

Background Materials for:

**Making Advanced Technology Work for
Community Serving Organizations:
the Potential Impact of OSS and ASPs**

**University of Michigan
Ann Arbor, Michigan
May 12-13, 2000**

**Editor:
Jennifer Perkins**

**Contributors:
Michelle Bejian
Jeremy Birnholtz
Eric Hofer
Hyun Joon Kim
Cliff Lampe
Nat Poor
Joel Santodomingo
Brian Spolarich**

**Faculty Advisor:
Michael D. Cohen**

Table of Contents

INTRODUCTION	1
COMMUNITY SERVING ORGANIZATIONS	3
APPLICATION SERVICE PROVIDERS	10
OPEN SOURCE SOFTWARE	17
"NEW PHILANTHROPY"	25
CONCLUSIONS	27

Introduction

Community-serving organizations serve on the front lines of our society, providing assistance and support where they are needed. Like other organizations, however, they must manage information about their activities, clients and resources. Unfortunately, their limited resources - and often their small size - put them at a disadvantage when it comes to implementing technology in a way that is useful and productive.

Why is it that community-serving organizations have difficulties making advanced technology work for them? Is it a matter of cash for software and hardware? That is certainly a part of the problem, but it is by no means the thorniest part of the problem.

First, there are development costs. Purchasing a database application or other software is, of course, not the final solution to a particular problem. It must be customized, configured or programmed to some extent in order for it to be useful to the people on the front lines in the organization. In some situations, there are few feasible off-the-shelf applications available. The organization is then faced with the need to fund the development of an entire application.

Second, there are ongoing operational costs, such as the technical staff needed to administer complex and complicated technology. In addition, there are the costs of the underlying infrastructure needed to support many of the technologies needed to address the needs of the community-serving organizations.

In both of these cases, the costs to any one organization could be reduced by pooling their resources and sharing the development and operational costs. Open Source Software is one approach for sharing development costs across organizations. Separate organizations or individuals can build on the effort of others and provide additional effort to the group. Application Service Providers are an approach to distributing the operational costs of technology across many organizations. Small organizations can, in effect, rent the services they need without having to build all of the infrastructure on their own.

In combination with some of the changing attitudes toward social investing, these two models provide the potential for great strides in bringing technology to the front lines of our communities. It is our hope that this conference will help determine whether this potential can indeed be leveraged into success.

What follows is a collection of summaries created by students and faculty at the University of Michigan, which we hope will allow the Workshop participants to start the weekend with some common ground. The first section addresses the capabilities, funding models and technology needs of small community-serving organizations. The second section looks at Application Service Providers (ASPs), providing an overview of the current state of the industry. The third section describes the Open Source Software (OSS) paradigm, and its potential for ASPs and community-serving organizations. The fourth section describes some of the changes in the world of philanthropy.

While reading this document, please keep in mind that each section is necessarily an incomplete, and somewhat oversimplified picture of the world. Each component is intended as an introduction to that topic, so that participants will have some familiarity with issues outside their own area of expertise. We are acutely aware that each reader of this briefing document is an expert in some part of the larger picture that the Workshop is trying to bring into focus. This more or less guarantees that every reader will find some fault - or at least insufficient detail and nuance - in some portion of what we have provided for the whole group! While we are sure we have paid that cost, we also hope we've achieved the intended benefit: that everyone attending the Workshop will gain a better sense of the areas outside their own expertise that are key to our joint effort.

Nonprofit Organizations: Information Technology Capabilities and Resources

Introduction

According to a report made available by the Nathan Cummings Foundation, the nonprofit sector in America encompassed 1.4 million organizations and \$500 Billion in annual expenditures, as of 1993. These organizations provided vital arts and civic activities, and health, educational, and social services. Considering this Workshop's aim to clarify and realize the potential of new developments for community-serving organizations, a clear understanding of nonprofits' current status is necessary for all participants from other sectors.

A word is in order at the start on our somewhat casual terminology. We are going to use "community-serving" and "nonprofit" more or less interchangeably here although a more scholarly approach would make a number of detailed distinctions. We realize that there are nonprofits that are not closely tied to communities, and that organizations can serve communities while operating for-profit. Indeed, the growing importance of mixing nonprofit and for-profit models is a focus in a later section. But the background information we want to provide about community-serving organizations is mostly collected on "nonprofits". So we'll use both terms, and try to keep in mind that we have the smaller and more community-oriented organizations particularly in mind.

The following represents a brief survey of the IT capabilities and awareness of community-serving organizations (Section I) and major revenue sources and strategies for nonprofits (Section II). The information in this section has been drawn from a number of studies. It is intended to provide a picture of the sector as a whole, but given the size of the nonprofit sector, we acknowledge that there are many exceptions to the generalizations that can be drawn from these studies.

