Licenciamento de Software Público - um estudo de alternativas para o Portal do Software Público Brasileiro

Carlos Denner dos Santos Júnior

Daniel Montenegro Hastenreiter

Universidade de Brasília

Resumo:

A evolução da tecnologia da informação (TI) oferece múltiplos benefícios às organizações privadas e públicas. Para o governo, aumenta a eficiência e transparência de suas operações e promove a sociedade da informação. O Software Público Brasileiro (SPB), coordenado pelo Ministério do Planejamento, Orçamento e Gestão (MPOG), inseriu-se nesse contexto e consolidou-se como política pública, reduzindo custos e compartilhando conhecimento dentro de um ecossistema livre e colaborativo. A demanda por maior impacto do software público na oferta de softwares, na adoção das soluções disponibilizadas e na atração de colaboradores e usuários requer intervenções. Entendendo que um dos principais desafios de um ambiente de software livre é atrair e motivar colaboradores e que um fator-chave da atratividade é a licença, demonstra-se a necessidade de um estudo de propostas de licenciamento e seus impactos para o Portal SPB. As recomendações preliminares são a elaboração de critérios para a aceitação de licenças pela SLTI, visando à ampliação das opções disponíveis, e a aceitação de soluções não maduras no Portal SPB, devido à importância das fases pré-estáveis no desenvolvimento de softwares livres. Espera-se que a implantação destas ações e de alternativas que ainda serão avaliadas e recomendadas impactem positivamente no Portal SPB, atraindo mais ofertantes, motivando colaboradores e disseminando ainda mais o software público brasileiro.

1. Introdução

No âmbito da administração pública federal, a Secretaria de Logística e Tecnologia da Informação – SLTI é responsável pelas normas relacionadas ao uso de Tecnologia da Informação. Também é sua atribuição planejar, coordenar, supervisionar e orientar, normativamente, as atividades do Sistema de Administração de Recursos de Tecnologia

da Informação – SISP (BRASIL, 2012a). O SISP é responsável pelo planejamento, coordenação, organização, operação, controle e supervisão dos recursos de tecnologia da informação dos órgãos e entidades da administração pública federal direta, autárquica e fundacional, em articulação com os demais sistemas utilizados direta e indiretamente na gestão da informação pública federal, conforme decreto 7579, de 11 de outubro de 2011.

A evolução da tecnologia da informação (TI) vem possibilitando novas estratégias de negócios e novos mercados a serem explorados. Desse modo, a TI ganha cada vez mais importância estratégica. Gurbaxani e Whang (1991) citaram os múltiplos papéis que um sistema de informação pode ter numa organização: aumentar eficiência nas operações; processar transações financeiras básicas; dar suporte às decisões gerenciais; monitorar e avaliar desempenho; documentar e comunicar. Os autores, a partir das teorias da agência e do custo de transação, concluem que a TI pode facilitar tanto a centralização como a descentralização de decisões, facilitar o crescimento horizontal e vertical de empresas, assim como a cooperação *business-to-business* dentro da cadeia produtiva. Além disso, ela permite a diminuição de custos de comunicação de informação, ao melhorar a qualidade e velocidade do processamento de informação (GURBAXANI; WANG, 1991).

Para o governo, a TI é uma oportunidade de oferecer transparência de modo eficiente e conveniente, promovendo a boa governança, reduzindo potencial para comportamentos corruptos, melhorando as relações entre funcionários do governo e cidadãos e permitindo que cidadãos monitorem suas atividades (ARPIT, 2012). Nesse sentido, governos buscam operar como plataformas, que “compartilham informação”, são “transparentes em suas operações e na tomada de decisão” e que “alavancam projetos liderados pelos cidadãos” (EAVES, 2010).

Uma das maneiras de suprir a necessidade por tecnologias da informação é desenvolvê-las internamente. Dessa forma, é maior a garantia de atendimento dos requisitos desejados e de manutenção da tecnologia. Porém, os elevados custos, a quantidade de tempo exigido e a expertise necessária, que pode estar concentrada em poucas ou nenhuma das pessoas da organização, impossibilitam muitos projetos. Outra opção é a compra, que poupa tempo e recursos, enquanto aproveita-se da experiência

da contratada em manutenção e treinamento. Apesar disso, o código pertence ao fornecedor e cria-se uma dependência em relação ao suporte técnico para resolução de quaisquer problemas (CINCOM, 2008). O aprimoramento é dificultado, já que apenas uma empresa pode fazê-lo legalmente e, muitas vezes, tecnicamente, além de ameaçar a governança da ferramenta. Ademais, a utilização de software proprietário usualmente sujeita o governo a adquirir outras soluções do mesmo vendedor em busca de compatibilidade com produtos já adquiridos e economia de gastos com treinamento e manutenção (DAFFARA; GONZALEZ-BARAHONA, 2010).

