

Quando o Bazar não Funciona: Custo total de "propriedade" no desenvolvimento de um sistema complexo em software livre, o GNUTECA

Eloni José Salvi, Cesar Brod, Pablo Dall'Oglio, João Alex Fritsch, Paulo Cattelan, Thomas Sprietersbach

Univates, Centro Universitário
Rua Avelino Talini, 171 – Lajeado – RS – Brazil

{esalvi, cesar, pablo, joaoalex}@univates.br, cattelan@control.com.br,
ts@interact2000.com.br

***Abstract.** Eric S. Raymond described in his article "The Cathedral and the Bazaar" the development style of the Linux kernel, calling it the Bazaar style: an anarchic form of software development that congregates thousands of independent developers among a common goal. Several free software projects, usually when involving programs needed by many people, have succeeded using the Bazaar style. What if you need a system that is not needed by many people? Actually, what if you want to develop a system that will address very specific needs: is it worth to do it using free software and releasing this system also as free software? In this paper we will describe the development of GNUTECA, a free-software system for library administration that handles today more than 60.000 individual materials being used by more than 5.000 people at our University Centre, and is being adopted and evaluated by several other Libraries.*

***Resumo.** Eric S. Raymond descreveu em seu artigo "The Cathedral and the Bazaar" o estilo de desenvolvimento do Kernel Linux, chamando-o estilo Bazaar: uma forma anárquica de desenvolvimento de software que congrega milhares de desenvolvedores em torno de um objetivo comum. Muitos projetos em software livre, normalmente os que envolvem programas que são necessários a muitas pessoas, obtiveram sucesso usando o estilo Bazaar. Mas se você precisa de um sistema que não é necessário para muitos? E se o sistema que você deseja desenvolver irá atender a requisitos muito específicos: vale a pena desenvolvê-lo e distribuí-lo como software livre? Neste artigo descreveremos o desenvolvimento do GNUTECA, sistema em software livre para a administração de bibliotecas que hoje trabalha com mais de 60.000 materiais individuais usados por mais de 5.000 pessoas em nosso Centro Universitário, e está sendo adotado e avaliado por várias outras Bibliotecas.*

1. Introdução

Muitos dos produtos que conhecemos hoje em software livre foram desenvolvidos de uma forma aparentemente anárquica, ou melhor, com uma organização peculiar que nasceu de uma anarquia original. Eric S. Raymond descreve muito bem este processo em seu artigo "The Cathedral and the Bazaar", usando como exemplo o desenvolvimento do kernel Linux. O estilo "Bazaar" parte do princípio de que virtualmente qualquer um pode contribuir para o desenvolvimento de um programa, e que as contribuições serão filtradas de forma "darwiniana": o melhor código sobreviverá. Problemas serão facilmente eliminados, uma vez que "com muitos olhos, todos os 'bugs' serão visíveis".

O "estilo Bazaar" funciona muito bem com softwares que muitas pessoas necessitam, para os quais é relativamente fácil encontrar um bom número de desenvolvedores voluntários que se integram a um projeto cujo principal centro de discussões e definições são listas de mensagens na Internet. É comum observarmos uma grande rotatividade de desenvolvedores nestes projetos, uma vez que não há nenhuma espécie de compromisso formal com o

desenvolvimento do software. Quando é preciso desenvolver um sistema que será utilizado por um número limitado de pessoas, dentro de prazos que precisam ser cumpridos à risca, e respeitando um orçamento, é virtualmente impossível utilizar o "estilo Bazaar".

Mas por que desenvolver em software livre um sistema que terá uma audiência limitada? A principal razão é o compromisso com o livre acesso ao conhecimento: novas idéias podem ser desenvolvidas em grande parte porque existe um enorme arsenal de idéias antigas que podem ser livremente utilizadas, e esperamos modestamente que, mantendo livre o conhecimento que produzimos, estaremos contribuindo para que idéias novas, originais e livres continuem a surgir. A razão prática que faz com que o desenvolvimento de um sistema em software livre seja viável, porém, é que seu custo total de propriedade¹ é menor, se comparado com a aquisição ou o desenvolvimento de um sistema proprietário equivalente. Este custo de propriedade pode reduzir-se ainda mais se mais instituições de características similares tornam-se parceiras no desenvolvimento e posterior suporte ao sistema, como veremos adiante.