For this purpose, we have drawn on two studies of IT usage in nonprofit organizations in Michigan and Silicon Valley for Section I. The data included statistics on Internet connectivity,

hardware and software inventory and usage, technology support, and attitudes towards technology. Given the similarities between the two studies, we feel it may be safe to extrapolate that these numbers are representative of nonprofit technology usage nationwide. Following that, Section II delineates the major revenue sources of nonprofits in order to clarify how new uses of information technology may be supported.

Section I. IT Capabilities and Awareness

Hardware

Over 90% of the organizations surveyed have at least one computer. Specifically, 93% of Michigan nonprofit organizations report having at least one computer, with 64% reporting having 10 or fewer computers, while 99.5% of Silicon Valley nonprofits report having at least one computer. More than half of the nonprofits surveyed report having a computer network in place -- 57% of Michigan nonprofits and 62% of Silicon Valley nonprofits. Given the presence of computers in the nonprofit workplace, the majority of Michigan organizations report that obtaining additional computers for staff members is not a major priority. Similarly, approximately half of Silicon Valley nonprofits report that more than 75% of their staff have their own computers in the office.

Connectivity

Like computer usage, Internet connectivity is common in nonprofit organizations. Over 75% of the nonprofit organizations surveyed report having some form of Internet access (77% Michigan, 90% Silicon Valley). Of the connected organizations surveyed in Michigan, only 28% report having a high-speed Internet connection, while 91% report connecting to the Internet via modem (some organizations do both). Of the 23% of Michigan organizations that are not connected to the Internet, 39% plan to connect, 22% are unsure, and 39% do not plan to obtain Internet connectivity. In Michigan, at least, there is still a relative strong belief that Internet access is too expensive for some nonprofit organizations. Nearly a quarter of organizations that are not yet connected in Michigan cite cost as a major obstacle to Internet access. Attitudes about the ability of the Internet to serve nonprofits are still very mixed. Only 58% of nonprofit organizations

surveyed in Michigan report that the Internet is improving nonprofit marketing and product/service delivery.

Operating Systems and Applications

The Microsoft Windows operating system seems to be dominant in the nonprofit sector. Of organizations surveyed, 86% report Windows 95/NT to be the most common OS in their location, while 50-60% of organizations report having and least one computer running Windows 3.1. Additionally, 15-25% of nonprofits report having at least one Macintosh.

The most important applications in Michigan (based on frequency of use and perceived value) are:

- Word processing, spreadsheets, Email, Web Browser and Finance packages.
- Less popular (but still noteworthy) are Volunteer management, Calendar management, graphics, and mapping/GIS software

The Silicon Valley study took a different approach. It asked what software organizations were using for various functions, and then (among lots of other things) asked them about the weaknesses of the software. It is also worth noting that about 10% of the organizations are using Microsoft Word for all of the functions listed below. All percentages are of the total. They are not subsets of each other.

- Donor Management - 63.3% use an off-the-shelf product, 12.2% say the software is effective, and 20% plan to replace it soon
- Fundraising - 64.3% use an off-the-shelf product, 13.5% say it is effective, and 17.9% plan to replace it soon
- Program evaluation - 55% use an off-the-shelf product, 6.9% say it is effective, and 20% plan to replace it soon
- Volunteer Management - 52.2% use an off-the-shelf product, 6.5% say it is effective, and 21.7% plan to replace it soon.

When asked about the effectiveness of software packages they currently use, very few organizations reported that their software effectively met their needs (where "very few" -<

~10%). In all of these cases, the plurality of respondents listed "It isn't flexible enough to organize, analyze or present the data as we need it" as a weakness of their software.

When asked about their goals for enhancing their use of computer technology, the top 3 responses were: training (42.1%), networking (LAN/WAN) (34.1%), and purchasing software (31.7%)

Budget and Technical Support

Of the organizations surveyed, 67% reported that expense was not a primary barrier to Internet access. Cost is, however, a primary obstacle in upgrading applications software.

Roughly 50% of the Michigan organizations surveyed depended on an existing staff member for hardware and software support. In most cases, these staff members were only informally trained in technology. 43.7% of Silicon Valley organizations report that a staff member is the primary resource for hardware support and 53% say that a staff member is the primary resource for software support. This is typically not the only function of this staff member. Many are high-level personnel.

Attitudes

Most organizations reported no need for additional computer hardware. For Michigan, the majority of organization felt that the Internet was improving nonprofit marketing and product/service delivery. 75.7% reported that their organization was "very supportive" of new technology and 22% reported they were "somewhat supportive."

