Commercial Open Source RIS/PACS:
The ClearCanvas Experience

Norman Young[1], Clinton Chau[1] and John Adziovsky[2]

[1] ClearCanvas Inc., 620-438 Richmond St. W., Toronto, M5V 3S6, Canada

[2] The Joint Department of Medical Imaging, University Health Network, Mount Sinai Hospital and Women's College Hospital, 600 University Ave., Toronto, M5G 1X5, Canada

Abstract. ClearCanvas, an open source, healthcare informatics company is currently developing an open source RIS/PACS for a large Canadian hospital. The experience thus far has shown that an emphasis on application usability, as well as professional service and support are necessary for enhancing the credibility of open source healthcare solutions. Supporting such initiatives requires a solid business model that can generate the needed revenue. A model being explored draws on multiple revenue streams, including service contracts and value added features, from both the founding organization and commercial members of the ecosystem.

1 Introduction

ClearCanvas is a private, open source, healthcare informatics company based in Toronto, Canada. Its primary mandate is to develop an industrial strength, open source RIS/PACS (Radiology Information System/Picture Archiving and Communication System) solution for The Joint Department of Medical Imaging (JDMI), University Health Network, Mount Sinai Hospital, Women's College Hospital. JDMI is one of the largest medical imaging departments in Canada, with an annual volume of over 600,000 studies. The RIS/PACS consists of three main components:

- The PACS viewer allows DICOM (Digital Imaging and Communications in Medicine) images to be viewed by radiologists, referring physicians and other image consumers.

- The PACS server archives imaging studies for long term storage and serves them up to DICOM clients such as the PACS viewer.

- The RIS manages the entire workflow of the imaging department, from the point a patient checks in with the receptionist, to when the final report is issued to the referring physician.

Underlying all three products is a comprehensive application framework and SDK (software development kit) that allows new functionality to be quickly and easily added via a plugin mechanism.

2 Tools and Methodology

A combination of open source and commercial, off-the-shelf development tools have been employed over the course of this project. Because all products are built on Microsoft's .NET 3.0 platform, the primary development environment used is Microsoft Visual Studio. In the backend, Microsoft SQL Server serves as the database management system. Subversion is used for source control, and Trac for ticket and requirements management. Source code hosting is outsourced to an online service provider. SourceForge was not chosen for code hosting because it was foreseen that proprietary code branches would need to be created.

An agile software development methodology known as Scrum has been successfully employed in the development of all ClearCanvas products. Scrum is a form of iterative development, where at the beginning of an iteration or "sprint"—typically one month in length—the customer negotiates with the development team a set of requirements to be implemented. That backlog of items represents a contract: the development team agrees to implement all of the items in the backlog and the customer agrees not to insert any new items during the sprint. At the end of the sprint, the customer reviews the features implemented and determines whether they have been done satisfactorily. If not, the "failed" items are fed back into the next sprint. This methodology of continuously integrating customer feedback into the development process acknowledges the reality that software requirements often cannot be known with full specificity at project commencement and in fact, can continue to change well into the development cycle. This approach enhances the likelihood that the final product is something the customer is, in fact, happy with.

3 A Unique Relationship

At present, ClearCanvas essentially plays the role of a development consulting firm, as it has been contracted by JDMI to build a new RIS/PACS. It is important to note, however, that the relationship with JDMI is, in practice, more than merely contractual. For example, the entire company is co-located on hospital premises. As well, members of the hospital staff are integral members of the project team. As discussed later, this unique partnership has created the conditions necessary for the development of a system that is highly user-centric.

4 Insights Gained

The project is still very much a work in progress. As such, it is premature to comment on the efficacy of particular initiatives and strategic decisions. However,

insights gained thus far into what constitutes a commercially sustainable open source project can be offered.

4.1 Usability is Paramount

Many open source projects begin out of a developer's need for a tool or application of some kind that does not yet exist. In this scenario, the developer is often also the user. This is very helpful in enhancing the usability of the application, since the so-called "eat your own dogfood" dynamic is at work. However, with open source healthcare applications, this is typically not the case. The user is almost never the developer. And so, because the developer is not forced to use the software in the way a user would, ensuring the application's usability is particularly challenging.

This risk has been mitigated in a number of ways. First, project leads have all had prior experience designing RIS/PACS solutions. Such experience has not only been informative as to what user interface metaphors work, but also those that do not. Second, on the RIS project, the project manager is actually a member of the hospital staff who has had several years of hands-on experience with managing radiology workflow. Her intimate knowledge of not only the department's overall process, but also its process idiosyncrasies has been invaluable in helping the development team design a system that is comprehensive in its features and requirements. Third, the ClearCanvas office is co-located at JDMI. This enables convenient interaction with users, and also facilitates the company policy that all employees attend at least one tour of the clinical areas to understand how users interact with the existing RIS/PACS. Fourth, the software has been subjected to formal usability testing in the hospital's own human factors laboratory. This has allowed the project teams to see how users respond to the applications under controlled conditions, the results of which are then fed back into the development process.

4.2 Aesthetics Matter

An application's aesthetic appeal is important. Given two functionally equivalent systems, users are more likely to use and feel better about the more visually attractive one. As open source applications are sometimes criticized for not being as "professional" as their commercial counterparts, aesthetics become that much more important in enhancing their credibility. Attention to aesthetic details comes in many forms, such as clean and consistent user interfaces, pleasant colour schemes and well-designed icons.

