

ASPECTOS ESTRATÉGICOS E DE CUSTOS RELACIONADOS À MIGRAÇÃO DE SOFTWARE PROPRIETÁRIO PARA SOFTWARE LIVRE

Nicolas Michel Bacic

Luiz Antônio Teixeira Vasconcelos

Resumo:

Existe uma oferta abundante e de qualidade de software de baixo custo ou gratuito - o software livre. Porém organizações públicas e privadas e usuários finais, têm dificuldade de abandonar modalidades de softwares pagos e migrar para o software livre. Quais são os fatores que dificultam esta migração? Que fatores podem ajudar na migração? Este estudo visa identificar e explicar os principais condicionantes da expansão do software livre, avaliando os fatores que dificultam e os que favorecem sua adoção. Para responder a essas questões propõe-se uma abordagem interdisciplinar, estudando, a partir do campo da economia, o conceito de aprisionamento tecnológico, e, a partir do campo da computação, os aspectos técnicos e de qualidade de software, que possibilitem o entendimento, tanto da natureza técnica dos aprisionamentos, como dos esforços que os desenvolvedores de software livre estão fazendo para diminuir o custo que as empresas, organizações e pessoas, tem para sair da situação de aprisionamento. Finalmente, propõe-se um modelo de custos para avaliar a oportunidade de adotar o software livre.

Palavras-chave:

Área temática: *Gestão de Custos e Tecnologia da Informação*

ASPECTOS ESTRATÉGICOS E DE CUSTOS RELACIONADOS À MIGRAÇÃO DE SOFTWARE PROPRIETÁRIO PARA SOFTWARE LIVRE

Nicolas Michel Baic
Faculdades de Campinas
nicolas.baic@uol.com.br

Luiz Antônio Teixeira Vasconcelos
Universidade Estadual de Campinas

Existe uma oferta abundante e de qualidade de software de baixo custo ou gratuito - o software livre. Porém organizações públicas e privadas e usuários finais, têm dificuldade de abandonar modalidades de softwares pagos e migrar para o software livre. Quais são os fatores que dificultam esta migração? Que fatores podem ajudar na migração? Este estudo visa identificar e explicar os principais condicionantes da expansão do software livre, avaliando os fatores que dificultam e os que favorecem sua adoção. Para responder a essas questões propõe-se uma abordagem interdisciplinar, estudando, a partir do campo da economia, o conceito de aprisionamento tecnológico, e, a partir do campo da computação, os aspectos técnicos e de qualidade de software, que possibilitem o entendimento, tanto da natureza técnica dos aprisionamentos, como dos esforços que os desenvolvedores de software livre estão fazendo para diminuir o custo que as empresas, organizações e pessoas, tem para sair da situação de aprisionamento. Finalmente, propõe-se um modelo de custos para avaliar a oportunidade de adotar o software livre.

Área Temática: Gestão de Custos e Tecnologia da Informação

ASPECTOS ESTRATÉGICOS E DE CUSTOS RELACIONADOS À MIGRAÇÃO DE SOFTWARE PROPRIETÁRIO PARA SOFTWARE LIVRE

1- Introdução

Segundo pesquisa realizada pela Sociedade para Promoção da Excelência do Software Brasileiro (SOFTEX) em conjunto com o Massachusetts Institute of Technology (MITI), o Brasil é o sétimo mercado de software no mundo, com vendas de US\$ 7,7 bilhões em 2001, rivalizando com a Índia e a China, com respectivamente US\$ 7,9 e US\$ 8,2 bilhões. Entre 1991 e 2001, a participação do segmento no PIB triplicou, alcançando 0,71%. (MITI- SOFTEX, 2002).

O setor de produção de software tem características diferenciadoras, no que se refere ao comportamento do ciclo de produto (1), em relação a outros setores. Na maioria dos setores, o peso dos investimentos na etapa correspondente ao ciclo operacional (aquisição dos insumos e produção dos produtos) é importante se comparado com o valor dos investimentos realizados nas outras etapas do ciclo do produto (planejamento e desenvolvimento do produto). No setor de software, conforme Vasconcelos (1999) as etapas de planejamento e desenvolvimento são bem mais importantes, em termo de investimento de recursos que a etapa produtiva (entendendo esta etapa, como a geração da cópia do produto a ser vendido): “quando se atinge o auge das vendas de um determinado produto, os gastos de produção característicos - insumos materiais, mão de obra direta, processamento, etc. - são, em geral, residuais em relação ao total de gastos de planejamento e desenvolvimento do produto em questão”. (Vasconcelos, 1999). Este fato coloca importantes desafios para as empresas de software, quanto às estratégias a serem seguidas para a gestão de seus negócios, dado o peso importante dos investimentos iniciais em relação aos investimentos da etapa produtiva.

Outra característica diferenciadora é observada pelo lado da demanda. Os compradores do software incorporam o produto a seus processos e rotinas e passam, em muitos casos, a ter uma situação de forte dependência em relação ao software escolhido. Esta situação se dá por diversos aspectos, por exemplo, em função do aprendizado, ou por problemas de compatibilidade que devem ser mantidos entre distintos softwares que trabalham trocando informações. Neste caso, tendem a ficar “presos” ao fornecedor inicial e tem um custo significativo (comparativamente a outros produtos) para mudar de fornecedor, dado que devem superar um conjunto de custos relativos ao aprendizado do uso do novo software e a superação dos problemas de compatibilidade. Esta situação de aprisionamento, tende a gerar vantagens para as empresas melhor sucedidas, que passam a usufruir de uma situação de quase monopólio, com os usuários sendo induzidos a acompanhar suas modificações de produto, sem terem condições de trocar por outros produtos, oferecidos por outras empresas.

Verifica-se assim, um interessante paradoxo, existindo uma oferta abundante e de qualidade de software de baixo custo ou gratuito - o software livre- empresas, organizações públicas e privadas e usuários finais, têm dificuldade de abandonar modalidades de softwares pagos e migrar para o software livre. Quais são os fatores que dificultam esta migração? Que fatores podem ajudar na migração?

Este estudo visa identificar e explicar os fatores que condicionam a expansão do software livre, avaliando os fatores que dificultam e os que favorecem sua adoção. Finalmente propõe um modelo de custos para avaliar a oportunidade de adotar o software livre.

2 - Justificativa

Hoje qualquer que seja a instituição pública ou privada necessita utilizar programas de computadores para auxiliar os seus processos, gerando uma enorme demanda por softwares. No entanto, a aquisição deste programas não é barata, tornando-se inviável a aquisição ou a atualização do software para diversas organizações. Cabe observar que o software é um componente de peso no custo dos computadores e, por exemplo, a implantação ampla de computadores em toda a rede de escolas públicas no Brasil demandaria um alto investimento em software.