Section I Conclusion

Nonprofit organizations clearly demonstrate an awareness of and a positive attitude towards technology. For most organizations, hardware and funding don't appear to be the major obstacles to effective technology usage. Rather, it seems that the software tools currently in use are woefully inadequate. The market seems ripe for system solutions that are customizable and are better matched to the specific needs of individual organizations.

Section II. Revenue Sources

Section Introduction

There are three primary sources of funding from nonprofits. These revenue sources include private contributions from individuals, corporations, and foundations, grants and contracts from government agencies, and fundraising through commercial activities. This section outlines the prevalence and issues surrounding each of these funding sources. While we use information on nonprofit revenues in general, the issues that become visible can be applied to the particular problems of supporting advanced uses of technology. (Much of this material comes from Karen A. Froelich, "Diversification of revenue strategies: Evolving resource dependence in nonprofit organizations", *Nonprofit and Voluntary Sector Quarterly*, San Francisco; Sept.1999).

Revenue Sources

Private Contributions

Private contributions include donations from individuals, donations from corporations, and foundation grants. This source of funding has been declining as a percentage of total revenue for most nonprofits, down to 19% of total revenue in 1996, from 27% in 1986, and 30% in 1980.

One of the sources of private contributions is individual contributions. These contributions are typically costly for a nonprofit organization to secure, as they frequently involve outside paid solicitors, instead of a nonprofit's own employees. As such, soliciting individuals often incurs excessive fundraising costs. This high cost of obtaining funds is coupled with a high degree of unpredictability as to the success of the fundraising effort and high instability of contribution over time for individual contributors. Additionally, nonprofits must also worry about goal displacement in order to satisfy the wishes of contributors.

Another source of private contribution is donations from corporations. These contributions take the form of either monetary donations or donations of goods and services. Like individual contributions, stability is a major concern for corporate contributions, as donations are highly dependent on year-to-year revenue volatility. Additionally, there are concerns over the increases in formality of process and structure in nonprofits to meet criteria for gift allocation.

Foundation grants are the last source of private contribution. Like corporate donations, there are concerns about how the pursuit of foundation grants can alter structure and process and contribute to goal displacement in trying to meet the eligibility criteria.

Government Funding

The availability of government funding, in the form of grants and contracts, has varied greatly over time as changes in political leadership and public policy initiatives has occurred. Currently, the percentage of nonprofit revenue from government sources is declining, outside the healthcare field. This source of funding is often the least available, but has fewer concerns than many of the strategies concerning private contributions. Government funding is the most stable of the revenue sources and goal displacement appears to be less serious. There are, however, concerns about how this revenue source can change the internal processes and structures in an organization towards bureaucracy.

Commercial activities

Another fund raising strategy is for nonprofit organizations to become more like for-profit organizations, charging for services in order to support costs. Mission or service related commercial activities appear to be increasing common. Fees for service represented 25% of nonprofit revenue in 1980, but increased to 39% of total revenue in 1996. Considerable controversy exists around this funding strategy. The pro’s and con’s are listed below.

Cons	Pros
<ul style="list-style-type: none"> • Potential loss of values distinctive to the NP • Degrading Social missions second to revenue and profitability goals • Leading to an identity crisis • Eventual elimination of special privileges and protections 	<ul style="list-style-type: none"> • Little evidence of goal displacement • Enabling greater flexibility • Autonomy for nonprofit organizations

Section II Conclusion

Side effects that are often similar accompany each source of funds across various revenue strategies. A key to organizational viability and integrity is to understand the opportunities and tradeoffs, choose revenue strategies that are consistent with the mission, and conscientiously respond to management challenges presented by each strategy. However, a greater variety of resource providers typically leads to a corresponding increase in funding criteria, and satisfying the criteria of one provider may preclude satisfying another. Thus, revenue diversification can be seen as a double-edged sword.

Application Service Providers

Introduction

Over the past few years an increasing number of organizations are migrating business functions that they have traditionally performed using in-house information technology staff to external service providers, who host these functions on their own systems and typically provide access over the Internet. These service providers, called Application Service Providers (ASPs), represent a new development in the information technology industry, and a significant opportunity for organizations who want to use advanced applications and systems but cannot afford, or do not wish to bear, the high costs of in-house systems deployment, implementation, customization, and management. This section examines the ASP industry, its history, makeup, and some of the drivers that enabled its emergence. We also examine briefly some of the advantages and disadvantages of the ASP model.

What is an ASP?

Typical Internet Service Providers (ISPs) provide Internet access, e-mail, and host web pages for individuals and organizations. As the ISP market has developed and matured, more advanced ISPs have provided customers with database hosting and other complex services to support more complex web applications beyond mere 'presence pages'. Application Service Providers take this one step further and host on their systems packaged software applications, such as enterprise resource planning systems, payroll, accounting, or human resources applications. ASPs also provide application support, customization and integration services, and service level agreements to guarantee that the customer's application will always be available when they need it.