4.3 Developers are Users Too

Usability is often thought of only from the perspective of the end user. But in an open source project, developers are users too, though in a different sense. Thought must be given to how to make it as easy as possible for external developers to participate in the project, or at the very least, how they can extend the application for their own purposes. It is not good enough that the code is open source. The code base needs to

be well-organized and easy to navigate. APIs (application programming interfaces) must be easy to use. The system architecture needs to be well documented so developers have the "big picture" in mind. Patterns and best practices need to be employed consistently.

With all these usability elements in mind, significant effort was invested into developing an application framework and SDK, with the goal of making common customizations easy, and more obscure customizations possible. The plugin-based framework provides developers a well-defined and well-understood method of integrating new functionality into the products without having to be familiar with the entire code base.

As with any open source project, there is significant variation in the skill level of the developers who use the SDK. Some are professional developers working for companies. Others are students and hobbyists. Although the developer documentation is fairly comprehensive, it has had to be supplemented with explicit code samples and an online discussion forum in which developers can ask questions.

4.4   Software is Not Enough. Service and Support is Key

The vast majority of hospitals and clinics in the North American healthcare IT market, especially the smaller ones, have little desire to support their own RIS/PACS. They want a system where they can call a single number to have problems addressed. For this reason, simply providing software, even if it is free (in the "free beer" sense) and open source is insufficient. To the customer, a service contract is tantamount to an insurance policy for when problems arise. For many, the lack of such support services automatically disqualifies the software, no matter how good it may be. It is a risk they simply cannot assume. From a revenue generation perspective, this is good news for open source healthcare IT companies, as it suggests that customers in fact want to pay for service and support.

It is worth noting, however, that this view of service contracts may be particular to developed nations. ClearCanvas' limited experience with institutions in developing countries suggests that they tend to be reluctant to pay for service contracts for financial reasons, and thus are much more willing to support the software themselves.

4.5   Open Source as a Basis for an Ecosystem

In the traditional proprietary software model, knowledge tends to be centralized in one place: the founding organization. The risks of this approach, such as vendor lock-in, are well known. In the open source model however, knowledge is decentralized. This is important, as it allows for people outside the founding organization to become experts. Such experts are then in a position not only to make code contributions, but can also go on to provide commercial services and/or other value-added features. This is essentially an ecosystem, of which both commercial and non-commercial entities can be a part.

Such an approach is not without precedent. DotNetNuke (DNN), a very successful open source web based Content Management System (CMS) has a thriving ecosystem that benefits from both open source and commercial contributions. Entire companies

based on the open source DNN platform have formed, offering everything from plugins, to website "skins", to training videos.

ClearCanvas is in the early stages of experimenting with bringing such an approach to the RIS/PACS domain. One can envision an ecosystem consisting of, for example, first tier service providers, professional service companies, custom development shops, etc. However, it is important to realize that RIS/PACS is a highly specialized market, whereas CM systems such as DNN have general market appeal and are not sector specific. Expectations need to be tempered by this important difference.

4.6 Open Source as a Platform for Value Added Features

One of the great strengths of DNN is its plugin architecture. It has allowed developers to extend the functionality of the base platform by making available to the community a multitude of plugins for a wide range of specialized purposes. What is notable about the DNN ecosystem is that there are both open source and commercial (closed source) plugins.

This "mixed source" model, while likely problematic for some open source purists, may hold the most promise in the RIS/PACS domain. The practical reality is that the types of vendors who would take advantage of a platform such as that offered by ClearCanvas, are most likely specialty companies (e.g. advanced visualization, computer aided diagnosis, etc.) whose competitive advantage depends on being able to protect their intellectual property through closed source plugins. Releasing platform software under a viral open source license would likely stifle the growth of a commercial ecosystem, as vendors would be reluctant to open source their plugin code. For this reason, ClearCanvas chose to release all of its products under the permissive New BSD license, so as to encourage plugin development by commercial vendors. Doing so also allows ClearCanvas to add value to its own commodity system through specialized commercial plugins for say, speech recognition, enterprise management, data mining, etc.

This approach can be taken one step further by encouraging plugin vendors to sell their plugins via e-commerce on the ClearCanvas website. By taking a percentage of the sales, another revenue stream can be developed. For the vendor, there is incentive to sell on the ClearCanvas site because the availability of a free RIS/PACS naturally generates web traffic, thereby increasing sales. Plugin vendors could also conceivably buy advertising or product information pages on the site to promote their products. The open source platform then becomes a means to generate revenue.

5 Conclusions

Factors such as usability, aesthetics, developer friendliness and the availability of service and support are necessary for open source healthcare solutions to gain acceptance. They are important in part because open source software still has a credibility problem in the healthcare sector. It is still seen as "unreliable" and "risky", both technically, and in terms of how problems are resolved when things go wrong with a live system. Overcoming this image problem through what essentially amounts

to elevated professionalism has a cost. For this reason, solid business models must exist. In developed countries, a significant source of revenue can likely be found in service contracts, as healthcare institutions view them as being essential. If an open source system is architected effectively, it can also be a platform for revenue generating, value added features, which can be developed by the founding organization, or by commercial members of the ecosystem.