Como o software, freqüentemente, é fundamental e indispensável para o desenvolvimento dos processos – básicos ou não – executados e/ou administrados pelas organizações, estas podem acabar decidindo utilizar o programa mesmo sem obter sua devida licença, incorrendo assim na pirataria de software, que é uma prática ilegal. Com a intensificação do combate a pirataria, esta prática pode sair cara, como mostra a Lei de Software, Lei nº 9.609, de 19 de fevereiro de 1998. Assim o artigo Art. 12º da lei enuncia que quem violar direitos de autor de programa de computador terá pena de detenção de seis meses a dois anos ou multa e se a violação consistir na reprodução, por qualquer meio, de programa de computador, no todo ou em parte, para fins de comércio, sem autorização expressa do autor ou de quem o represente a pena será de reclusão de um a quatro anos e multa.

Para não incorrer nesta prática ilegal, as organizações tem dois caminhos. O primeiro consiste em pagar as devidas licenças de uso. O segundo caminho consiste em buscar no software livre uma alternativa de qualidade e de baixo custo.

A opção pelo software livre se justificaria pelas seguintes e principais razões:

- Ser considerado, por especialistas, de qualidade igual ou superior aos softwares pagos.
- Poder ser instalado em quantas máquinas forem necessárias sem ter pagar pela licença.
- Permitir acesso ao código fonte, o que possibilitaria realizar programas de melhor qualidade e realizar alterações que visem adequar melhor o programa as necessidades do usuário.
- Oferecer grande variedade de programas em quase todos os campos de aplicação de Softwares.

Por outro lado, a utilização do software livre pelas instituições poderia permitir uma melhora no Balanço de Pagamentos, já que os gastos com os softwares proprietários, via aquisição de licenças de uso, resultam, segundo o Banco Central (2003), em uma saída na ordem de US\$ 1,12 bilhão, anuais. Ainda, a utilização dessa categoria de software pelas instituições públicas permitiria uma melhora nas contas públicas, possibilitando, por exemplo, nas escolas públicas, que fossem utilizados os recursos não gastos em Software para a ampliação do número de computadores disponíveis aos estudantes e professores, que utilizariam softwares livres. Evidentemente, que se a decisão é tomada num momento em que o parque

computacional instalado é pequeno, os eventuais custos de transição serão menores que quando se trate de mudar um parque amplo e já instalado.

O Governo Federal entende que o software livre é uma “solução robusta, com alto grau de portabilidade, permitindo a integração com soluções já existentes e com software comercial. Sua adoção na Administração Pública Federal pode representar uma significativa redução de custos, além de possibilitar independência na seleção de fornecedores de hardware, software e serviços, proporcionando maior flexibilidade na definição de estratégias na área de TI” (MCT, 1ª. Oficina de Software Livre, 2003).

3 – Software Proprietário e Livre

O conceito de software livre foi desenvolvido, inicialmente, por Richard Stallman no início dos anos 80. Tudo começou quando Richard trabalhava no Laboratório de Inteligência Artificial do MIT e a impressora que ele utilizava quebrou, sendo necessário adquirir uma nova impressora (Hauben, 2003).

Para continuar a realizar seu trabalho precisava conhecer o funcionamento desta nova impressora, no entanto suas solicitações ao fabricante para que este liberasse os códigos fontes do equipamento foram negadas. A partir deste momento Stallman passou a pensar em uma forma de tornar acessíveis os programas e códigos, surgindo a idéia de software livre.

A filosofia do software livre está baseada em quatro conceitos básicos:

- A liberdade de executar o programa, para qualquer propósito.
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. O acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo a ampliar as possibilidades de acesso de pessoas e instituições a tais programas.
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie, sem gastos adicionais.

Esses conceitos buscam garantir que o usuário possa executar, copiar, estudar, modificar o software, visando sempre à liberdade de produção (ou aperfeiçoamento) e utilização. Software livre pressupõe a liberdade de utilização, por isso é melhor evitar termos como “de graça” ou “doador”, pois não se trata de uma questão de preço, mas sim de liberdade de acesso ao código-fonte e de uso. Um usuário pode redistribuir as cópias do software de maneira gratuita ou cobrando uma taxa pela distribuição, as pessoas possuem livre escolha para decidir como distribuir um programa, alterá-lo, etc., sem ter que pedir permissão a ninguém. Cabe observar que esta possibilidade de cobrança é marginal, não havendo obrigação de pagar as licenças de uso, assim sendo ao comprar um programa de um usuário que cobre pelo serviço de distribuição, o comprador passa a ter direitos de realizar o que quiser com o software, podendo alterá-lo e instalá-lo em outras máquinas sem que se tenha que pagar nada para ninguém.(Bacic, N., 2003).

Já o software proprietário é regido por uma série de normas que visam limitar o seu uso ao número de licenças adquiridas. São estabelecidas licenças pelas quais é necessário pagar por cópia instalada, não sendo permitida a alteração do código;

ademais, é ilícita a livre distribuição do programa e normalmente não se tem acesso ao código fonte destes programas.

Como a grande maioria dos softwares proprietários não possui o código aberto não é possível personalizar o programa, verificar a qualidade do código, realizar melhorias no programa e corrigir erros.

Os softwares livres mais conhecidos são: o sistema operacional LINUX, o pacote de ferramentas para escritório OpenOffice, o editor de imagem GIMP, o navegador de Internet Mozilla, etc.

O fabricante de softwares proprietários mais conhecido é a Microsoft com programas como o Windows 95/98/NT/2000/XP, o pacote Microsoft Office, o navegador Internet Explorer, etc.

4 - Aprisionamento Tecnológico

Ao optar por uma determinada tecnologia, uma organização se submete a determinadas restrições que as levam a ficar dependente da tecnologia escolhida, devido à dificuldade de troca da dessa tecnologia por uma outra. Este processo é denominado de aprisionamento tecnológico.

Estas dificuldades de mudança decorrem das incompatibilidades entre as tecnologias, que as levam a altos custos de mudanças. Por exemplo, um usuário de tecnologia de informação (TI) ao escolher um determinado software de computador poderá ter problemas ao tentar migrar para outro programa, devido à incompatibilidade entre os tipos de arquivos, impossibilidade de migrar diretamente os dados, necessidade de treinamento ao novo software, etc.

Assim, antes de mudar de um sistema ou tecnologia, é necessário levar em consideração que, normalmente, os custos de troca são significativos. Isto, muitas vezes, leva aos responsáveis pelo gerenciamento da informação dentro das empresas a optarem por continuar utilizando o sistema vigente, ou seja, leva a um aprisionamento aos sistemas já instalados.

Nem sempre o aprisionamento prevalece, existem determinados momentos nos quais a tecnologia torna-se obsoleta e precisa ser substituída. No entanto, os custos de mudança interferem nas estratégias e opções as empresas.

Para lidar melhor com o aprisionamento é necessário reconhecer seus diversos tipos e os custos de troca nos quais implicam, assim como seu ciclo. Os tipos de aprisionamento e custos de troca associados são mostrados na tabela apresentada na página seguinte.