A crescente demanda por ferramentas de TI na gestão pública expõe a dependência dos governos em relação às empresas desenvolvedoras de software. Os contratos de aquisição de software normalmente terminam com a entrega do produto final, dificultando aprimoramentos contínuos (BURTON, 2010) e o controle dos recursos tecnológicos por parte da contratante. A rigidez do processo de contratação e a burocracia são entraves que geram perda da capacidade de correções responsivas e enfraquece a relação com o usuário final, ameaçando a eficiência e utilidade das soluções.

Nesse ambiente, o software livre transparece como remédio e governos de todo o mundo se engajam em ambientes abertos, pois além de promover a sociedade da informação, a inserção do software livre nas agendas políticas beneficia usuários intensivos de serviços baseados em software (GONZALEZ-BARAHONA; ROBLES, 2005). No caso brasileiro, o software livre potencialmente supre a demanda de informatização da administração pública enquanto cumpre as diretrizes constitucionais de incentivo à pesquisa e capacitação tecnológica e de incentivo ao mercado interno através da autonomia tecnológica (FALCÃO *et al*., 2009).

A principal vantagem do software livre é o compartilhamento do código fonte. (SABINO; KON, 2009). Esse compartilhamento permite que várias pessoas tenham acesso ao código e cada uma contribua de certa forma com seu aprimoramento e correção de erros. Desse modo, é comum haver extensa revisão dos trabalhos pelos pares (RAYMOND, 1999), já que cada usuário aborda a tarefa de caracterização de bugs de forma e ângulo diferente (VON HIPPEL 2005). Isso torna o desenvolvimento do

software distribuído e independente de um grupo específico, podendo se tornar menos custoso e de melhor qualidade (SANTOS JR., 2010).

Do ponto de vista econômico, a adoção de softwares livres permite maior opção de provedores de aplicações e serviços, reduzindo o custo e propiciando mais segurança em caso de saída de fornecedor do mercado ou de falha deste em oferecer atualizações. Ao permitir que vários fornecedores compitam entre si, sinaliza-se a continuidade de aperfeiçoamento do software e provoca-se redução de preços. Fornecedores também se beneficiam, pois custos e risco associados ao desenvolvimento do software são diluídos entre os diversos concorrentes (KON et al., 2011).

A difusão rápida e bem sucedida de softwares livres como Linux, Apache e Firefox chamou a atenção de pesquisadores e líderes de projetos de software (FUGGETTA, 2003) e o elevado interesse por esse modelo de desenvolvimento de software pode ser percebido com a abertura do código fonte de aplicações de projetos públicos e privados (SANTOS JR., 2010).

O Software Público é um projeto coordenado pelo Ministério do Planejamento, Orçamento e Gestão (MPOG), por meio da SLTI (Secretaria de Logística e Tecnologia da Informação) e insere-se no atual contexto de busca por transparência, controle social e redução de custos (BRASIL, 2014). O Software Público Brasileiro iniciou em 2006, quando Governo Federal priorizou o software livre como opção estratégica para o desenvolvimento de software no âmbito do Governo e ocorreu a abertura de código do CACIC, software para inventário de infraestrutura de TI. A compreensão dos impactos da abertura e a percepção do potencial ganho que um ambiente colaborativo podia oferecer motivou o empenho em criar um ambiente virtual para tal (BRASIL, 2012b).

A característica que distingue o Software Público de software livre é assunção de certas responsabilidades que garantam uso adequado dos softwares. São elas (ALVES *et al*., 2009):

> “Prover um software com documentação completa de instalação e preparado para funcionar; Disponibilizar um ponto focal ou uma equipe que possa fazer interlocução com a sociedade e encaminhar suas demandas; A existência de um ambiente virtual que operacionalize a comunicação com o usuário (fórum, ferramentas de controle de versão, etc.); Gestão da colaboração - além da

> gestão da comunidade virtual associada ao software liberado, a entidade também se compromete a realizar ações para incentivar a colaboração e gestão do conhecimento produzido (controle de versões, etc.)".

