Open Source
Business Models

Fred Domke, President, Business Integration Technology, Inc.

Overview

Open Source projects like Apache have shown that Open Source can play an important role in many enterprises' strategy. Other enterprises will choose licensed software for the innovations of the vendor and the support they offer.

The Open Source business model is getting noticed in the mainstream media. BusinessWeek notes in "Open Source: Now It's an Ecosystem" that this software movement is branching into not just mainstream business applications but also the associated services. Matt Rand at Forbes notes that "Open-source software is tearing through B2B-land." His article "Open Source Invades the Enterprise" asks "Can enterprise software giants adapt?"

One important consideration when evaluating your approach to open source is the business model behind the open source projects you may consider. This paper is an attempt to help enterprises distinguish between the business models and determine which models are compatible with enterprise strategy. The agendas behind the business models are widely disparate and that makes it easy to make the mistake of embracing or rejecting a project based on guidelines set across the models.

We have identified six basic open source business models: the "Lite" version of a commercial product, enterprise vendors' embrace of open source to show "I'm Open", the spin-off of commercial "Legacy" products into open source, the dual strategy of free and supported "Parallel Universes," the "Jump Start" opening of an un-launched commercial product, and true open source "Community" projects. We will examine each model and how projects may move from model to model. We will compare the value, concerns, and features of the various models. And we will promote discussion of this important topic.

### *Open Source Business Model Transitions*

The only thing that stays the same is change and open source is no different. Open source projects arise from different origins and may progress from one business model to another. They may start out as commercial code and they may give rise to commercial enterprise. Some of the possible paths that a project may take in its business model are depicted below.

Open Source Business Model Transistions

Page 1
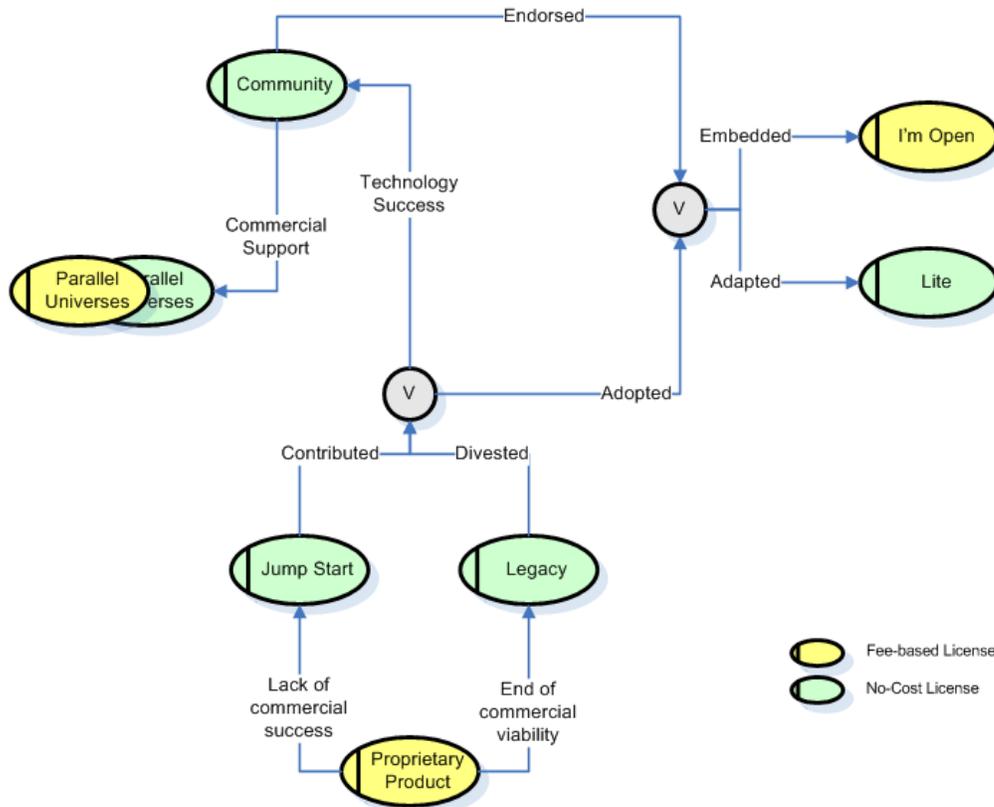
The Models Examined

"I'm Open!" – Embedded (Wannabe) – Enterprise-class software vendor includes open source in their for-fee licensed solutions. The code may have originated at the vendor and been `contributed' or is may be community code that is pervasive.