Reconhecer esses padrões ajuda a identificar e antecipar o aprisionamento, avaliar os custos de troca e planejar uma estratégia para os efeitos deste processo.

Cabe observar que os padrões levantados por Shapiro e Varian, não se aplicam unicamente ao caso da TI e são válidos para um amplo espectro de situações. Porém, no caso de produtos baseados na TI é mais comum o surgimento de custos de mudança que levam a situações de aprisionamento.

Alem desses fatores de aprisionamento mencionados por Shapiro e Varian, cabe observar que existem outros dois tipos de aprisionamento, que mesmo que não mencionados pela literatura são bem conhecidos pelos profissionais da área de TI: a

incompatibilidade de sistemas e o comprometimento da cadeia em favor de determinado padrão.

Tabela 1 – Tipos de aprisionamento e custos de troca associados

Tipo de aprisionamento	Custos de mudança
Compromissos contratuais	Indenizações compensatórias ou liquidadas.
Compra de bens duráveis	O custo de substituição de equipamento tende a cair à medida que o bem durável envelhece.
Treinamento em marca específica	Aprender um novo sistema demanda tempo e incorre em custos, que tendem a aumentar com o tempo.
Informação e banco de dados	Conservação de dados para o novo formato. O custo tende a aumentar com o tempo, pois a quantidade de dados aumenta.
Fornecedores especializados	Financiamento de novo fornecedor -tende a ser maior quanto mais difícil for encontrar um novo fornecedor.
Custos de busca	Custos combinados do comprador e fornecedor – incluem o aprendizado sobre a qualidade das alternativas.
Programas de lealdade	Quaisquer benefícios perdidos do fornecedor, mais possível necessidade de reconstruir o uso cumulativo.

Fonte: SHAPIRO e VARIAN (1999)

Incompatibilidade de sistemas

Na área de TI é necessário que sistemas distintos “conversem”, ou seja, que possa haver transferência e leitura de dados de forma simples e eficiente entre ambos. A mudança de um elemento não pode ser realizada sem estudar seu impacto nos demais componentes do sistema. No caso dos softwares, a escolha de determinado ambiente operacional (p. ex. Windows) leva a comprar e desenvolver programas que são desenvolvidos tendo esse ambiente como pressuposto de programação. Mudar o ambiente significa realizar pesados esforços para readequar todos os programas ao novo ambiente.

No caso de uma mudança de sistema operacional é necessário não apenas garantir o funcionamento de todas as aplicações para o novo ambiente, mas também o funcionamento de todos os hardwares utilizados pelo sistema.

Comprometimento da cadeia com determinado padrão

Alguns produtos precisam de uma cadeia de apoiadores para seu sucesso. Assim, grande parte do sucesso da Matsuchita com o padrão VHS, deve-se a sua estratégia de licenciar sua tecnologia para outros concorrentes, o que convenceu os produtores de filmes, os distribuidores e os lojistas a apostar na expansão do padrão, deslocando o Betamax da Sony.

Na produção de *softwares*, as *soft-houses* tendem a adquirir os programas (meios de produção) que sejam compatíveis com o padrão dominante, ou seja, se esse padrão é o Windows, tenderão a adquirir programas que trabalhem com esse ambiente e produzirão programas (produtos finais) que vão reproduzir tal

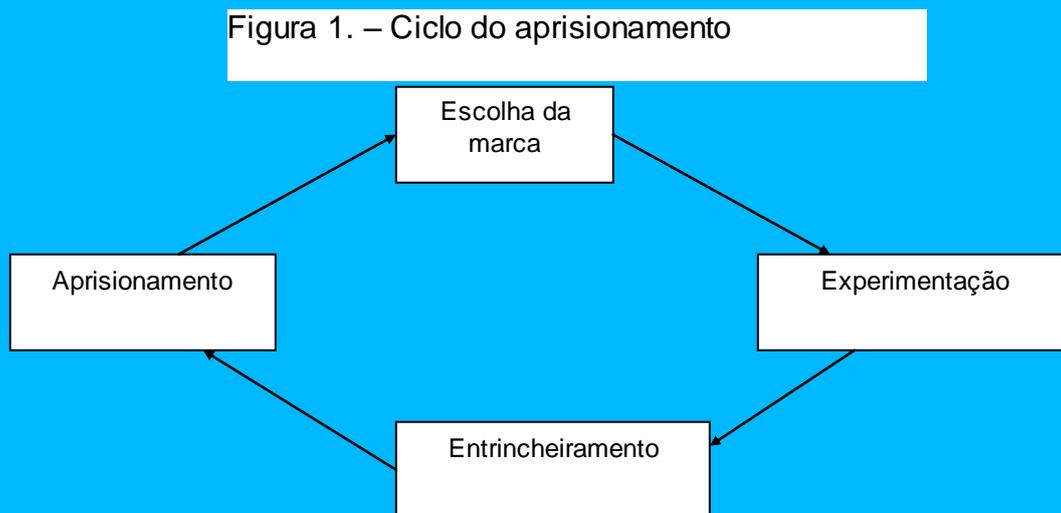
necessidade de padronização. Os programadores terão habilidade no ambiente Windows. Assim, a oferta de produtos, passa a ser condicionada pelo padrão dominante, qualquer mudança exige alterações tecnológicas tanto no produtor como no cliente.

5 - O Ciclo do Aprisionamento

O ciclo do aprisionamento inicia-se no momento da seleção de uma marca. Quando a escolha é a primeira que se realiza, não existe preferência por uma marca, mas após optar pela primeira marca incorre-se no aprisionamento, não ficando tão livre para escolher a próxima marca.

Após a escolha da marca, entra-se na fase de experimentação, na qual o usuário estará testando a marca e usufruindo as vantagens e incentivos que receber da mesma.

Posteriormente entra-se na fase do entrincheiramento, que ocorre quando o usuário acostuma-se a nova marca, dando preferência a ela em relação às outras marcas. Quanto mais tempo durar esta fase, maiores serão os custos de troca, levando a fase do aprisionamento, onde os custos de mudança tornam-se altos demais.



Fonte: SHAPIRO e VARIAN (1999)

Volta-se então a escolha da marca, que não mais ocorrerá livremente, uma vez que podem existir contratos com os fornecedores, os usuários estão acostumados com a marca utilizada, os arquivos de dados estão em determinado formato, etc. Assim, os custos de troca tornam-se mais altos, dificultando a migração para outras novas tecnologias que possam ter surgido.

Portanto para uma empresa diminuir seu aprisionamento frente a determinado sistema de informação, torna-se necessário que ela consiga antecipar vários ciclos no futuro ao traçar determinada estratégia, procurando ficar o mínimo possível dependente de determinada tecnologia.

