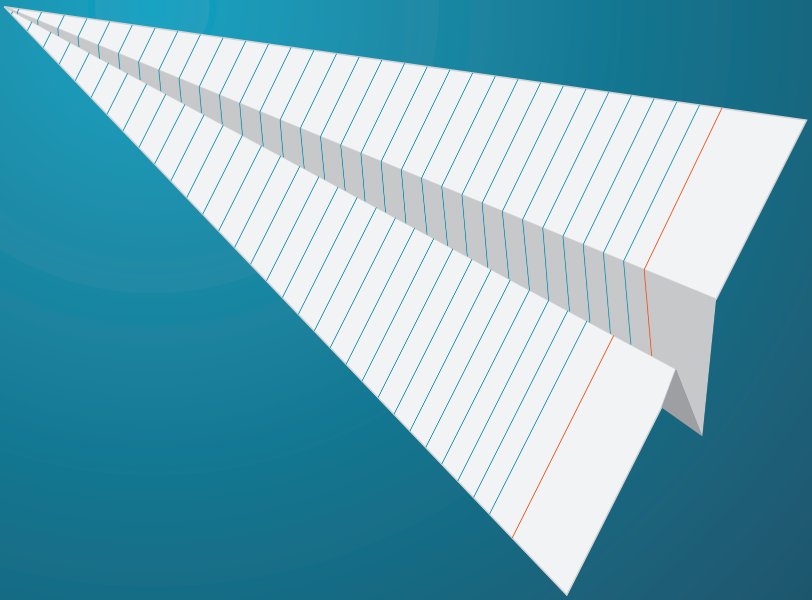
concursive
where business takes flight

WHITE PAPER

Considerations for Enterprise-Class Open Source Applications



concoursesuite
www.concursive.com

Considerations for Enterprise-Class Open Source Applications

Written by Tom Manos

The Emergence of True Enterprise Class Applications

A quick Google search of "enterprise open source" returns over 1.5 million results. There are web sites devoted to the topic; there are dedicated conferences and trade shows, and even a fairly large number of books. Most of these sources discuss using open source applications and infrastructure in one's business, and tacitly assume that this defines enterprise open source. This White Paper investigates a more rigorous rationale for the term, and describes what should be necessary for a particular piece of software to call itself "enterprise open source". It also describes a growing movement in open source: the enterprise open source application.

Defining "Enterprise Software"

Over the years, the term "enterprise software" has taken on many meanings, ranging from heavy-duty operating systems, such as Sun's "Solaris Enterprise System" and "Red Hat Enterprise Linux" to more classical definitions like Martin Fowler's at right.

For the purposes of this white paper, we will use a definition similar to Martin's: Enterprise Software is software that solves problems across an enterprise, rather than specific problems within an enterprise. Thus, most desktop applications would not be considered enterprise applications nor, in this case, would operating systems or database systems since they don't in and of themselves solve problems or process information.

Examples of true enterprise applications include Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), Content Management, and corporate financial applications. Does this mean that every ERP system is an example of Enterprise Software, somehow deserving of the title? No, not necessarily. There are some additional requirements that must be met before calling an application an "enterprise application". In no particular order, some additional requirements might include:

- **High availability.** If operations or revenue stop when the application stops, then the application must be continuously available.
- **Security**. The application will presumably be processing the enterprise's most precious possession: its information. As has been so clearly reported recently in the media, when dealing with competitive information, financial information, or customer information, strong security is a must.
- **High performance.** The application must perform adequately under the load it is given to support its users.
- **Scalability.** The application must scale to whatever size is required of it.
- **Support.** The application must be fully and formally supported. While a wonderful piece of software that fulfilled all other requirements might be called an enterprise application, no real company would use it. In the enterprise application world, support from the open source community alone is not enough.

Though there are doubtless many more requirements that could be added, in our view these are the most important, and are certainly enough to illustrate the case.