According to International Data Corporation (IDC), an ASP has the following defining characteristics:

- Application-Centric Business Model: The focus is on selling and managing the application.

- Product is Access: The customer buys access without having to make other investments up front.
- Central Management: The customer doesn't manage anything at their site.
- One to Many: Unlike IT outsourcing, the emphasis here is on defining opportunities where the same solution, with minimal customization, will meet the needs of many customers.
- The Buck Stop Here: From the point of view of the customer and the ASP the ASP is responsible for delivering on all terms of the contract, even if the fault lies elsewhere.

In their report titled "ASP's Impact on the IT Industry" (September, 1999), IDC forecasted that "worldwide spending on ASP services for enterprise applications will reach USD\$2 billion by 2003. Based on preliminary estimates, we believe the full range of collaborative and productivity applications will bring the entire ASP market to approx. USD\$4.5 billion by 2003."

History of Outsourcing

(adapted from <http://www.businessfinancemag.com/archives/appfiles/Article.cfm?IssueID=304&ArticleID=13186>)

Early on in business computing, mainframe systems were very expensive and out of the reach of all but the largest organizations. Many organizations instead turned to "service bureaus", who essentially rented mainframe time generally for standardized tasks such as payroll processing or accounts receivable billing. These services were focused on batch-oriented, number-intensive processing and did not support end-to-end applications. As more affordable mainframes, then minicomputers and finally desktop workstations emerged, many of these tasks were brought in-house and demand for service bureau services dropped.

Organizations focused on moving applications off of the mainframe and closer to their users. The goal here was to align IT applications closely with business processes and operations, and to gain both flexibility and control in the process. Over the past ten years IT has moved increasingly out of the back office and into every part of the enterprise, driven by the need for increased automation of business processes and an increasing reliance by the organization on accurate and immediate data. With it has come dramatically increased IT expenses, staffing requirements, and complexity. Organizations are now looking to simplify the IT systems management equation

without dramatically sacrificing access to the information they need to keep their operations competitive.

The past ten years have also seen the emergence of the Internet, which represents a very cost-effective, standard mechanism for distributed and diverse computing systems to communicate. Layered over the Internet connection is the web browser, which provides yet another standardized, (relatively) non-proprietary mechanism for deploying platform-independent content and applications. Neither of these components add significant cost, yet enable the explosion of ASP products and services that are emerging today by facilitating standardized access to remotely-hosted services.

ASP Industry Drivers

There are a number of intersecting phenomena that are driving the development and growth of the ASP industry:

- Portal and interface software suppliers want to turn one-time revenues into annuities, increase customer loyalty and lock-in.
- Complex packaged software vendors want to extend market reach and reduce complexity of deployments.
- Systems integrators want to leverage intellectual property created during custom software development into an ongoing revenue stream.
- Organizations want to gain easier access to powerful applications without high up-front installation costs and high ongoing support costs and predictable ongoing costs.

Organizations are also looking to move IT costs out of their capital budgets and turn them into operating expenses, and want to focus on 'core' and outsource 'context' activities to someone else. Skilled IT personnel are hard to come by and should be focused on increasing shareholder value (or delivering services to the community), and not on running email servers.

ASP Industry Enablers

There are also a number of important developments that have enabled the ASP industry to flourish. Among them are:

- Increasing ubiquity of information technology in all areas of business.
- Increasing complexity and costs associated with information technology.
- Increasing pace of information technology development and innovation.
- Broad support for and deployment of Internet-based communications protocols.
- Large amounts of network bandwidth becoming available at increasingly lower prices.
- Increasing standardization across technology and business process domains.

ASP Industry Market Segments

The ASP marketplace is a complex one, with many organizations participating in several capacities or roles. The breakdown below is based on an industry directory maintained by ASPNews (<http://www.aspnews.com/>), and is intended to provide a general sense of the scope of the industry and some of the organizations who are participating in it.

Service Providers

Traditional ASPs:

- Enterprise ASPs: High-end business applications, which typically require significant customization and support. Companies in this space include Digex, IBM Global Services, Qwest, UUNET, and USWeb.
- Local/Regional ASPs: Support smaller organizations in a defined geographical area (similar to regional ISP market).
- Specialist ASPs: Supports a particular application domain, such as HR automation, customer relationship management applications, web site services, etc.
- Vertical Market ASPs: Provides solutions tailored to the unique needs of a given industry, such as health care or financial services.
- Volume Business ASPs: Prepackaged application services for the small to medium-size business marketplace. ASPNews does not categorize standard web hosting providers here, but rather with ISPs.