6. Efeitos do Feedback

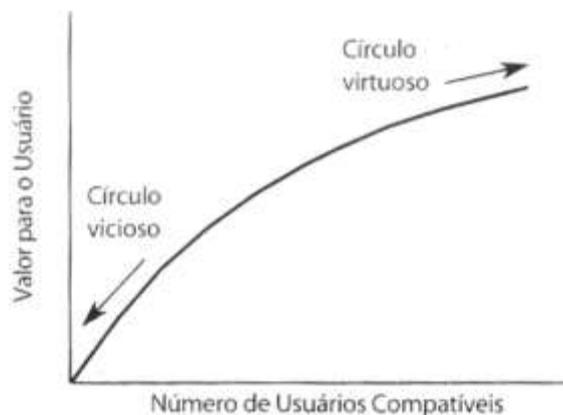
Quando um sistema se beneficia do maior número de usuários que o utilizam, fazendo com que os novos usuários também optem por utilizar o mesmo sistema, temos o chamado feedback positivo. Sistemas que apresentam esse padrão alcançam maiores vantagens quanto maior for à base instalada de usuários de um determinado sistema, pois maior será o número de novos usuários que optaram por utilizar este sistema.

Esse efeito pode ser facilmente visto a partir da observação do uso do Sistema Operacional Microsoft Windows, para o qual, tendo em vista sua enorme base de usuários, acaba se impondo para as pessoas que queiram aprender informática, que procurem um curso que ensine ou ofereça treinamento para o uso deste sistema operacional, assim como, e por decorrência, uma nova empresa tenderá a optar por utilizar este sistema devido a maior facilidade de encontrar pessoas que o saibam operar. Além disso, a popularização do sistema operacional estimulou a produção de um maior número de programas e *drivers* para o sistema, fazendo com o Windows ficasse mais atraente para novos clientes.

No entanto, o feedback não age somente de forma favorável, podendo acelerar o fracasso de determinado produto, caso o mercado não entenda o produto como promissor. Assim, o produto entra no chamado círculo vicioso do feedback ou feedback negativo.

Como exemplo do círculo vicioso, se pode pegar o caso da Netscape, onde a perda de usuários para o Internet Explorer levou ao fechamento do grupo. Já como exemplo do círculo virtuoso existe o caso dos sistemas operacionais da Microsoft, onde o Windows é valorizado devido ao grande número de usuários que o utilizam, fazendo com que seja o padrão do setor.

Figura 2 – Valor x Popularidade



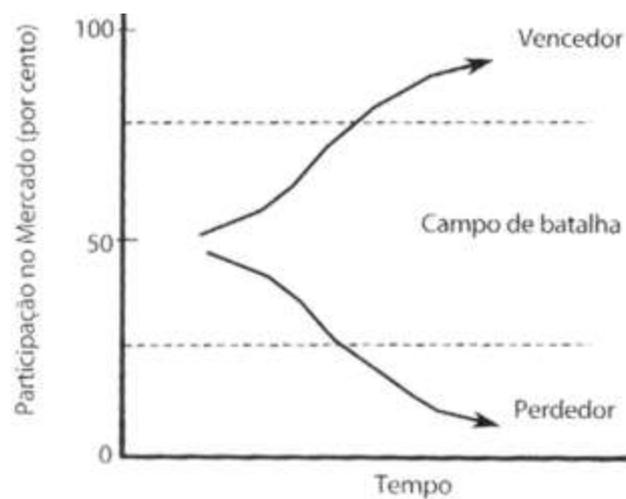
Fonte: (SHAPIRO, VARIAN, 1999:212)

Assim o feedback positivo acaba por fortalecer o mais forte e enfraquecer o mais fraco, como mostra a figura 2. No entanto, isso não significa que o produto que estiver atrás no mercado irá necessariamente perder, como exemplo disso temos o MS-Word que, apesar de ter permanecido atrás do WordPerfect por um bom tempo, hoje é líder do mercado de editores de textos.

Na economia da informação o feedback positivo em seu ciclo virtuoso reforça ainda mais a posição de líder do produto, através das economias de escala do lado da demanda, onde um produto como o MS-Windows torna-se uma razão para ser usado devido a grande quantidade de pessoas que utilizam esse sistema. Isso faz com que qualquer sistema operacional concorrente que surja tenha dificuldade em obter massa crítica para concorrer com o sistema da Microsoft.

As exterioridades de rede positivas, que agem no sentido de conformar situações nas quais um membro de um mercado interfere sobre outros membros sem gerar contrapartidas de receitas ou rendimentos, faz surgir o feedback positivo. Assim, se um usuário começa a utilizar um programa que grava e lê vídeos em um determinado formato, outros usuários ficarão mais propícios a utilizar o mesmo programa para poderem trocar arquivos entre si. Um exemplo bastante recente disso são as redes de troca ponto-a-ponto, tais como o Napster, o Kaaza, e Fastrack onde a rede que obtiver um maior número de usuário tendem a receber mais usuários novos, pois possuem maiores possibilidades para a troca de arquivos entre usuários.

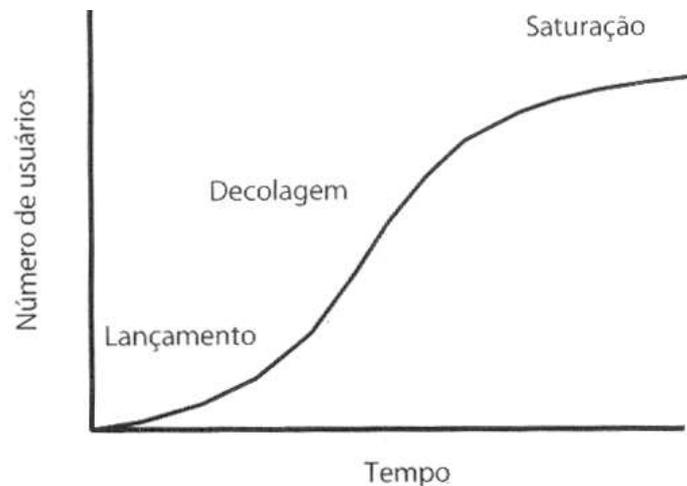
Figura 3 – Feedback positivo



Fonte: (SHAPIRO, VARIAN, 1999:208)

O feedback positivo segue um padrão de uma curva em S, onde na fase inicial temos o lançamento do produto, a decolagem do produto (acentuada pelo feedback positivo) e por último a saturação do produto. Esse comportamento pode ser visto na figura 4:

Figura 4 – Curva em S



Fonte: (SHAPIRO, VARIAN, 1999:210)

7. Estratégias para combater o possuidor da vantagem do feedback positivo

Shapiro e Varian colocam duas abordagens para contornar as vantagens inerciais das empresas que são estão sendo alavancadas pelo feedback positivo e pelas vantagens do aprisionamento tecnológico dos clientes: a estratégia da evolução tecnológica e a estratégia da revolução tecnológica

A estratégia da evolução se baseia em facilitar a migração dos usuários para a nova tecnologia, para isso se devem utilizar táticas que permitam reduzir o aprisionamento tecnológico, diminuindo os custos de troca e proporcionando uma migração sem ruptura.