> "Enterprise Application is the name I give to a certain class of software systems: the data intensive software systems on which so many businesses run. Another, and perhaps better, name for them is Information Systems since these are systems that process and manipulate information."
>
> *– Martin Fowler, Chief Scientist - ThoughtWorks*

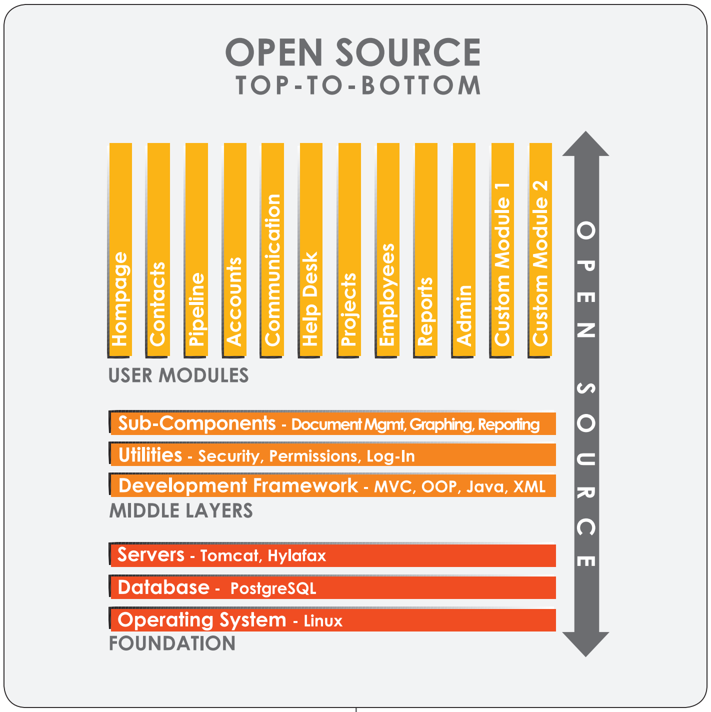
Building Enterprise Open Source Applications

So what does it take to design, build, market, and sell enterprise open source applications? Great and relevant software is obviously at the top of the list, but control is most certainly also on the list. Customers want and need the ultimate in control and flexibility, and the greatest protection against vendors whose interests are not, in the end, perfectly aligned with their own.

–continued on next page

After all, many software companies see their business first as selling software, and only secondarily as pleasing their customers and partners. Open source software helps shield the customer from these worries. Because the actual application code is in their hands, they are no longer at the mercy of an uncooperative or even adversarial vendor.

It is no surprise, then, that ever-larger enterprises are beginning to seriously evaluate open source solutions to determine whether such products provide a legitimate alternative to traditional proprietary applications. “We are making a strategic move to embrace open source technologies,” says Brian Shield, CIO of The Weather Channel. “Finding enterprise-ready products like ConcourseSuite have allowed us to mothball several expensive proprietary systems already.”



The challenge for open source providers, therefore, is to build an application that simultaneously embodies the spirit of open source, while also satisfying all the requirements necessary to be seriously considered ready for the enterprise, including the very large enterprise. This means solving a number of potentially conflicting requirements. On the one hand, it must appeal to an open source community, be accessible, understandable, inexpensive or even free, and run entirely on an open source infrastructure. On the other hand, it must be built on an enterprise architecture , support very large installations, and be capable of running on any application stack the customer specifies, with little or no change.

An Example of an Enterprise Open Source Application

Customer Relationship Management (CRM) is one of the most critical activities successful companies engage in. CRM is first-and-foremost using a business strategy. Successive waves of technologies–starting with index cards!–have emerged to help businesses do a better job of managing their customer relationships. Most CRM solutions available to date have involved expensive, proprietary software. Some 7 years ago, my company, Concursive, set out to change all that. I would like to use our long (at least by open source standards) experience to illustrate some important points about enterprise open source development.

We set out to design ConcourseSuite first-and-foremost using only open source or freely available components. We chose the following:

Platform

We wanted our application platform to be enterprise class and totally cross-platform and portable. The only available platform that solved all our problems was Java J2EE. While the Java language itself is not open source(although I think it will be soon), it is freely available and has a variety of free development tools and libraries. There are also compelling open source Java application servers such as Apache Tomcat and JBoss. We forewent scripting languages such as PHP and Python because, while they make great prototyping and rapid development platforms, they lack the enterprise security and scalability features we believed our enterprise customers required. They also are not well suited to building very large and complex systems. We did not think the CIO of a typical Fortune 500 company would be receptive to a critical business application built on such a platform. And although Microsoft’s .Net platform would have been a good candidate in terms of performance, scalability and power, it certainly fell short in the open source category!