O objetivo inicial do software público era propiciar o compartilhamento de software pelas instituições públicas. Formaram-se, porém, comunidades virtuais no entorno destas ferramentas e um ambiente colaborativo complexo foi sendo formado (ALVES *et al*., 2009). Atualmente, o Portal contém 71 soluções de software (63 de código aberto), mais de 170 mil usuários cadastrados e uma média de aproximadamente 5.900 usuários por comunidade. Mediante cadastro prévio, qualquer cidadão pode obter gratuitamente as soluções.

São benefícios do Software Público: a economia dos recursos, atingida pelo compartilhamento e aproveitamento de aplicações existentes; a independência de fornecedores, devido à abertura do código que estimula competição; a segurança, já que a abertura do código permite seu estudo e aprimoramento constante; e o compartilhamento do conhecimento, alcançado pela troca de informações constante e disponibilização dos códigos fonte dos softwares.

Desde seu surgimento, os softwares do Portal SPB atendem às demandas das esferas público e privada. Além da agregação de usuários e colaboradores em torno de cada aplicação, uma rede de serviços desenvolve-se e a sociedade passa a beneficiar-se dos produtos do ambiente.

Para a administração pública, a criação de um arcabouço jurídico e a evolução de um ambiente cooperativo tornou real o desenvolvimento e o compartilhamento colaborativo de soluções. Sem dúvida, de 2006 até hoje a expansão do número de softwares, usuários e colaboradores no Portal demonstra a consolidação da iniciativa. Porém, nos últimos anos não foram atingidos objetivos desejados.

A Controladoria Geral da União - CGU emitiu um relatório preliminar de auditora a respeito do software público em agosto de 2014. O levantamento realizado pela CGU constatou que mais da metade das unidades pesquisadas não utiliza software público. Alguns dos motivos citados são a falta de soluções, o desconhecimento das utilidades oferecidas, a dificuldade de customização das aplicações e falta de engajamento das unidades à filosofia do software público.

Além disso, 92% das unidades consideram que as soluções disponíveis no Portal SPB não atendem ou atendem parcialmente às necessidades do órgão. O suporte da comunidade também é insatisfatório: 43% das unidades que tiveram dificuldades no uso de softwares públicos apontaram problemas com o apoio da comunidade. Também foram evidenciadas desatualização ou inexistência de documentação, erros e instabilidades em diversas aplicações, que causaram, em alguns casos, o abandono de softwares públicos em favor de soluções de mercado.

O compartilhamento de soluções entre a administração pública também não vem ocorrendo de modo adequado. Algumas das razões listadas são as exigências de documentação e registro prévio em INPI e a responsabilidade da unidade ofertante em indicar coordenadores das comunidades e oferecer suporte técnico.

As adversidades citadas podem ser combatidas com a atração maior de usuários, colaboradores e ofertantes. Para tal, estuda-se a aceitação de mais modelos de licença e a flexibilização de alguns requisitos técnicos e jurídicos. Em relação aos requisitos obrigatórios, a exigência de softwares apenas em versões maduras e estáveis inibe soluções jovens dos benefícios do ecossistema SPB, diminuindo a oferta potencial de aplicações. As imposições de manual, documentação e registro prévio das aplicações no INPI também são entraves a um possível aumento do impacto do software público. Uma resposta a isso seria a aceitação temporária de softwares no ambiente que não cumpriram ainda todos os requisitos. Estes seriam atendidos dentro de um prazo razoável enquanto o desenvolvimento do software se estende no ambiente do Portal. Acerca do licenciamento, a escolha da licença interfere na atração e repulsão de usuários e colaboradores de projetos de software livre (COLAZO; FANG, 2009; HASTENREITER, 2013; LERNER; TIROLE 2005; SANTOS JR. *et* al., 2011; SEN; SUBRAMANIAM; NELSON, 2008).

O software, como obra intelectual, está protegido pela legislação autoral. Isso significa que seu uso, modificação e distribuição estão sujeitos às condições estabelecidas pelo seu criador. É por meio da licença que são concedidos os direitos aos licenciados, declaradas as condições para exercício destes e assegurado o exercício dos direitos de propriedade em caso de violação dos termos da licença (LINDBERG, 2008).