Examples: Apache Tomcat in WebSphere; Eclipse

Value:
- Leading enterprise software vendors often have deep pockets and can and do fund research efforts with outstanding talent and resources devoted to leading technology. They may genuinely want to promote standards and provide critical mass to advance technology. Their agenda may me that a "rising tide lifts all boats" – and their boat will be the first one in the water.
- Projects like Eclipse have succeeded where few thought it was possible in providing a platform where otherwise competing vendors can coexist. This can save on support and training when two or more vendors' products are deployed on the same open source framework.

Concern:
- It can be a slippery slope to use the open source components in a mixed model without backing into use of other well-integrated vendor software that may be bundled. This is usually either a trial version of a proprietary product or a "lite" open source version (see below).

"Lite" – Loss Leader (Trojan Horse) – real agenda is licensed software sale. Source is "open", but for all practical purposes, the only contributions or support comes from one vendor.

Examples: ActiveBPEL/ActiveWebFlow; Celtix/Artix; MySQL Community/Pro

Value:
- The "free" licenses provide an opportunity for experimentation and may, in some special circumstances, actually eliminate the need for a paid license.
- In distributed processing scenarios, a central platform may use the 'full' proprietary licensed and paid-for version and branches or partners or departments without enterprise-class service level requirements can implement the "lite" versions and be assured of compatibility.

Concern:
- If you're using procurement as a mechanism to enforce enterprise standards, a well-intentioned maverick may use the absence of a signed license document and the lack of need for a purchase order as an opportunity to introduce non-conforming technology.
- Likewise, a project leader under the gun to deliver a solution on time and on budget might invest resources in learning, deploying, and building a solution based on technology that will not scale to enterprise requirements or that does not provide the level of support necessary to meet service level agreements. In a dangerous tail-wags-the-dog scenario, the enterprise may be forced into purchasing licenses and making an undesired strategic commitment.
- Note that it may be difficult or impossible to mix open source and enterprise vendor strategies for support. The 'lite' version may provide source code, but it may be difficult to fold user-community bug fixes and enhancements back into the code. The supporting vendor will always control and guide the development of the 'lite' version for compatibility with the full version's technology and market position. Don't make the mistake of thinking that you can ever enhance this version to eliminate the need for the 'full' version. That contradicts the reason the 'lite' version exists and the vendor cannot let that happen.

"Legacy" – new life for old code (Dinosaur) – Unfortunately vendors cannot be expected to support unprofitable products forever. But many enterprises have implemented business solutions that continue to provide business value based on "legacy" products. There is nothing pejorative in the word legacy, and modern software architecture, especially service-oriented architecture, provides mechanisms to continue to derive value from these systems. But enhancements may be necessary to enable integration. How do we resolve these conflicting agendas? The software vendor may "contribute" the once for-fee licensed software as open source to the user community. The vendor no longer officially provides support. But the customers may well still be important to the vendor, so the vendor may provide resources to incubate the project. This is a short-term commitment and the real intention is to shift support to a community of users. Besides the possibility of re-invigorating the product as a community open source project, it is possible that a commercial interest may come forward and provide support in the "parallel universes" model.

Examples: OpenIngres, Fedora

1310 Papin Street
First Floor Center
St. Louis, MO 63103

Phone: 314-635-6351
Fax: 314-601-3092
http://BusinessIntegrationTechnology.com/
Email: info@BusinessIntegrationTechnology.com

3

Value:
- Since the product was functionally near end-of-life and on-going vendor support was in question, release of the source code in some form was necessary. Many software license contracts provide a source code escrow for this eventuality. But that shifts the burden to the respective internal IT departments and doesn't foster collaboration and economies of scale and skill. Opening the source enables the user community to take control and determine the value as a group rather than as individuals.
- The vendor may well provide a venue to incubate the community as well as expertise and marketing. This support may be well-beyond the means of a "start-up" open source project.
- Since the product was mature and providing value to (former) customers, it is in considerably better shape for mission-critical deployment than most open source projects.

Concern:
- Unless the community as a whole embraces the open source project, enterprises may be in danger of allocating resources to a dead-end effort.
- Without commercial support, use of the open source code may impact the ability to meet service level agreements and data center certification requirements. This impacts existing deployments because, from one perspective, this is a matter of discontinued vendor support.
- There may be alternative investments of resources that leverage advances in technology and software development methodology that provide higher returns.
- Mechanisms to establish a roadmap for the project, validate, prioritize, and allocate resources will need to be established before the value of community support can be realized. The respective former licensees may not effectively and efficiently embrace the culture shift to open collaboration.