> “We are making a strategic move to embrace open source technologies. Finding enterprise-ready products like ConcourseSuite have allowed us to mothball several expensive proprietary systems already.”
>
> *– Brian Shield, CIO - The Weather Channel*

Database

For our reference database, we wanted a system that we could distribute with ConcourseSuite, that was very standards compliant, and that provided very high performance and scale. The only database in this class at the time was PostgreSQL.

Operating System

There was really only one choice here: Linux.

–continued on next page

Java as the Platform for Enterprise Open Source Development

The Java J2EE stack is at once reasonably well understood and accepted by the open source community and one that with thoughtful and correct design and architecture will allow an application to fit nicely into the large enterprise.

Choosing Java as the development platform allows enterprise open source application developers to be completely operating system independent. If the application is designed with portability in mind, J2EE will allow it to run identically on Linux, BSD, Solaris, Windows, and just about any other operating system you or a CIO would be likely to specify. If the database interface uses only JDBC, standard SQL 92/99 queries, and no special or non-portable database capabilities, such as triggers and stored procedures, you have the beginnings of database portability. And if all the application's business logic is written in Java, the application will run identically and quite happily on PostgreSQL, Microsoft SQL Server, MySQL, Oracle, IBM DB2, and a host of other standards compliant databases. (There is certainly a tradeoff in performance and convenience using this database architecture, but as long as adequate performance across a range of installations sizes is possible, I believe that the customer should be the final arbiter in the choice of database.) Choosing Apache Tomcat as a reference application server allows one to easily move to JBoss, WebLogic or WebSphere when customers need the power and scalability provided by a full-blown J2EE application server environment. Taking advantage of advanced features of enterprise middleware then becomes a matter of writing code that is "vanilla" enough to run on all of the available choices. Sometimes this involves tricky design, but I have always found it possible. Furthermore, the development discipline such an approach enforces leads to the creation of very well structured code.

> "It may take some getting used to, but open source is the way of the future for enterprise software. With the product source code, customers have the ability to customize applications to their own specifications, in much less time and at a much lower cost of financial and human resources."
>
> *–Peter Winston, CEO, Project.net*

The Emergence of an Enterprise Open Source "Ecosystem"

Concursive is by no means the only open source application developer who has thought through the issues involved in building true enterprise class open source solutions. Indeed, one of the most exciting trends in the IT industry today is the emergence of a recognizable handful of serious enterprise open source products. Without the guiding hand of a single organization, there is now available a suite of complementary, enterprise-class products that solve a wide range of corporate IT problems. For example, in the content and document management space, you have Alfresco
 and Magnolia

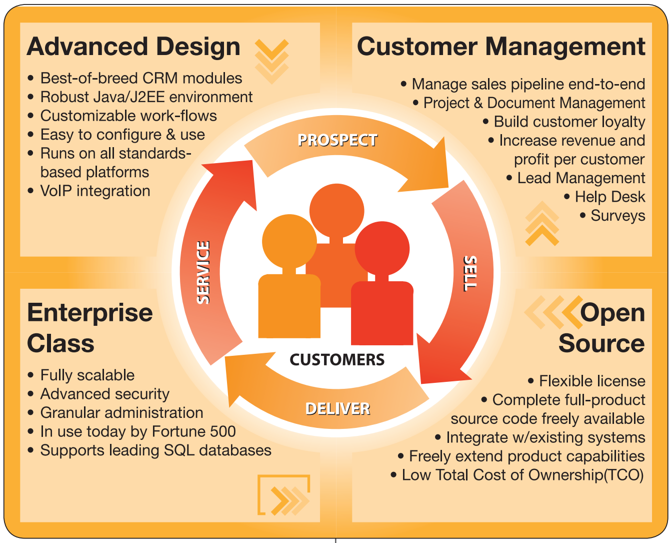
The emergence of this suite of enterprise class applications is also creating tremendous opportunities for VARs and Systems Integrators with open source expertise. "We are now able to offer our corporate customers cross-departmental open source business functionality," says Ron Bongo, CEO of CorraTech, a large east-coast open source integrator. "By delivering solutions using open source code within a Service Oriented Architecture (SOA), we can provide deep integration that's maintainable."