Na UNIVATES percebemos com o desenvolvimento do SAGU (Sistema Aberto de Gestão Unificada), responsável pela automação dos processos de relacionamento dos alunos com a instituição de ensino através de seus vários módulos (vestibular, acadêmico, financeiro, e outros), que muitas ferramentas de software livre já disponíveis prestam-se muito bem ao desenvolvimento de sistemas complexos, e outras que ainda não existem podem ser criadas tomando por base as que já existem. O SAGU foi desenvolvido de forma bastante rápida, em função da urgente necessidade de substituição do sistema administrativo existente, desenvolvido em software proprietário e que não suportaria a carga imposta pela matrícula de novos alunos no segundo semestre do ano 2000. Um dos fatores determinantes para a opção pelo desenvolvimento do SAGU foi o fato de que nossa instituição não possuía recursos para investir em uma opção proprietária comercial. A necessidade de rápido desenvolvimento do SAGU não nos permitiu a aplicação de medidas que comprovassem de forma eficaz a percepção que tínhamos de que tal desenvolvimento deu-se de maneira eficaz e econômica. Ao invés de coletarmos dados sobre o que já havia sido feito, preferimos confiar na percepção de que estávamos no caminho certo e fazer um estudo de custo total de propriedade em um novo sistema que desenvolveríamos, utilizando a experiência no desenvolvimento de sistemas anteriores, métodos clássicos e outros nem tanto. O sistema escolhido foi o GNUTECA.

2. O GNUTECA

O GNUTECA é um sistema em software livre composto de módulos para gerir acervos bibliográficos, controlar empréstimos, pesquisar em bases bibliográficas e administrar o sistema de forma local e remota, promovendo maior agilidade e qualidade aos serviços prestados por bibliotecas, assim como prover o fácil intercâmbio de informações com outros sistemas de Bibliotecas, mantendo-se dentro de normas internacionais ISO e do padrão internacional de catalogação MARC 21. É um sistema abrangente e genérico que pode moldar-se a diferentes realidades de diferentes usuários, permitindo também a criação de uma infra-estrutura de colaboração entre bibliotecários e demais funcionários das bibliotecas, evitando a repetição desnecessária de trabalho: uma vez feita a catalogação de um título em uma biblioteca, estes dados catalográficos podem ser "importados" para o sistema de outra biblioteca que adquira o mesmo título.

3. Metodologia de Desenvolvimento

O desenvolvimento do SAGU e do GNUTDATA na UNIVATES, a experiência anterior de cada um de nossos desenvolvedores e a troca de experiências na participação em projetos

1 Utilizamos o termo *Custo Total de Propriedade* como referência a uma metodologia conhecida de determinação do gasto total feito com a aquisição de alguma coisa, considerados todos os fatores que influem neste custo. Em software livre, porém, não existe a "propriedade" sobre o sistema, e um termo substituto seria Custo Total de Viabilidade e Operacionalização do Sistema. Optamos pela utilização do termo já consagrado pelo uso, fazendo a devida ressalva com relação à propriedade.

cooperativos de software livre e código aberto (PHP, PHP-GTK e SourceForge) já nos levaram a padronização cada vez maior do código que escrevíamos.

Com a integração de novas pessoas ao nosso CPD, e já com alguma perspectiva de agregar desenvolvedores externos a nossos projetos, decidimos adotar alguma metodologia consagrada que nos permitisse documentar nossos sistemas, facilitando a interação com os usuários (clientes) e a comunicação de necessidades e resultados ao público externo. Optamos por utilizar a UML (Unified Modeling Language) como padrão de desenvolvimento e algumas características de Extreme Programming (XP)² como padrão de atitude. Estas metodologias possuem ampla documentação disponível, por isto, ao invés de descrevê-las, vamos ilustrar diretamente sua aplicação no desenvolvimento do GNUTECA.

Antes de iniciar o projeto, realizamos uma oficina prática de UML e orientação à objetos para trazer à nossa equipe a experiência de profissionais³ da área de análise e projeto de sistemas. Esta oficina acabou por definir muitos dos conceitos que seriam aplicados ao GNUTECA.

Quem deve definir o que o sistema fará é quem irá utilizá-lo. Assim, o GNUTECA iniciou-se com o levantamento de todos os processos utilizados em uma biblioteca, desde a forma de catalogação de acervo e seus padrões até a interação com o usuário no balcão de atendimento, passando por consultas ao acervo, relatórios e o relacionamento com outros sistemas. Tudo isto foi documentado em "casos de uso", "diagramas de seqüência", "diagramas de classe" e "diagramas de atividade". Além de envolvermos o cliente interno (nossas bibliotecárias e funcionários) no processo, contamos com a consultoria da Control e com a sugestão de vários bibliotecários que começaram a contribuir com sugestões a partir da apresentação do projeto conceitual em 13 de junho de 2001. O código-fonte em produção foi sempre disponibilizado prematuramente no site do projeto, a fim de que a comunidade pudesse acompanhar sua evolução, ainda que em estágios não funcionais.