Esta estratégia foi utilizada diversas vezes pela Microsoft para conseguir que usuários que utilizavam programas de versões antigas ou de concorrente, migrassem para uma nova versão.

No entanto esta estratégia esbarra em obstáculos técnicos e legais. O primeiro refere-se às dificuldades de criar uma nova tecnologia que se mantenha compatível com a tecnologia anterior e ao mesmo tempo possua recursos novos que a tornem atraente para novos consumidores.

Manter a compatibilidade com a tecnologia anterior é essencial para reduzir os custos de trocas para os clientes. No entanto esta estratégia pode ter como principal custo à limitação do desempenho da nova tecnologia, abrindo caminho para que novos produtos revolucionários conquistem o mercado.

O obstáculo legal surge da possibilidade de não conseguir o direito legal de produzir produtos similares aos já instalados. Essa barreira normalmente aparece nas formas das patentes que mantêm o direito a propriedade intelectual de determinado fabricante, podendo este cobrar pelo licenciamento do produto ou até mesmo impossibilitar que o produto seja fabricado por um novo concorrente. Um exemplo disso foi à estratégia da Hewlett-Packard quando patenteou um número

muito grande das peças de suas impressoras a jato de tinta, causando sérios transtorno para que a Epson conseguisse lançar estes produtos.

A estratégia da revolução parte do princípio da ruptura, oferecendo uma tecnologia que traga uma grande quantidade de melhorias, tornando-se irresistível para os usuários.

Segundo Shapiro e Varian, esta estratégia primeiramente atrai os clientes mais preocupados com o desempenho, que são os usuários pioneiros e normalmente são especialista da área e possuem a capacidade de influenciar os clientes de massa. Ao conquistar esses usuários pioneiros, os fabricantes buscam utilizar a influência desses clientes para começar a impulsionar o feedback positivo, fazendo com que os consumidores acreditem que o sucesso do produto é inevitável.

Esta estratégia tende a proporcionar melhores resultados em um mercado em rápido crescimento e com baixo aprisionamento. No entanto, a estratégia da revolução é arriscada, uma vez que é quase impossível saber se a sua tecnologia obterá sucesso ou não.

8 - Consolidação de padrão: o projeto dominante

Utterback (1994) desenvolve um modelo explicativo da inovação industrial após o estudo de diversos, casos. O modelo, segundo o autor, é válido para explicar as inovações introduzidas dentro do contexto industrial dos Estados Unidos nos últimos 150 anos, em produtos onde custo e desempenho sejam fatores importantes. Nesse modelo, o autor explica como, a partir de projetos inovadores numa indústria, se consolida um determinado padrão que atende as necessidades dos clientes e como esse padrão torna-se dominante. Segundo Utterback o projeto inovador vai se cristalizando ao longo do tempo, até atingir a forma de um projeto dominante.

Utterback distingue três fases na inovação da indústria que vão ocorrendo ao longo do tempo: fluída, transitória e específica.

Na fase fluída de um setor, o novo conceito é lançado por uma ou mais empresas. Os vários concorrentes operam com projetos diferentes para atender a mesma necessidade dos clientes. Os concorrentes tentam descobrir quais as características mais adequadas do produto do ponto de vista dos clientes. Como um produto é um “*pacote de valor ou feixe de benefícios*” (Kotler, Fox, 1994:257) não é evidente para os produtores, qual é a melhor solução do ponto de vista dos clientes.

Num certo momento há uma escolha por parte dos clientes, os que manifestam preferir uma solução em lugar de outra. O projeto preferido conquista a fidelidade do mercado. Esse projeto é aquele “*que satisfaz o usuário em termos de interação das possibilidades técnicas e opções de mercado*” (Utterback, 1994:27). Além dos aspectos técnicos, vários fatores adicionais contribuem para que um projeto seja dominante: uma forte imagem corporativa, as vantagens que uma empresa estabelecida já possui e que pode alavancar em seu favor, os canais de distribuição, a imagem, os regulamentos setoriais, as manobras estratégicas e a interação com os clientes, forçando uma aprendizagem de mercado que seja favorável.

O projeto dominante reduz a liberdade inicial de requisitos de desempenho a serem atendidos por um produto, e os transforma em implícitos ao próprio projeto. Isto ocorre, pois os clientes “votaram” em favor de uma solução, o que diminui as

possibilidades de sobrevivência de outras soluções, salvo em nichos específicos. Assim, quando a IBM lançou o PC, esta solução tornou-se o projeto dominante, no caso de computadores pessoais, levando a saída de mercado a empresas que já operavam (Sinclair, Amiga, etc.) e deslocando a Macintosh para um nicho de mercado.

Tendo ocorrido a escolha do projeto dominante, as outras empresas serão obrigadas a atingir a lógica do projeto dominante para poder continuar concorrendo. O projeto dominante consolida a definição de padrões técnicos e a padronização de partes. A concorrência desloca-se das características inovadoras para dimensões de custo, escala e desempenho (Utterback, 1994:34). A taxa de inovação se reduz, passando a ser mais de natureza incremental para apoiar melhorias de eficiência e desempenho.

Como que a adequação do próprio produto ao projeto dominante leva tempo, muitas empresas podem não conseguir-lo ou podem ficar em desvantagem, frente ao rival mais rápido, que já se apropriou de vantagens de custo e de escala. Isso pode significar sua saída da indústria. O número de empresas concorrentes diminui sensivelmente, uma vez estabelecido o projeto dominante. Se uma empresa agrega a ocupação de mercados, a obtenção de vantagens de custo e de escala à vantagem da detenção do projeto dominante e, ademais, aposta em contínuas inovações incrementais terá, possivelmente assegurada, uma confortável liderança em seu setor, até o aparecimento de alguma inovação radical que ameace sua posição.

Uma empresa que obtenha um projeto dominante usufruirá vantagens competitivas alavancadas pelo feedback positivo, conseguindo um número cada vez maior de usuários que irão reforçar sua posição como projeto dominante.

Por tanto, uma empresa na área de software, que se aproprie dos benefícios de um projeto dominante, será alavancada pelos efeitos positivos do *feed-back* e os concorrentes terão sérias dificuldades para desalojá-la. Pelo contrário, essa empresa terá condições de expulsar a maioria (ou totalidade dos concorrentes) tendendo a impor uma situação de monopólio. Este é o caso da Microsoft, que impôs um monopólio no referente aos sistemas operacionais.

Para desalojar essa empresa, serão necessários esforços coordenados de diversos agentes, sejam empresas concorrentes, sejam usuários além do papel do setor público apoiando esses esforços. Esses agentes, além de gerar uma alternativa ao projeto dominante, deverão obter os “votos” do mercado em favor de sua solução.

