux. Sometimes called GNU Linux, this is the original open source operating system started (and still managed to this day) by Linus Torvalds. This is ground zero for the open source movement [67]. There is a very popular OSS programming stack for building web applications called LAMP which is short for Linux, Apache, MySQL, and Perl. We have already talked about the last three. There is a long history of *distros* (short for distribution) of Linux [68]. Here are some of my favorites.

- **Redhat** is positioned as the best of breed Linux distro for server applications. It comes in two flavors, Fedora and RHEL or Red Hat Enterprise Linux. Fedora is more developer friendly and has lots of new and potentially unstable releases of applications. RHEL is a stable O.S. intended for production server work [69]. Redhat got a big kick start by IBM who endorses them as a server O.S. [70].
- Linux is open source and, therefore, free but you must pay money for RHEL. How is that possible? The Redhat corporation can sell RHEL for money because the desktop experience is branded with their trademarked and copyrighted red hat logo. **CentOS** is the free version of RHEL without the branding [71].
- **Debian** is a great and venerable incubator for desktop linux distros. Many Linux distros are based on debian including the following two [72].
- **Ubuntu** is the current media darling for consumer desktop linux and a direct competitor to Vista. It is very easy to use and set up and has a GUI for just about everything you can imagine [73].
- **Knoppix** is a ‘must have’ tool for every system administrator who needs to retrieve files from a dead MS-Windows machine. You can boot straight from the CD. It will run on just about any type of PC. Once booted, you have a sweet desktop including Open Office, Firefox, and the ability to transfer files to any networked Windows file system [74].
- Though not technically an O.S. or even open source, **Cygwin** provides a linux style command line for when you have to work on MS-Windows [75].

6.2 Mozilla

The Mozilla Foundation is deeply committed to making sure that users will always be able to use the Internet with open source software.

1. In the traditional web application, the web server fires off some SQL to the database server and uses the results to generate some HTML and Java Script to send to the client machine. The windows application that interprets the HTML and Java Script and renders a GUI from them is called the web browser. **Firefox** is the favorite web browser for OSS developers because it is OSS. because it is the most compliant of web standards, and because of its plug in architecture and active developer community [76].
2. **Thunderbird** is an email client. It has the usual great email features such as POP3 and IMAP support, encryption support, spam filters, multiple folders, and a syndicated content (e.g. RSS, ATOM) reader [77]. You would think email being such the *killer app* that there would be a lot

of email clients but there isn't. Microsoft has Outlook. Qualcomm has Eudora. There is a competing OSS email client in Novell's **Evolution** but that product is too buggy for my tastes.

3. Mozilla offers **Lightning** as a plug in to add calendar functionality to thunderbird [78].

References

- [1] *Main Page - FreeMind - free mind mapping software*, <http://freemind.sourceforge.net/wiki/index.php/Main_Page>
- [2] *Inkscape, Draw Freely*, <<http://www.inkscape.org>>
- [3] *GIMP - The GNU Image Manipulation Program*, <<http://www.gimp.org>>
- [4] *LaTeX project: LaTeX - A Document Preparation System*, <<http://www.latex-project.org>>
- [5] *OpenOffice.org: Home*, <<http://www.openoffice.org>>
- [6] *proTeXt - MiKTeX-based distribution for Windows*, <<http://www.tug.org/protext>>
- [7] *GanttProject.org - Home*, <<http://ganttproject.biz>>
- [8] *GSview*, <<http://pages.cs.wisc.edu/~ghost/gsview>>
- [9] *pdfforge.org*, <<http://www.pdfforge.org>>
- [10] *Object Management Group - Unified Modeling Language*, <<http://www.uml.org>>
- [11] *UML Tool for Fast UML Diagrams*, <<http://www.umlet.com>>
- [12] *The Entity-Relationship Model - Toward a Unified View of Data*, <<http://csc.lsu.edu/news/erd.pdf>>
- [13] *Homepage of DDT (Database Design Tool)*, <<http://www.visi.com/~jjanssens>>
- [14] *Dia a drawing program*, <<http://www.gnome.org/projects/dia>>
- [15] *OWL Web Ontology Language Overview*, <<http://www.w3.org/TR/owl-features>>
- [16] *The Protege Ontology Editor and Knowledge Acquisition System*, <<http://protege.stanford.edu>>
- [17] *Eclipse - an Open Development Platform*, <<http://www.eclipse.org>>
- [18] *Python Development Environment*, <<http://pydev.sourceforge.net>>