2. Software como Propriedade Intelectual

É importante frisar que o direito autoral protege as formas de expressão das ideias e não as ideias em si: “São obras intelectuais protegidas as criações do espírito, expressas por qualquer meio ou fixadas em qualquer suporte, tangível ou intangível, conhecido ou que se invente no futuro” (art.7º da Lei 9.610/98).

Os direitos autorais são regulamentados no Brasil por meio da Lei nº 9.610 de 19 de fevereiro de 1998. A lei divide os direitos autorais em: patrimoniais - direitos de utilizar, fruir e dispor da obra- que podem ser negociados e transferidos; e morais - referem-se à imagem do autor e à ligação com sua obra - que são inalienáveis e irrenunciáveis. De modo geral, a legislação autoral reconhece programas de computador como obras intelectuais sujeitas à proteção no inciso XII do seu art. 7º. De modo específico, a proteção da propriedade intelectual de software se fundamenta na lei 9.609/98 de 19 de fevereiro de 1998, conhecida como Lei de Software. O regime de proteção cabível é o mesmo conferido às obras literárias pela legislação de direitos autorais e conexos vigentes no País, observadas algumas modificações. Não se aplicam ao software a proteção dos direitos morais (Art. 2º, § 1º), exceto o direito de reivindicar a paternidade do programa de computador e o direito do autor de opor-se a alterações não autorizadas, quando estas impliquem deformação, mutilação ou outra modificação do programa de computador que prejudiquem a sua honra ou a sua reputação. Diferentemente da Lei de Direitos Autorais, que estende a proteção por setenta anos após a morte do autor, o software tem direitos assegurados pelo prazo de cinquenta anos, contados a partir de 1º de janeiro do ano subseqüente ao da sua publicação ou, na ausência desta, da sua criação (Art. 2º, § 2º da Lei 9.609/98). Outra especificidade é o fato da titularidade dos direitos pertencerem ao empregador em casos de vínculo entre este e o autor (art. 4º da Lei 9.609/98).

Ao definir programa de computador como “expressão de um conjunto organizado de instruções em linguagem natural ou codificada” estende-se a proteção ao código objeto e ao código fonte do software. Ainda que essa proteção independa de registro, é possível fazê-lo junto ao INPI.

Nos casos em que for omissa a Lei 9.609/98, aplica-se a Lei 9.610/98 e, persistindo a omissão, aplica-se o Código Civil (FALCÃO *et al*., 2009).

Para que seja analisada a compatibilidade de licenças, como será feito adiante, é necessário o conhecimento das definições legais de obra derivada e coletiva, consoante Lei 9.610/98:

- Obra derivada - “a que, constituindo criação intelectual nova, resulta da transformação de obra originária” (Art. 5º, inciso VIII, alínea “g”).
- Obra coletiva - “a criada por iniciativa, organização e responsabilidade de uma pessoa física ou jurídica, que a publica sob seu nome ou marca e que é constituída pela participação de diferentes autores, cujas contribuições se fundem numa criação autônoma” (Art. 5º, inciso VIII, alínea “h”).

No artigo 17, § 2º, a Lei informa que “cabe ao organizador a titularidade dos direitos patrimoniais sobre o conjunto da obra coletiva”; ou seja, caso uma obra não seja transformada, mas apenas compilada com outras, os direitos patrimoniais pertencem ao organizador.

3. Software Livre

Desenvolver um software livre não significa somente oferecer acesso livre ao código fonte, nem ser um software desenvolvido por voluntários. A definição de Richard Stallman, fundador do movimento de software livre, se baseia em quatro liberdades fundamentais (www.gnu.org/philosophy/free-sw):

- Liberdade 0: A liberdade de executar o programa como quiser, para qualquer finalidade.
- Liberdade 1: A liberdade de estudar como o programa funciona, e modificá-lo. Acesso ao código-fonte é um pré-requisito para isso.
- Liberdade 2: A liberdade de redistribuir cópias para que você possa ajudar ao seu próximo.
- Liberdade 3: A liberdade de distribuir cópias de suas versões modificadas para outros. Acesso ao código-fonte é um pré-requisito para isso.

Efetivamente, a forma mais segura e amplamente aceita de delimitar o que é software livre é pela definição da Open Source Initiative (OSI). De acordo com os padrões da OSI, organismo reconhecido pela comunidade que aprova novas licenças,

para que seja considerada uma licença de software livre deve-se atender a uma série de condições (http://opensource.org/osd):

- Distribuição livre

  A licença não pode restringir qualquer parte de vender ou doar o software como componente de uma distribuição que contenha programas de diferentes fontes, nem cobrar *royalties* ou taxas.