Business Process Providers: Outsourcing providers for business functions such as benefits administration, distribution, field support, or maintenance services. KnowledgePoint is a well-established player in the HR space.

Commerce Network Providers: Commerce networks were traditionally proprietary value-added networks used for electronic data interchange (EDI) between companies. Today the category includes the new generation of electronic purchasing and trading networks. New players in this market segment include companies like OpenMarket and BroadVision. More traditional EDI network vendors include CompuServe Network Services (now part of UUNET/MCI WorldCom), GE Info Services (GEIS), Harbinger, and Sterling Commerce.

IT Service Providers: These are the traditional IT services businesses, including IT outsourcers, systems integrators, IT consultancies and value added resellers. CompuWare, EDS, and IBM's service business are big players in this industry segment.

Internet Service Providers: The ISP category embraces both access providers and Web server hosting providers. These are companies like UUNET (part of MCI WorldCom below), PSINet, BBN, etc.

Portals: This category is for companies whose primary business is operating a Web destination site, hosting content and applications for access via the Web. Companies like Yahoo, Excite, etc. come to mind here.

Telecom Providers: Both traditional and new-wave telecommunications network providers fall into this category. These are companies like ATT, MCI WorldCom, Sprint, Qwest, Savvis, and Level3.

Vendors

- Hardware Manufacturers: Manufacturers of servers and client devices come in this category. These are companies like Dell, Intel, Compaq, Cisco, and Sun.
- Online: Online vendors operate and deliver software which they themselves have developed exclusively for online delivery, typically accessed via a Web browser. They deliver the applications either direct to users or via other service providers. AOL and MSN are big players here.
- Onsite: Onsite vendors develop software designed for installation and operation on a customer site, including those who develop software for use on customer's own Web servers. Although the software may be offered by online providers, the vendor's mainstream business

is to develop software for sale rather than online delivery. This is where a large bulk of the software industry fits into our ASP picture. Big players here are backend companies like Oracle and Microsoft, as well as application vendors such as: BAAN, SAP, Lawson, and PeopleSoft in the Enterprise Resource Planning space; OpenText, Documentum, Lotus, and FileNet in the Document Management space; Remedy, Clarify, Applix, and SilkNet in the Customer Relationship Management space. One of the current big opportunities in the ASP space is for these vendors to find ways to package their applications so that they can be easily hosted by ASPs.

- **Platform:** This category is for vendors who develop the infrastructure software which supports the hosting and online delivery of applications.

ASP Advantages

Outsourcing applications and systems to an ASP have some distinct advantages for organizations that use them. Among them are:

- Sharp reduction of the need for in-house hardware, systems development and support staff, etc. Systems can be managed with higher quality, reliability, and network bandwidth by distributing costs across many customers.
- The ASP can distribute fixed costs of upgrading and maintaining complex software applications and systems across a large set of users.
- Typical deployment models (using a web browser) eliminate need for costly managing of more complex desktop systems and applications.
- Simplified implementation and deployment of applications to new users (ASPs do it better than you do, hopefully).
- Significantly decreased cost of entry for smaller organizations.
- Predictable costs (usually per user per month).

ASP Disadvantages and Risks

The ASP model is not right for every organization. Some potential risks and disadvantages include:

- Outsourcing increases risks associated with external factors such as network congestion, slow response time, security intrusions, etc. Proprietary or sensitive data could be stolen or corrupted.
- Not every application has a complete web interface. This may limit the available choices for some customers.
- Organizations that require significant customization beyond an ASP's standard offerings may not be able to outsource, or if they do it will be at a higher cost (both up-front and ongoing).
- Use of multiple ASPs for different portions of an organization's application suite may create serious problems of data integration.

Open Source Software

Introduction

This section introduces the concept of open source software (OSS), and shows how it may dovetail with the objectives of application service providers (ASPs), nonprofits, and social investors.

The open source process is based on the notion that the source code for software should be available to all, programmers and users. Access to the source code allows programmers to make modifications if they want to fix problems or add new functions to the software. Availability of the source code does not preclude profit making, but changes the manner in which the profits are created. Many OSS organizations create revenue by selling various types of services as well as products.

Some people use the metaphor of automobiles when explaining open source. Closed source software is like buying a car with the hood locked. Only the dealer knows what's under the hood, and only the dealer can access the engine to fix it. If the car's hood could be opened by the owner, the owner could make modifications, fix problems by him- or herself, or take the car to a different mechanic. It would be very unusual if the hoods of cars were locked and only the dealer could open them, but that's the way a lot of software is today.

In OSS, since the source is "open" it is usually free of charge for all, users and programmers (depending on the license used). This has caused some consternation among traditional software businesses and some misunderstanding in the press, but new companies (such as Red Hat) have adapted to the open source business model. Instead of charging for the software product, they charge for distribution and services (such as user manuals and technical support). Thus although the software is often free it is possible to generate revenue.