An Enterprise Open Source License

Despite the lack of a guiding hand, all of the above vendors have made substantially similar choices in the architecture they have chosen. In particular, they have all chosen Java J2EE as the correct development platform. They have also made substantially similar choices about how they license their products. Consensus is building around the key elements an enterprise open source license must embrace. In particular, the following elements are important to an effective enterprise license. First, the source code must be truly open. Virtually all of the vendors mentioned above have turned away from a "dual license" in which a stripped-down community version is offered under an open license, while a separate proprietary license governs the use of the complete version of the product. Instead, most of these vendors–Concursive included–make their full product source code available for download under a single open license. Second, the license must discourage "forking" of the code. As long as large enterprises have to worry about multiple variants of a product emerging, they will not embrace open source solutions for mission-critical applications. Instead, they need a unitary code-base backed by a real vendor offering professional service and support. An enterprise open source license creates this condition by restricting redistribution of the source code. Thus, while the full source code is truly open (i.e., freely available to download and inspect), it cannot be resold (presumably with modifications) without the vendor's permission. Finally, any modifications that a customer makes to the application for their own internal use must belong to them. Customers who undertake meaningful development to create significant competitive advantage for themselves cannot be forced to involuntarily submit their enhancements back into the core product. That said,

–continued on next page

we have found that in practice, all of our customers want their enhancements merged into the core code so that they are relieved of the burden of maintaining those enhancements in the future.



Advanced Design

- Best-of-breed CRM modules
- Robust Java/J2EE environment
- Customizable work-flows
- Easy to configure & use
- Runs on all standards-based platforms
- VoIP integration

Customer Management

- Manage sales pipeline end-to-end
- Project & Document Management
- Build customer loyalty
- Increase revenue and profit per customer
- Lead Management
- Help Desk
- Surveys

Enterprise Class

- Fully scalable
- Advanced security
- Granular administration
- In use today by Fortune 500
- Supports leading SQL databases

Open Source

- Flexible license
- Complete full-product source code freely available
- Integrate w/existing systems
- Freely extend product capabilities
- Low Total Cost of Ownership(TCO)

True Enterprise-Class Open Source

After seven years of dedicated design and development, we now have exactly the same code, governed by a single license, running in both a five-person shop under Linux, Tomcat and PostgreSQL, and in a Fortune 500 company running HP/UX, DB2, and WebSphere in an installation that will soon have thousands of simultaneous users in a fully clustered and replicated infrastructure. (There are, of course, many other customers between these two extremes.) Our goal all along has been that the customer should have the ultimate control of their application and the infrastructure it runs on. I would assert that this should be a key attribute of true enterprise class open source software.

Summary

In summary, by choosing the correct operating environment, using sound and sensible design, and keeping one's eyes on the goal, it is possible to craft an enterprise open source application that is accessible to customers and the community, is portable across operating systems, databases, and application servers, and runs on a 100% open source stack. I have provided a high-level view of how a single identical code base, governed by a single open source license, can be as appealing and accessible to a shop running it on an old desktop machine with 256MB of RAM as to a large data center using all the power of enterprise-class infrastructure and run by a battle-hardened CIO.

True Enterprise Open Source applications have arrived!

References

1 *For a description of enterprise Architectures, see the Wikipedia entry at http://en.wikipedia.org/wiki/Enterprise_architecture*

2 *See http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9000534 for a representative sample of the news.*

3 *http://www.alfresco.com*

4 *http://www.magnolia.info/en/magnolia.html*

5 *http://www.hyperic.com6 http://project.net*

About The Author

Tom Manos is the CTO and co-founder of Concursive Corporation (formerly Centric CRM). A nationally respected Internet expert, Tom has founded and held senior technology posts at several other notable ventures including North Wall LLC, and NetworkTwo Communications Group. In 1993, he co-founded InfiNet, one of the very first national ISPs and the largest "private label" ISP in the world at that time. InfiNet was subsequently sold to Landmark Communications. Prior to becoming an entrepreneur, Tom had a distinguished career as a US Naval officer, including a successful stint as the Director of Computer Operations for the World Wide Military Command and Control System (WWMCCS) for the Navy's Atlantic Command. Tom is a long time member of the Executive Advisory Board of the International Engineering Consortium, holds seats on several other boards of directors and is a frequent speaker at telecommunications industry conferences. Tom has a BS in Electrical Engineering and his MS in Computer Science. He lives with his wife, Gail and their two children in Chesapeake, VA.

concursive

where business takes flight

Concursive Corporation
223 East City Hall Avenue
Suite 212
Norfolk, VA 23510
Phone: 757.627.3002
Fax: 757.627.8773