- Código fonte

  O programa deve incluir código fonte e permitir distribuição em código fonte e em forma compilada. Caso a distribuição não inclua o código fonte, deve-se divulgar o modo de obtê-lo, cobrando por isso não mais que um custo razoável de reprodução.

- Trabalhos derivados

  A licença deve permitir modificações e trabalhos derivados e permitir que todos sejam licenciados sob a mesma licença do programa original.

- Integridade do autor do código fonte

  A licença pode restringir a distribuição do código fonte em forma modificada somente se a licença permitir distribuição de *patchs*. A licença pode requerer que trabalhos derivados carreguem um nome diferente do original.

- Não discriminação de pessoas ou grupos

  A licença não pode discriminar contra qualquer pessoa ou grupo de pessoas.

- Não discriminação de campos de trabalho

  A licença não pode restringir qualquer pessoa de usar o programa para um campo de trabalho específico.

- Distribuição da Licença

  Os direitos ligados a um programa devem se aplicar a todos para os quais o programa é redistribuído, sem a necessidade de execução de uma licença adicional.

- A licença não pode ser específica a um produto

  Os direitos ligados a um programa não podem depender de ele ser parte de uma distribuição específica de software.

- A licença não pode restringir outros softwares

  A licença não pode colocar restrições sobre outros softwares distribuídos juntamente com o software licenciado.

- A licença deve ser tecnologicamente neutra

Nenhuma cláusula pode ser baseada em qualquer tecnologia individual ou estilo de interface.

Atendidos aos pontos acima, cada licença pode conter diferentes restrições. São essas restrições que afetam os atores envolvidos no desenvolvimento de software livre, pois limitam o uso e modificação do software e seu código fonte. Caso uma licença, como a GPL, exija que trabalhos derivados distribuídos estejam sob a mesma licença do trabalho original, esta é classificada como *copyleft*. O escopo da cláusula de *copyleft* é definido pelo alcance da reciprocidade da licença, o que também é conhecido como efeito viral. O efeito viral pode aplicar-se apenas às partes do software que utilizaram o código fonte licenciado (*copyleft* fraco) ou se aplicar ao software como um todo (*copyleft* forte), gerando uma restrição de compilação. Desse modo, ao utilizar software sob licenças *copyleft* forte (e.g. GPL), compromete-se a transferir as restrições do software original para qualquer trabalho derivado que venha a ser criado.

Caso uma licença não contenha exigências de reciprocidade em relação a trabalhos derivados, ela é não *copyleft*. É o caso das licenças permissivas (e.g. BSD, MIT, Apache), que impõem menos restrições e se assemelham com software em domínio público. A idéia por trás dessas licenças é a de que alguns softwares são mais valiosos se abertos ao público para cópia, modificação e distribuição (ROSEN, 2004).

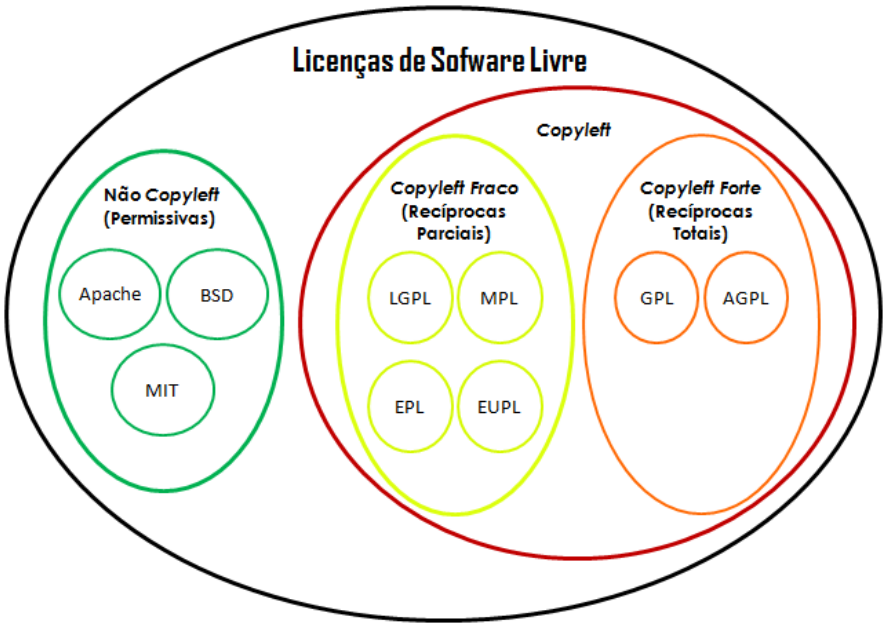

Figura 1, Fonte: Elaborado pelo autor.