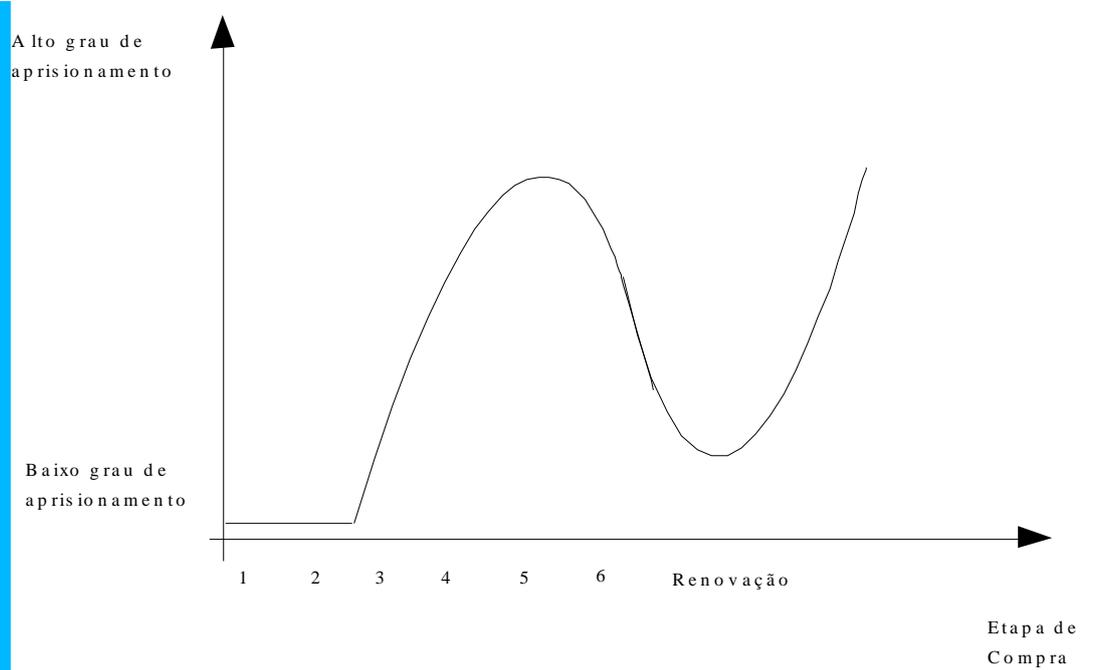
9 - Aprisionamento voltado ao software

As etapas do processo decisório para compra de um software são:

1. Definição dos requisitos.
2. Escolha do software
3. Aquisição
4. Implantação e integração com outros sistemas
5. Uso inicial
6. Substituição (em função da obsolescência do software)

O custo de aprisionamento cresce a partir da etapa de aquisição e torna-se extremamente elevado com o uso inicial, atingindo valores máximos com a integração que vai naturalmente ocorrendo com outros softwares. A obsolescência do software leva a estudos de substituição, nesse ponto o aprisionamento torna-se mínimo (porém não nulo). Nessa etapa o usuário pode decidir manter o padrão anterior, reforçando novamente o aprisionamento, ou escolher outro padrão, entrando em um novo padrão de aprisionamento, que pode ser mais forte ou mais fraco e com custos maiores ou menores. A figura 5 mostra a variação do grau de aprisionamento ao longo do tempo.

Figura 5– Grau de aprisionamento nas distintas etapas de compra



Fonte: elaboração própria

Os principais tipos de aprisionamento que afetam mais fortemente o software são:

- Informações e Banco de Dados.
- Custos de busca.
- Treinamento em marca específica.
- Incompatibilidade de sistemas.
- Comprometimento da cadeia com determinado padrão.

O mais sério tipo de aprisionamento que dificulta a migração entre softwares está relacionado à preservação e conservação dos dados já existentes. Torna-se praticamente impossível para uma empresa adotar um novo software caso seus dados não possam ser transferidos, se o processo de transferência é caro ou arriscado demais.

Portanto, é imprescindível que um novo software seja capaz de ler e gravar dados das versões líderes de mercado, diminuindo o aprisionamento e tornando-se uma opção para os usuários.

Os custos de busca surgem na hora em que a organização (ou o usuário) optar pela migração de software, pois é necessário procurar programas que atendam as necessidades do usuário e realizar diversos testes para garantir a qualidade do software e sua compatibilidade com o resto do sistema do usuário.

Outra forte barreira à migração do software é o treinamento em um software específico. Os usuários que estiverem habituados a utilizar um software poderão, no início, não apresentar o mesmo desempenho com o novo programa. Além disso, existe o fator psicológico para os usuários de terem que aprender a lidar com uma nova ferramenta, fazendo com que muitos deles imponham resistência à mudança do software.

A incompatibilidade de sistema constitui de mais um problema para do software, pois o usuário precisa garantir que os diversos sistemas que possui possam trocar dados. Por exemplo, quando se deseja trocar de banco de dados, é necessário garantir que os programas que acessavam os dados do banco de dados antigo consigam acessar a nova base de dados. Outro exemplo se refere à mudança de sistema operacional, onde programas feitos para o sistema vigente podem não rodar no novo sistema operacional. Além disso, é necessário garantir que existam drivers para todos os hardwares utilizados pelo sistema.

O comprometimento da cadeia com determinado padrão poderá agravar o problema da incompatibilidade de sistemas, pois um sistema operacional que seja líder absoluto do mercado, tenderá a possuir um número maior de programas e drivers sendo feito para o ambiente dominante, limitando assim, as opções de sistemas operacionais concorrentes.

Portanto, o usuário ao estudar a possibilidade de migração deve considerar todos os aprisionamentos com os quais está envolvido. Além disso, para garantir uma melhor chance de sucesso, o usuário deve realizar a migração em um momento no qual o aprisionamento esteja reduzido.

Os desenvolvedores de software livre têm reduzido o aprisionamento tecnológico tornando os programas criados capazes de importar e exportar dados com os programas líderes. Além disso, estes programadores têm buscado criar interfaces gráficas similares com os programas mais usados. Isso reduz o aprisionamento uma vez que torna mais simples o aprendizado do novo software.

Outro fator que reduz o aprisionamento é que hoje existem diversos softwares similares aos principais programas proprietário. A tabela a seguir mostra alguns programas similares entre o Windows e o Linux .

Apesar de tudo, o aprisionamento existente através dos custos de busca e treinamento em marcas específicas, conversões de dados e incompatibilidade de sistemas ainda são relevantes.