4. Ferramentas

Para o desenvolvimento do módulo Web de interação com o aluno (pesquisas, renovações, reservas), utilizamos o MIOLO⁴, que é um padrão e uma ferramenta de desenvolvimento de projetos em PHP orientado a objeto, com bases de dados distribuídas, voltados à Internet. O MIOLO possui objetos de conexão à base de dados e lógica de apresentação, permitindo que os desenvolvedores se concentrem, basicamente, apenas na lógica do negócio. Para os módulos usados no balcão de atendimento escolhemos o PHP-GTK⁵.

O GNUTECA, entrou em fase de testes no campus da UNIVATES da cidade de Encantado em Outubro de 2001. Nessa fase foram realizados diversos ajustes ergonômicos e operacionais do sistema de acordo com solicitações dos nossos usuários. Em 25 de fevereiro de 2002 o sistema entrou em produção no campus central da UNIVATES atendendo a mais de 5 mil usuários com um acervo de mais de 50 mil exemplares. As funcionalidades diárias mais exigidas foram cumpridas com rigor. A próxima etapa do projeto será a implementação de diversos relatórios gerenciais.

5. Custo total de propriedade

2 <http://sagu.codigolivre.org.br/article.php?sid=50>

3 Deixamos aqui nosso agradecimento a Alessandro Binhara, da TECPAR (Curitiba, PR) e Leandro Pompermayer da HP (Porto Alegre, RS).

4 <http://miolo.codigolivre.org.br>

5 <http://gtk.php.net> – O sistema de atendimento em balcão do GNUTECA é o maior projeto já escrito em PHP-GTK

Caso a UNIVATES tivesse optado pela aquisição de um sistema proprietário ao invés do GNUTECA, ainda assim ocorreriam gastos com hardware. Simplificaremos aqui nossa análise de custo desconsiderando os gastos com hardware, mesmo que nossa experiência tenha comprovado que com a utilização de software livre temos conseguido trabalhar com equipamentos de menor porte e com grande índice de reaproveitamento. Alguns quiosques de nossa biblioteca, por exemplo, são antigos Pcs 486, que rodam um browser em modo gráfico para a consulta no acervo.

<i>Descrição</i>	<i>Quantidade</i>	<i>Valor Total</i>
Horas de desenvolvedores (Univates)	2000	R\$ 20.000,00
Horas de gestão de projeto	200	R\$ 6.000,00
Consultoria interna (bibliotecárias)	200	R\$ 3.000,00
Custo administrativo (20% dos custos acima)		R\$ 5.800,00
Consultoria externa	48	R\$ 4.800,00
Treinamento		R\$ 1.000,00
Bibliografia		R\$ 1.000,00
Total		R\$ 41.600,00

Pudemos contar com a cessão de horas de pessoal da empresa Control e da Interact2000, parceiras da UNIVATES no desenvolvimento do GNUTECA. A tabela abaixo, entretanto, ilustra a estimativa de custos em serviços, caso tivéssemos pago efetivamente por todos eles. Os valores usados têm por base tabelas de referência de custos internos e valores médios de contratação de serviços de terceiros.

Existem outros custos difíceis de ser estimados, como as modificações que foram necessárias ao MIOLO, mas que acabaram sendo absorvidas também por outros projetos. Nossa experiência mostra que a manutenção do produto em produção deve consumir no máximo 40 horas mensais (custo interno) por um período de um ano. Assim, é seguro dizer o TCO do desenvolvimento do GNUTECA, para um período de dois anos, está na faixa de R\$ 50.000,00. Como não há a necessidade de aquisição de nenhum produto proprietário, e não há o aumento em custo de software independente do número de usuários, o GNUTECA é um produto bastante competitivo ainda que seu preço de implantação subsequente fosse igual a seu TCO.

6. Conclusão

A escolha pelo desenvolvimento de um sistema complexo em software livre é viável quando comparada à aquisição de um produto proprietário equivalente, mesmo quando apenas uma instituição arca com este custo. Quanto maior o número de instituições envolvidas na produção de sistemas que possam ser compartilhados, menor o custo total de desenvolvimento e suporte, e maior a velocidade de produção (com mais pessoas envolvidas).

7. Referências

- FURLAN, José Davi. *Modelagem de objetos através da UML – the Unified Modeling Language*. São Paulo, Makron Books, 1996.
- FOWLER, Martin & KENDALL, Scott. *UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos*. Porto Alegre, Bookman, 2000.
- MARC 21: *Formato condensado para dados bibliográficos*. Marília, UNESP – Marília Publicações, 2000
- RAYMOND, Eric Stefen. *The Cathedral and the Bazaar*.
<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>