4. Licenças de Software Livre

Ao tratar o software como propriedade intelectual, refere-se aos direitos autorais não só da versão executável, mas, sobretudo, do código fonte do programa. O detentor dos direitos autorais do software tem poder para distribuir cópias sob diferentes licenças, enquanto os licenciados só podem exercer os direitos concedidos pelo licenciante.

A licença de um software é o documento que formaliza o modo como o detentor dos direitos autorais de um software confere direitos como de uso e modificação do código fonte, aos licenciados. Enquanto que licenças proprietárias ou de código fechado permitem o uso do software, mas impõem uma série de restrições (principalmente quanto ao acesso e modificação do código fonte), licenças de software livre permitem que outras pessoas além do detentor do software o utilizem e modifiquem.

A importância prática das diferenças entre as licenças de software livre é refletida pelas diversas restrições que cada uma oferece ao uso e modificação do software e seu código fonte. Em um dos extremos está a GPL, altamente restritiva, e no outro está a BSD, altamente permissiva. Essas duas licenças expressam ideias diametralmente opostas sobre o movimento de software livre (DE LAAT, 2005). A Figura 2 ilustra de modo simplificado as diferenças entre algumas das principais licenças de software livre.

	BSD 3-Cláusulas	BSD 2-Cláusulas	MIT	Apache	LGPL	MPL 2.0	EPL 1.0	EUPL	GPL 2.0	GPL 3.0	AGPL
Uso Comercial	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Modificação	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Distribuição	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sublicenciamento	✓	✓	✓	✓	✓	✓	✓	✓	*	✗	✗
Garantia	✓	✓	*	*	*	✓	*	*	✓	✓	✓
Uso de Marca	✗	*	*	✗	*	✗	✗	*	*	*	*
Responsabilização	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Direitos Autorais Originais	!	!	!	!	!	!	!	!	!	!	!
Inclusão da Licença	*	*	!	!	!	!	!	!	*	!	!
Distribuição nos Mesmos Termos	*	*	*	*	!	!	!	!	!	!	!
Declaração das Mudanças	*	*	*	!	!	*	*	*	!	!	!
Uso de Patentes	*	*	*	*	*	✓	*	*	*	✓	✓

*Não especificado na licença.

✓ PERMITIDO ✗ PROIBIDO ! OBRIGATÓRIO

Figura 2, Fonte: Elaborado pelo Autor.

Defensores da GPL argumentam que softwares livres devem continuar livres para beneficiar o maior número de pessoas na comunidade. Richard Stallman criou, em 1984, a licença GPL. Seu principal objetivo era impedir que códigos fonte de softwares livres fossem licenciados sob termos proprietários após serem modificados por organizações comerciais (fenômeno conhecido como *hijacking*). Essa defesa se concretiza na inclusão de uma cláusula de *copyleft* nos termos da licença. Essa cláusula define que, quando versões modificadas do programa forem distribuídas, o código fonte deste deverá ser distribuído aberto e sob os mesmo termos do programa original. A GPL, por ser uma licença de *copyleft* forte, se mostra menos atrativa para atores comerciais. Apesar disso, ela é uma ferramenta que propõe manter a sustentabilidade do software livre.

Do outro lado da moeda, está a: Berkeley Software Distribution da Universidade da Califórnia, Berkeley. Ela permite o livre uso, modificação e distribuição sem nenhuma obrigação recíproca de copyleft. Em geral, licenças permissivas como a BSD, tornam os softwares bens quase-públicos e qualquer pessoa pode utilizá-los para qualquer fim, até mesmo para incorporação do código fonte em softwares proprietários.

A forte aceitação da GPL 2.0 pela comunidade de desenvolvedores e usuários nos anos 1990 e 2000 levou à criação de um “ambiente uniforme de licenciamento” (PHIPPS, 2012), gerando uma gama de softwares interoperáveis tanto no nível de código quanto no de compartilhamento (LINDBERG, 2008). O impacto da introdução do *copyleft*, imprescindível para muitos projetos de software livre, perdura até os dias atuais e a GPL ainda figura como a licença mais popular de software livre em diversos cenários. Por esses motivos, foram enfatizadas neste estudo as análises das licenças da família GNU GPL.