Tabela 2 – Programas similares entre Windows e Linux

Tipo	Windows	Linux
Navegadores de Internet	Internet Explorer, Netscape / Mozilla, Opera [Prop], Firebird, ...	1) Netscape / Mozilla. 2) Galeon. 3) Konqueror. 4) Opera. [Prop] 5) Firebird 6) Nautilus 7) Epiphany. 8) Links. (with "g" key). 9) Dillo. (Russian language patches - here). 10) Encompass.
Clientes de e-mail	Outlook Express, Netscape / Mozilla, The Bat, Eudora, Becky, Datula, Sylpheed / Sylpheed-claws, Opera	1) Evolution. 2) Netscape / Mozilla messenger. 3) Sylpheed / Sylpheed-claws. 4) Kmail. 5) Gnus. 6) Balza. 7) Bynari Insight GroupWare Suite. [Prop] 8) Arow. 9) Gnumail. 10) Althea. 11) Lianail. 12) Aethera. 13) MailWarnor. 14) Opera
Mensagens Instantaneas	ICQ Lite, ICQ Corp, MSN, AIM, Yahoo, ...	1) Licq (ICQ). 2) Centericq (Nearly all IM protocols, console). 3) Alicq (ICQ). 4) Micq (ICQ). 5) GnomelCU (ICQ). 6) Gaim. (Nearly all IM protocols). 7) Aytim. (Nearly all IM protocols). 8) Kopete. 9) Everybuddy. 10) Simple Instant Messenger. 11) Imici Messenger. 12) Ickle (ICQ). 13) aMSN (MSN). 14) Kmerlin (MSN). 15) Kicq (ICQ). 16) YSM. (ICQ, console). 17) kxicq. 18) Yahoo Messenger for Unix. 19) Kmess (MSN). 20) AIM. 21) MSNra. (console)

Fonte: LinuxShop (2003)

10 - Custos do aprisionamento e da migração para o software livre

O custo do uso do software (C) envolve além dos custos de compra das licenças (l), o valor do treinamento para o uso (t) e o valor atual da renovação periódica das licenças (r) (dentro de um determinado horizonte de planejamento). Também devem ser considerados os custos de adequação do software novo ao ambiente computacional existente (a). Esta variável (a) representa o montante de custos necessários para que o software atenda às necessidades dos usuários, para que sejam superados os problemas de compatibilidade de programas e para atender a demanda de hardware).Ou seja:

$$C = l + t + r + a \quad (1)$$

Uma empresa nova, que necessite de 16 estações de trabalho, possuindo cada uma um sistema operacional e um aplicativo de escritório, instalados, além de servidor de e-mail, um servidor de web, um servidor de banco de dados e duas estações para desenvolvimento conteúdo cada uma, um sistema operacional, um aplicativo Office, um programa para editoração de imagens e aplicativo para desenvolvimento de software, gastará cerca de R\$73.915,00 caso opte por uma solução baseada em softwares proprietários, como mostra a tabela 3.

Tabela 3 – Gastos para aquisição de softwares proprietários

Software	Preço Unitário	Quantidade	Total
Microsoft Windows XP professional Edition	R\$ 989,00	18	R\$ 17.802,00
Microsoft Office XP full professional	R\$ 1.850,00	18	R\$ 33.300,00
Antivirus	R\$ 60,00	20	R\$ 1.200,00
Microsoft SQL Server 2000	R\$ 5.013,00	1	R\$ 5.013,00
Microsoft Windows 2000 Server	R\$ 2.800,00	2	R\$ 5.600,00
Adobe PhotoShop	R\$ 2.000,00	2	R\$ 4.000,00
Microsoft VisualStudio (programa para desenvolvimntno de software)	R\$ 3.500,00	2	R\$ 7.000,00
		Total	R\$ 73.915,00

Valores retirados de www.buscape.com.br em 01/11/2003

Além disso, a empresa terá que voltar a gastar com software a aproximadamente a cada três anos, caso deseje manter seus computadores com softwares atualizados.

Já um sistema baseado em softwares livres com a mesma funcionalidade terá custo zero, como pode ser visto na tabela 4. Isso é possível, pois os programas podem ser baixados de forma gratuita na Internet. Além disso, a empresa não incorrerá em gastos caso deseje manter seu parque de computadores com softwares atualizados.

Tabela 4 – Gastos para aquisição de softwares livres

Software	Preço Unitário	Quantidade	Total
Linux RedHat 9	R\$ 0,00	20	R\$ 0,00
OpenOffice	R\$ 0,00	18	R\$ 0,00
Sendmail (servidor de e-mail)	R\$ 0,00	1	R\$ 0,00
PostgreSQL	R\$ 0,00	1	R\$ 0,00
Apache (servidor de web)	R\$ 0,00	1	R\$ 0,00
Eclipse (programa para desenvolvimntno de software)	R\$ 0,00	0	R\$ 0,00
GIMP (programa equivalente ao PhotoShop)	R\$ 0,00	2	R\$ 0,00
		Total	R\$ 0,00

Portanto, no caso de uma empresa nova, os valores de I e r serão nulos, caso opte pelo software livre. Provavelmente terá custos semelhantes de adequação (a) entre ambas soluções. Poderá ter algum custo adicional, que deve ser entendido como marginal em treinamento, t , se opta pelo software proprietário dado que a maioria das pessoas, conhece Windows (o que é uma externalidade que favorece os softwares proprietários) ou um custo maior se opta pelo software livre, mais que em função dos esforços realizados pelos desenvolvedores de software livre não devem ser muito importantes. No referente aos custos de treinamento, segundo a IT Mídia, o custo de treinamento para capacitação na tecnologia do software livre é equivalente a 30% do gasto de um ano com licença de software proprietário. Os testes foram realizados pela IT Mídia em parceria com a Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo (FEA), com a Fundação Instituto de Administração (FIA) e com o Instituto de Ciências Matemáticas e de Computação da unidade da Universidade de São Paulo em São Carlos (ICMC / USP São Carlos) (Terra Informática, 2003). Portanto, os valores mostrados, são bem claros quanto à vantagem da adoção dos softwares livres.

Outra vantagem é o fato que o software livre pode ser configurado de forma tal a utilizar menos recursos de máquina, possibilitando assim a sobrevida dos equipamentos que acompanhavam a demanda de hardware dos softwares.

Uma empresa que já tem instalado um parque computacional com softwares proprietários deverá comparar, para um determinado período de planejamento, o valor atual do custo esperado de ficar pagando a renovação das licenças para os softwares atuais (r) e de adquirir as licenças dos novos softwares (l) e o custo do treinamento para usar os novos softwares proprietários (tp) com o valor atual do custo dos treinamentos para uso do software livre (tl) e adequações necessárias (alterar programas incompatíveis com o sistema livre e gasto para conversão de dados).