There are some very widely-used examples of open source software, such as the Apache web server (the most widely used web server), the GNU project, the Linux kernel, the Mozilla web browser, and Sendmail (the most widely used mail router).

The flexibility, adaptability, and licensing of software developed in an OSS model has potential benefits for a partnership of nonprofit organizations and ASPs. By using software that is open source, ASPs are able to tailor the software to their needs and the needs of their customer nonprofits, take advantage of the contributions of other programmers, and provide different levels of pricing based on service rather than on product. Philanthropists can fund the development and retooling of open source projects, reducing the cost of the services provided by ASPs to nonprofits and improving the quality of their operations.. Nonprofits who use the services of ASPs, they then have time and resources available which would otherwise be devoted towards in-house IT operations. Thus all groups benefit.

Where ASPs use closed source software, nonprofits may not be assured the level of quality that is available from OSS. A philanthropic investment would only benefit one ASP and not a larger community, since the open source software would not be widely available. By using and contributing to an OSS community, both nonprofits and philanthropists would be contributing towards their larger goals, and ASPs could spend more time on service and less time on programming, since most likely other programmers would help with coding in the traditional OSS community way.

Open Source

The term “open source” refers to a few different things simultaneously: a licensing model, an economic model, and an often community-based method of programming. Generally, open source software is software where users have access not only to what is usually thought of as “the software,” i.e. the application or whatever it may be, but also the source code for that software. Users have access to the source code so they can recompile it into a piece of software that improves upon the original. Most open source licenses require that the software, even when modified, remain “open” with the source code available. One may not modify a piece of open

source software and then copyright the result under a traditional copyright if the original license does not permit it.

In open source software development, bugs may be less of a problem. One well-known adage in the open source community is from Eric Raymond. “With enough eyeballs, all bugs are shallow,” meaning that with enough programmers bug fixing becomes less of a problem than it is in traditional software companies. Certainly it would be difficult to pay and house such a large number of programmers, but that’s where open source and the Internet work hand in hand. By making a piece of software open source and by making it available over the Internet, the software is available to every almost programmer in the world who might want to work on it.

The financing of open source software is also an often-misunderstood topic. People and organizations can download open source software under the conditions of the license for that software. Many open source programmers contribute their time for free. This does not mean that there are neither economic benefits nor other gains to be made with open source software. If a software house releases a piece of software under an open source license, then that company could receive help with programming from people outside of the company. This could mean a loss of revenue, depending on the license chosen, since the software could then be available for free. Yet this does not preclude the selling of services related to that piece of software, and indeed this is so far the main economic model that many businesses involved in open source use, and use very successfully. In terms of a goods and services framework, the goods (software) are free and one then charges for related services (such as technical support, books and manuals, or other such things). If the goods are free, it makes it more attractive to all users, thus increasing the market for the services. If the software had a high cost, then fewer people would use it and the market for the services would be smaller.

The number of programmers who might work on any piece of open source software is potentially very large, although if no one is interested in coding for free then they can be paid instead, and some large projects used this combined approach. All users can benefit because fixes, updates, and additions to software can happen faster with more programmers working on a project. The quality of the software can be better than other commercial offerings, although historically it has

not always been this way because often programmers have coded for other highly savvy users like themselves. This is changing as open source software attracts a larger user community. If users need services related to the software, they are available, and usually the prices are comparable to those for closed source software. Moreover, a surprising level of voluntary support and advice is available in many OSS communities. This "gift economy" is one of the striking attributes of OSS that well run projects for community-serving organizations may be able to mobilize.

There are many benefits for all stakeholders in developing software with the open source model. Companies who produce or contribute to open source efforts maximize the market for their services, while greatly reducing the effort needed for programming. Users, who might be programmers or companies that can hire programmers, get software that meets their needs and that can be updated, fixed, or added to, quickly and easily.

Open Source Licensing

Licensing is an important issue for open source, much as it is for closed source. With closed source, there are many laws which relate to the end users' rights, some of which are very recent, as well as manufacturers' shrink-wrapped end-user license agreements (EULAs). There are different licenses that meet the open source criteria (below). Choosing a license for a product is an important step, as it dictates the terms under which users can modify the code.

A Brief Introduction courtesy of O'Reilly - ...

Tim O'Reilly, an influential member of the open source community, wrote this overview of three open source licenses. We have reproduced it here with permission. The original is at <http://www.edventure.com/release1/1198.html>. (Originally published in Release 1.0, November 1998 (31 pages in print); <http://www.edventure.com>. © Copyright 2000, EDventure Holdings Inc. All rights reserved.)