Como todo ambiente colaborativo, o SPB necessita das contribuições de voluntários para desenvolver suas soluções e se beneficiar das vantagens do modelo de software livre. Além disso, ao valer-se de tecnologias abertas, os gestores do Portal SPB devem estar cientes dos impactos das diversas políticas de licenciamento no ecossistema de desenvolvimento e na governança do software público. A figura 3 mostra as 20 licenças mais utilizadas em projetos de software livre. A GPL 2.0 é ainda hoje a mais utilizada.

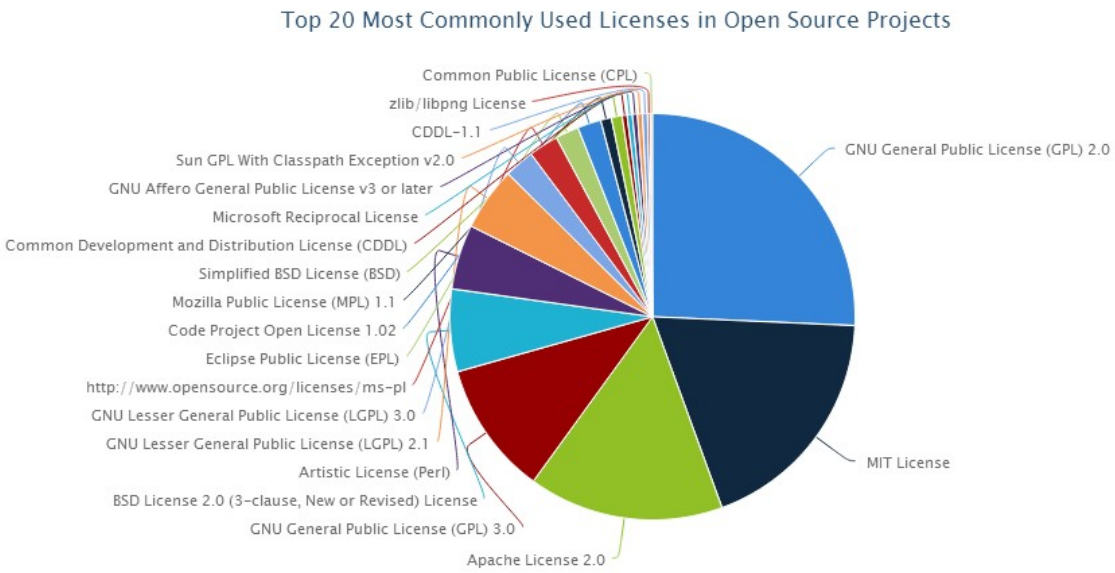

Figura 3, Fonte: BLACKDUCK, 2014.

5. Impacto do Licenciamento no Desenvolvimento de Software Livre

A escolha da licença interfere na atração e repulsão de usuários e colaboradores de projetos de software livre (HASTENREITER, 2013; SANTOS JR. *et* al., 2011). As condições impostas pelas licenças são responsáveis por parte da motivação dos desenvolvedores e usuários interessados em contribuir para o projeto (STEWART *et* al., 2006). O estudo de Lerner e Tirole (2005) encontrou resultados consistentes com a hipótese de que licenças mais restritivas são mais adotadas quando o software é direcionado a usuários finais, enquanto que licenças menos restritivas são frequentemente adotadas para projetos orientados para desenvolvedores, para a Internet ou sistemas operacionais proprietários. Os autores sugerem que licenças mais restritivas, ao proteger o software de apropriações proprietárias, protegem o trabalho realizado e dão mais segurança aos desenvolvedores, enquanto que licenças não restritivas permitem que cada colaborador persiga seus objetivos com mais autonomia. Hastenreiter (2013) aponta que licenças não restritivas avançam mais rapidamente do estágio pré-alpha para o estágio beta. Fershtman e Gandal (2007) encontraram evidências na mesma direção: a atividade dos colaboradores (mais especificamente o número de linhas de código fonte) em projetos de software livre é maior quando as licenças são menos restritivas e mais orientadas comercialmente.