"Parallel Universes" – Enterprise-class support (Brand Name) – Software vendor "piggybacks" on robust open source project to provide SLA-class support to enterprises that require it or find it provides value.

Examples: Red Hat Linux, Debian, JBoss, TAO

Value:
- The vendor provides support for the open source, enabling mission-critical deployment. They may provide training and consulting in the use of the open source, leading to partial or full self-sufficiency in the technology.
- The vendor usually extensively tests the open source across a variety or platforms and resolves portability issues when needed.
- The vendor provides a mechanism to request additional features to be added to the open source that they will build (possibly for a fee) and contribute to the open source projects.

Concern:
- You should be careful with building your business case to ensure that you actually realize the savings you anticipated. Open source products with single vendor support do not experience the same competitive pressures on pricing. It may well be that you still save a lot – just be careful.
- The concentration of expertise in a single vendor may limit the impact of the user community on direction for the product.

"Jump Start" – Bleeding-Edge Technology (Embryo) – vendor-created and attempted launch of a for-fee licensed software product, but unable to achieve critical mass in the marketplace. This may have been because supporting tools were not yet available or it may have been because the product was so advanced that potential buyers did not yet recognize the value. Open source effort is designed to appeal to like-minded early adopters – who may not have the budget to purchase an unproven but innovative technology.

1310 Papin Street
First Floor Center
St. Louis, MO 63103

Phone: 314-635-6351
Fax: 314-601-3092
http://BusinessIntegrationTechnology.com/
Email: info@BusinessIntegrationTechnology.com

4

Examples: PXE

Value:
- Innovative code is made available without cost.
- The vendor may provide support (possibly for a fee) and contribute to the open source projects.

Concern:
- Until the project migrates to a more stable business model, this is a high-risk project that should be undertaken with the understanding that there may be no outside support available – with or without payment of fees.

"Community" – Collaborative Software (Good Guys in White Hats) – technologists created a solution to a known problem for their own needs and saw the benefit of sharing code, innovation, and support across organizations for the benefit of all.

Examples: Apache Commons, Mule

Value:
A robust community will provide many 'sets of eyes' to find defects and 'many sets of hands' to remedy. It also supports innovation and rapid adoption of standards.

Concern –

- There is no commercial support available to back up a service-level agreement.

Comparison

Open Source Features by Business Model

	Lite	I'm Open	Legacy	Parallel Universes	Jump Start	Community
Free Version	Y	N	Y	Y	Y	Y
Paid Version	?	Y	N	Y	?	N
Vendor Support	Y	Y	N	Y	Y	N
Vendor Control	Y	Y	N	?	Y	N
Community Support	?	Y	?	Y	?	Y

Conclusion

There are many business models for open source and each has an important and valuable role to play. It is important for an enterprise to consider their specific needs and how the business model behind the open source project aligns with these needs. It is also important to recognize that the business model may evolve as the open source project and the community that uses and supports it evolves. Will the business model continue to provide the value you need? If not, what are your options?

Open source provides an exciting channel for innovation and a fascinating array of business models. If used wisely, it can provide dramatic benefits in cost and quality. Business Integration Technology's open source strategy is to work with our customers and leverage their existing IT infrastructure where it makes sense, but to leverage open source to fill the gaps and integrate it all in a service-oriented approach.

About Business Integration Technology

Whichever choice you make, BIT can work with you to effectively implement a collaborative trading partner network that meets your needs and conforms to your standards. If you would like an independent assessment of how Open Source may fit in your organization or an assessment of the best-of-breed offerings from software vendors, BIT can help you make the choice that best meets the needs of your organization.

Business Integration Technology (BIT), a leader in B2B integration, designs and implements highly cost effective business-to-business connections that eliminate the costs of doing business with paper, phone and fax. BIT brings innovative value to any firm looking to improve cycle time and reduce cost.

You only pay us for the work our business-to-business experts do, not for licenses or transaction fees. You own the solution, you run it, and you save month after month.

BIT is also a partly-owned subsidiary of Daugherty Business Solutions with over 20 years experience and over 400 consultants in St. Louis, Atlanta, Minneapolis and Chicago.

1310 Papin Street
First Floor Center
St. Louis, MO 63103

Phone: 314-635-6351
Fax: 314-601-3092
http://BusinessIntegrationTechnology.com/
Email: info@BusinessIntegrationTechnology.com