- [19] *Python for Windows*, <<http://www.python.org/download/windows>>
- [20] *Aptana, The Web IDE*, <<http://www.radrails.org>>
- [21] *Open Source Software Engineering Tools*, <<http://subversion.tigris.org>>
- [22] *GForge Collaborative Development Environment*, <<http://gforge.org>>
- [23] *Cruise Control*, <<http://cruisecontrol.sourceforge.net>>
- [24] *The coolest Interface to (Sub)Version Control*, <<http://tortoisesvn.tigris.org>>
- [25] *October 2007 Web Server Survey*, Netcraft, <http://news.netcraft.com/archives/web_server_survey.html>
- [26] *The Apache HTTP Server Project*, <<http://httpd.apache.org>>
- [27] *lingerd - an accelerator daemon for Apache*, <<http://www.iagora.com/about/software/lingerd>>
- [28] *Apsis Gmbh*, <<http://www.apsis.ch/pound>>
- [29] *Apache Tomcat*, <<http://tomcat.apache.org>>
- [30] *Apache Struts*, <<http://struts.apache.org>>
- [31] *James Project*, <<http://james.apache.org>>
- [32] *The Apache Velocity Project*, <<http://velocity.apache.org>>
- [33] *Welcome to Xerces*, <<http://xerces.apache.org>>
- [34] *The Apache Xalan Project*, <<http://xalan.apache.org>>
- [35] *Apache FOP*, <<http://xmlgraphics.apache.org/fop>>
- [36] *The Jakarta Project*, <<http://jakarta.apache.org>>
- [37] *Apache Ant*, <<http://ant.apache.org>>
- [38] *Opening Up: Laurie Tolson on Open Source Strategy for the Java Platform*, <http://java.sun.com/developer/technicalArticles/javaopensource/OS_qa/index.html>
- [39] *JUnit.org Resources for Test Driven Development*, <<http://www.junit.org>>
- [40] *Spring Framework*, <<http://www.springframework.org>>
- [41] *ANTLR Parser Generator*, <<http://www.antlr.org>>
- [42] *Pentaho Analysis Services: Mondrian Project*, <<http://mondrian.pentaho.org>>

- [43] *The Cathedral and the Bazaar*, <<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar>>
- [44] *The Perl Directory*, <<http://www.perl.org>>
- [45] *The Comprehensive Perl Archive Network*, <<http://cpan.perl.org>>
- [46] *Python Programming Language*, <<http://www.python.org>>
- [47] *Welcome to Zope.org the Website for the Zope Community*, <<http://www.zope.org>>
- [48] *Plone CMS: Open Source Content Management System*, <<http://plone.org>>
- [49] *Ruby Programming Language*, <<http://www.ruby-lang.org>>
- [50] *Ruby on Rails*, <<http://www.rubyonrails.org>>
- [51] *Prototype JavaScript framework: Easy Ajax and DOM manipulation for dynamic web applications*, <<http://www.prototypejs.org>>
- [52] *It's About the User Interface, Baby!*, <<http://script.aculo.us>>
- [53] *The Yahoo! User Interface Library*, <<http://developer.yahoo.com/yui>>
- [54] *Google Web Toolkit - Build AJAX Apps in the Java Language*, <<http://code.google.com/webtoolkit>>
- [55] *MySQL AB :: The world's most popular open source database*, <<http://www.mysql.com>>
- [56] *PostgreSQL: The world's most advanced open source database*, <<http://www.postgresql.org>>
- [57] JEREMY ZAWODNY: *Oracle buys Innobase. MySQL between rock and hard place?*, <<http://jeremy.zawodny.com/blog/archives/005490.html>>
- [58] *What is Mono?*, <http://www.mono-project.com/Main_Page>
- [59] *open.NET? Microsoft To Make .NET Libraries Available Under "Open Source"*, *The World Through the Eyes of an Evil Genius*, <<http://www.whurley.com/blog/2007/10/opennet-microso.html>>
- [60] *Welcome to TADS.ORG*, <<http://www.tads.org>>
- [61] *Lilypond, Music Notation for Everyone*, <<http://lilypond.org/web>>
- [62] *Blender*, <<http://www.blender.org>>
- [63] *Audacity: Free Audio Editor and Recorder*, <<http://audacity.sourceforge.net>>

- [64] *Nexuiz*, <<http://alientrap.org/nexuiz>>
- [65] *Sauerbraten / Cube 2*, <<http://www.sauerbraten.org>>
- [66] *Tremulous*, <<http://www.tremulous.net>>
- [67] *The GNU Operating System*, <<http://www.gnu.org>>
- [68] *Linux Distro Timeline*, <<http://730x.up.md/wp-content/44218-1.png>>
- [69] *redhat*, <<http://www.redhat.com>>
- [70] *IBM, Red Hat to push Linux in emerging markets*, <http://www.news.com/8301-10784_3-5866834-7.html>
- [71] *The Community ENTerprise Operating System*, <<http://www.centos.org>>
- [72] *Debian – The Universal Operating System*, <<http://www.debian.org>>
- [73] *Ubuntu Home Page*, <<http://www.ubuntu.com>>
- [74] *From Zero to Linux in 5 Minutes*, <<http://www.knoppix.net>>
- [75] *GNU + Cygnus + Windows = cygwin*, <<http://www.cygwin.com>>
- [76] *Firefox web browser — Faster, more secure, & customizable*, <<http://www.mozilla.com/en-US/firefox>>
- [77] *Thunderbird - Reclaim Your Inbox*, <<http://www.mozilla.com/en-US/thunderbird>>
- [78] *Calendar Project - Home of Lightning and Sunbird*, <<http://www.mozilla.org/projects/calendar>>