Sendo VAp, o valor atual do custo de posse do software proprietário e VAl o valor atual do custo da migração e posse do software livre, pode-se definir:

$$VAp = l+r+tp \quad (2)$$

$$VAl = tl + a \quad (3)$$

Assim, uma empresa que considere um determinado horizonte de planejamento (por exemplo 5 anos), deverá considerar a migração uma opção válida caso o custo de se fazer um novo treinamento, de alterar programas incompatíveis com o sistema livre e o gasto para conversão de dados for menor que o custo atual de treinamento da empresa com os softwares proprietários, aquisições de novos softwares e atualizações dos programas existente no período considerado no período de planejamento (ou seja deverá migrar quando $VAp > VAl$).

Adicionalmente cabe observar que o software livre tem a vantagem adicional de ser mais estável que o software proprietário atual, tendo menores problemas de manutenção e ataques de vírus (Terra Informática, 2003).

Neste caso pode-se revisar a equação (2) que indica o valor atual da posse do software proprietário, acrescentando mais uma variável de custo (m), o custo de manutenção do software proprietário em relação aos problemas de instabilidade, manutenção, ataque de vírus.

Assim a equação (2) deve ser expressa como:

$$VAp = l+r+tp + m \quad (4)$$

Pode-se também, repensar a equação (3) diminuindo do custo da migração para o software livre a necessidade de compra de novos equipamentos (h), dada a possibilidade de reaproveitar, ou aumentar a vida útil do hardware, em função de configurações que demandem menos recursos das máquinas. Portanto a equação 3 pode ser enunciada como:

$$VAl = tl + a - h \quad (5)$$

Ou seja, a análise dos casos mostra, que além das vantagens de qualidade, o custo de migração pode ser menor que aquele esperado a partir do enunciado original das variáveis das equações 2 e 3.

A migração obterá vantagens quando os softwares feitos sob medida para a empresa e que só rodem em uma plataforma proprietária estiverem obsoletos (ponto mínimo de aprisionamento da figura 5), sendo necessária a confecção de novos programas.

Mesmo que a empresa não se encontre em um momento de aprisionamento mínimo, esta pode considerar a migração de forma gradual de seus sistemas, uma vez que é possível manter a troca de dados os programas livres e proprietário. Isto diminuiria o grau de aprisionamento da empresa com os softwares proprietários.

Conclusão

O objetivo do trabalho foi apresentar um modelo que possibilitasse avaliar a oportunidade de trocar o software proprietário pelo software livre.

Um aspecto inicial abordado foi a investigação sobre as razões pelas quais, mesmo existindo softwares livres, empresas, organizações e pessoas continuavam usando softwares proprietários. Também interessava entender as condições que seriam necessárias para que houvesse migração para o software livre. Para estudar o assunto foi necessário formular um enfoque interdisciplinar. Parte das explicações veio do campo da economia, de onde foi entendido o conceito de aprisionamento tecnológico, enunciado por Shapiro e Varian, que mostra as razões das dificuldades de sair da situação de aprisionamento.

Outra parte das explicações veio do campo da computação, onde foram estudados os aspectos técnicos e de qualidade de software, que permitiram entender, tanto a natureza técnica dos aprisionamentos, como os esforços que os desenvolvedores de software livre estão fazendo para diminuir o custo que as empresas, organizações e pessoas, tem para sair da situação de aprisionamento.

Verificou-se que o software livre enfrenta dificuldade para ser adota em parte devido a resistência a mudanças dos usuários e gerentes, ao feedback positivo que o software proprietário usufrui e outros tipos de aprisionamento que foram discutidos ao longo do trabalho.

Finalmente foram explicitadas as condições nas quais uma empresa poderia pensar em migrar para o software livre. Considerando que o software livre disponibiliza programas de qualidade, é mais estável (com o que se reduzem os custos de manutenção), tem uma menor demanda de hardware e que os custos de treinamento não são tão altos, tal como verificados na pesquisa patrocinada pela IT Mídia (Terra Informática, 2003) e que representam 30% dos gastos com licenças de softwares proprietários, as empresas podem, como estratégia de diminuição de custos e aumento da qualidade dos sistemas encontrar oportunidades de migração ao longo das distintas etapas de compra do software.

Bibliografia

- BACIC, Nicolas. **O software livre como alternativa ao aprisionamento tecnológico imposto pelo software proprietário.** Monografia de graduação em Ciências Econômicas, Instituto de Economia, Universidade Estadual de Campinas, 2003,
- BANCO CENTRAL. **Balanco de Pagamento (série histórica).** Disponível em: <http://www.bcb.gov.br/ftp/notaecon/balpagm.zip>. Acesso em 02 de abril de 2004.
- HAUBEN, Ronda. **Unix History.** Disponível em <http://www.unixsup.com/unixlinux/historiaunixin.html>. Acesso em 14 de julho de 2003.
- LINUXSHOP. **Amazon The table of equivalents / replacements / analogs of Windows software in Linux.** Disponível em: <http://linuxshop.ru/linuxbegin/win-lin-soft-en/table.shtml>. Acesso em 7 de setembro de 2003.
- KOTLER, Philip; FOX, Karen F.A. **Marketing estratégico para instituições de ensino.** São Paulo: Atlas, 1994.
- LOTTERMANN, Jair. **Viabilidade da Utilização de Software Livre em um Ambiente Empresarial.** Dissertação de Mestrado. Puc-Campinas, 2001.
- MEDEIROS, Elizabet M. S., SAUVÉ, Jacques P. **Sistemas Avaliação do Impacto de Tecnologias da Informação Emergentes nas Empresas.** Rio de Janeiro: Qualitymark, 2003.
- MCT – Ministério de Ciência e Tecnologia. **1ª. Oficina de Software Livre, 2003** (<http://www.mct.gov.br/Temas/sl/Default.htm>, acesso em 03/05/2004).
- MIT- SOFTEX (2002). **A indústria de Software no Brasil -2002.** Campinas: SOFTEX, 2002.
- SANTOS, Enarni M. **Aprisionamento Tecnológico: Novos desafios da gestão das estratégias organizacionais na era da informação.** Disponível em: <http://www.ead.fea.usp.br/cad-pesq/arquivos/v08-1art06.pdf>. Acesso em 21 de abril de 2003.
- SHAPIRO, Carl, VARIAN, Hal. **A economia da informação: Como os princípios econômicos se aplicam à era da Internet.** Rio de Janeiro: Campus, 1999.
- TERRA INFORMÁTICA **Linux é 74% mais estável que Windows, aponta teste.** (13 novembro de 2003) Disponível em: <http://informatica.terra.com.br/interna/0,,OI210362-EI553,00.html>. Acesso em 18 de novembro de 2003.
- UTTERBACK, James. **Dominando a Dinâmica da Inovação.** Rio de Janeiro: Qualitymark Editora Ltda., 1996.
- VASCONCELOS, L. A. Os ciclos relevantes do processo de produção e a gestão estratégica de custos: breve análise do processo de custeio nos setores de produção de Software e de Serviços de Informática. Trabalho apresentado no: **VI Congresso Internacional de Custos**, 1999, Braga.