A Brief Introduction to Open-Source Licensing

At its best, open-source licensing fosters cooperation, sharing and symbiosis between software creators and software consumers. There are many established and successful approaches to open-source licensing. Here are three important licenses, presented from oldest to newest.

BSD-style licenses

BSD-style licenses (so called because they were used for the Berkeley Standard Distribution of UNIX) are the oldest and least restrictive. They give licensees the option of creating private derived works (traditional commercial software with unpublished source code). Contribution of changes back to the public version is optional.

Some in the Open-Source community resent third parties taking from the public pool of software without contributing. (In economics, this is called the free-rider problem.) But despite the lack of a mandate, voluntary cooperation abounds. BSD-licensed software provides a great deal of the Internet's functionality through BIND, Apache and sendmail.

GNU general public license (GPL)

The GNU GPL, authored by Richard Stallman in 1983, is the GNU Project's implementation of the Copyleft concept. While copyright provides a monopoly on the right to create copies and derivative works, Copyleft grants unlimited permission to copy and modify. However, Copyleft obligates the user to distribute, without fee or additional license terms other than Copyleft, the source code to all derivative works.

The focus of the GNU Project is "free software, where free refers to freedom and not price." You can sell free software, but you must also give away the source code. The GPL is "viral" in that one cannot combine GPLed work with work governed by different licenses. If you enhance a GPLed work, your enhancements fall under the GPL terms. The GPL has a less-viral form, the LGPL, used for function libraries.

The GPL excels at preventing the proprietary fragmentation that has caused so much harm to the UNIX market. GPL success stories include the Linux kernel, the GNU C Compiler and the Samba file server.

Mozilla public license (MozPL or MPL)

The MPL, authored by Netscape Communications as part of its open-source release of Communicator 5, strikes a balance between the BSD license and the GPL. Private derivative works are permitted, while changes to MPL-covered source must be made freely available on the Internet. The MPL, however, is non-viral: additions to (as opposed to modifications of) the MPL-licensed source which form a "larger work" may be licensed differently and need not be published at all.

The economics of open source are still evolving, and so are the approaches to licensing.

OSI's Definition

The following is a heavily edited (for length) copy of the Open Source definition used by the Open Source Initiative. It is not a license, but a set of criteria: OSI will label licenses which follow the criteria as open source. It can be found at <http://opensource.org/osd.html>. (Reprinted by kind permission of Eric S. Raymond.) We should also note that the effort to define "open

source" precisely continues to stimulate debate and revised definitions. This version will not satisfy everyone or remain unchanged.

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

2. Source Code

The program must include source code...

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code.

5. No Discrimination Against Persons or Groups.

6. No Discrimination Against Fields of Endeavor.

7. Distribution of License.

8. License Must Not Be Specific to a Product.

9. License Must Not Contaminate Other Software

Pitfalls of Open Source Development

Even though the open source model may seem like a rosy paradigm under which to develop projects for nonprofits, there are a few items that must be taken into consideration.

For every working open source project that exists, there are a dozen that have failed. The factors that lead to an operating OSS project as opposed to a dead one are complex and poorly understood. Some commonalities include how strong the initial code is, the role of leadership on the project, how interesting the coding challenge is to programmers and how well an information dissemination plan has been put into place.

A piece of open source software that is already useful, at least moderately developed and possessing clear goals for use is much more likely to be picked up by the greater community of programmers than is one which has poorly defined parameters or poorly written initial code. There is an attractiveness to being involved with a project that seems like it has a good chance at continued existence, which leads to an increasing returns of development. The better the project looks at the outset, the more coders become involved, the more likely the program is to be developed in a way that maximizes the benefits of OSS.

Several examples of currently successful open source software are nearly synonymous with their leadership. Linus Torvalds and Linux, Eric Raymond and Fetchmail, and Brian Behlendorf and Apache are all good examples of this phenomenon. Even projects where the leaders are not widely known are served by having a central administrator to keep focus on project scope and goals. An example is the Graphic Image Manipulation Program (GIMP), which was started by students, who then abandoned it due to an inability to put time into the project. The GIMP languished for a period of time, and did not resume development until someone actively took leadership of the project. Many open source sites spell out explicitly the norms for establishing leadership of similarly dormant projects in order to both encourage further development, yet discourage project splintering.

Another aspect of the value of leadership on these projects is the avoidance of software splintering. The early UNIX operating systems, while superior in many ways to the DOS operating system that later took precedence, were hindered by the splintering that occurred between all the different types of UNIX. In contrast, Linux has remained a single entity due to the leadership provided by Torvalds. Avoiding multiple and incompatible versions allows a project to gain market share, which is normally the advantage of software developed in a closed model. Part of the reason open source software has not had as much of the public's attention in the past was the relatively late development of the strong leadership model, partially in response to challenges posed by closed source developers like Microsoft.

The most difficult challenge for an open source project is to motivate unpaid programmers. Some motivations may include reputation, skills building and need for the software change they are working on. However, the problem becomes one of tasks that do not meet those requirements. Programming spreadsheets, writing API's and help manuals, or cranking out standard pieces of code are all unlikely to attract the large body of programmers necessary to realize the proposed benefits of open source software development. One solution used by Netscape in its Mozilla project is to have a combination approach, where the code is made available for people to work on as they will, while maintaining a corps of in-house programmers who take on tasks that have not been picked up by the larger community of hackers.

An important and often overlooked aspect of developing OSS is the need for a strong infrastructure to disseminate information about the project. Usually, a website is established that clearly defines how much of the software already exists, what the leader would like out of the community and where developers should send their submissions to the core software code. Once a project has been established, additional information may be passed through email distribution lists or Internet Relay Chat channels. The basic idea is to make sure that all people working on any particular project are aware of the overall progress being made, as well as some sense of the features on which their peers are working.

"New Philanthropy"

We are at a time of sharp change in many long-standing institutions of American society. Industry, government, and the nonprofit sector are all being deeply affected. We no longer have unquestioning confidence in the relatively bureaucratic ways of running organizations and businesses that were predominant and satisfactory for much of the century. In commerce we see clearly that a mighty upheaval is underway. But analogous changes are afoot also in the noncommercial parts of the society. Here too there are demands for activities that are more precisely targeted, for clearer indicators of desired effects, for initiatives structured to produce their own revenue streams rather than relying perennially on outside support, for faster results, for reduced micro-management, for a bolder attitude towards novel approaches and unconventional partnerships.

Social investors -- be they private foundations, volunteering citizens, or government bodies -- are all seeking new models for building social capacity, new forms of social investing that may bypass problems of earlier institutional forms.

There has been much publicity given to the tidal wave of wealth now being created in the United States. The numbers of family and community foundations in the U.S. have doubled in the past 15 years, and experts report there has been noticeable growth in foundations in and around Seattle and Silicon Valley. It is widely said that the generations amassing this wealth are impatient with the giving methods of their predecessors. There is no doubt some truth in this generational framing, as the baby boom generation has been impatient with each institution it has encountered, beginning with college, draft boards, and racial segregation in the sixties. And the even younger post-baby boom Internet entrepreneurs seem similarly impatient with their predecessors.

However, taking a wider view also reveals that the changing technological and economic landscape offers enormous new possibilities, no matter what generation one belongs to. Deep changes are occurring in "the rules" for making social investments, just as they are occurring in so many other parts of the society. The ASP and OSS ideas would not be as relevant for the

world of community-serving organizations if it were not a time when whole new models are welcomed for consideration.

We are free to imagine, for example, that nonprofits might get computational services from for-profits that target nonprofit clients. The Open Source Software model itself is an interesting hybrid that takes ideas of free software that have long been congenial to nonprofit organization and volunteerism, and combines them with novel notions of "business models" that declare it proper for substantial income to be generated from this freely accessible intellectual property.

Individual donors and corporations are making bold departures from traditional giving patterns. Internet sites are appearing to mediate giving and volunteering.. Suddenly wealthy individuals are devoting themselves to social investments and seeking new solutions to the age-old problem of giving effectively. Companies are exploring the use of foundations for giving activities that also reinforce corporate goals. Models are being tried with unprecedented combinations of for-profit and charitable incentives.

"New Philanthropy" is -thus part of the background for the Workshop because the movement under this label is bringing into focus a broad set of dramatic changes taking place in our long held conceptions of effective social investment. (Sources for this summary include various article on changes in social investment and a report entitled "e-Philanthropy, Volunteerism, and Social Changemaking" by the W.K. Kellogg Foundation, authored by Tom Reis and Stephanie Clohesy.)

Conclusions

As we mentioned at the outset, the intention of this document is to provide background on areas outside each participant's expertise. We know we have simplified matters in order to do this. But our hope is that we have provided initial bridges among the disparate communities that will be coming to Ann Arbor. For those who are keen to delve further, much more information is collected at the Workshop website.

If we have met our objective of building initial bridges, we will have laid the way for a productive meeting in which the diverse expertise of the participants may be woven into a compelling vision of new possibilities. We feel sure that there are new and better ways for rendering services in our communities that are waiting to be invented. There are possibilities that harness the potential of technology to the missions of organizations that have insight and commitment but have lacked the tools to amplify their energy. The Workshop brings together a group that can forge a vision of such possibilities and lay plans by which that vision can be made real.