Os resultados de Colazo e Fang (2009) encontram evidências distintas. Em sua amostra, projetos com licenças restritivas ou altamente restritivas possuem maior número de desenvolvedores, maior nível de atividade e menor retenção de colaboradores.

Além de impactar na atração e na atividade de desenvolvedores, licenças afetam a atratividade de uma solução. O conceito de atratividade desenvolvido por Carlos Jr. envolve o efeito das características do projeto na qualidade do software, mensurada pela “capacidade de captar voluntários, promover visitas ao portal do projeto e downloads do software para teste e uso” (SANTOS JR., p.427, 2010).

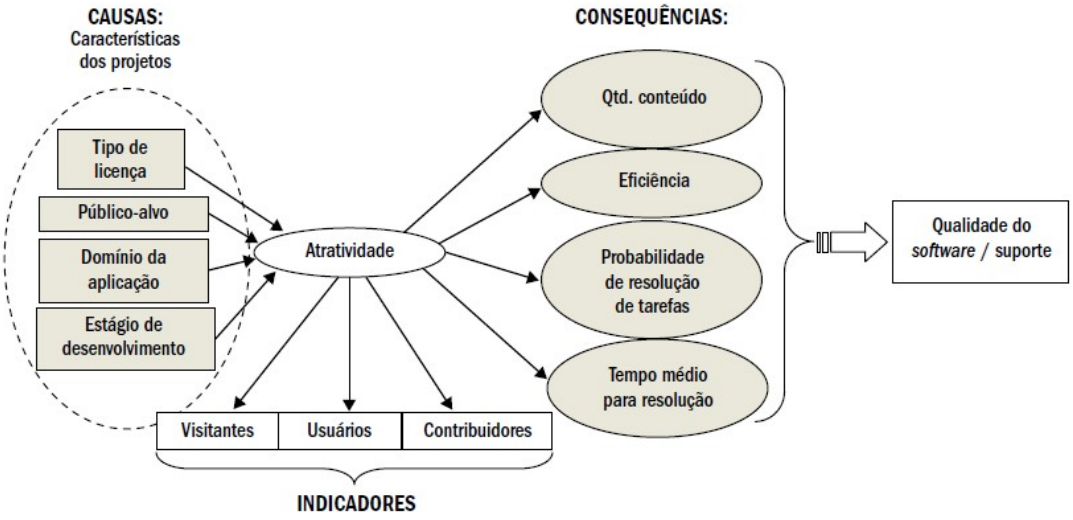

Figura 4, Fonte: SANTOS JR., 2010

O modelo (Figura 4) apresenta o impacto que o tipo de licença produz, junto com outros fatores, no interesse de usuários em experimentar um software livre e de contribuidores em participar ativamente do desenvolvimento. O mesmo se aplica às soluções do Portal SPB. Por meio da SLTI, a aprovação de novos modelos de licença pode impactar positivamente no aumento da atratividade das soluções atuais e potenciais, ocasionando uma qualidade superior.

5.1. A família GPL

A GNU General Public License (GPL), criada em 1989 pela FSF (Free Software Foundation), foi a primeira licença com cláusula de *copyleft*. Defensores da GPL argumentam que softwares livres devem continuar livres para beneficiar o maior número de pessoas na comunidade. Como mencionado por Stallman, fundador da FSF:

“*Copyleft* usa a lei de copyright, mas inverte-a para servir ao oposto do propósito usual: Ao invés de significar a privatização do software, ela se torna um meio de manter o software livre” (STALLMAN, p.5, 1999, tradução nossa).

5.1.1. General Public License 2.0

O objetivo da GPL é descrito em seu preâmbulo (http://opensource.org/licenses/gpl-2.0) que, apesar de não ter valor jurídico, serve como base para o resto do documento: “A GNU GPL se destina a garantir sua liberdade de compartilhar e modificar software livre para garantir que o software seja livre para todos os seus usuários”.

Na seção 0, encontram-se as definições: “Programa” é definido no texto da licença como “qualquer programa ou trabalho que contenha um aviso colocado pelo detentor dos direitos autorais informando que aquele pode ser distribuído sob os termos desta GPL”; "Trabalho baseado no Programa" é definido como o “Programa e qualquer trabalho derivado sob lei de direitos autorais, ou seja, um trabalho contendo o Programa ou porção dele”. Aqui se encontra uma passagem confusa, pois trata de forma idêntica trabalhos que contêm o programa e trabalhos derivados do programa, diferentemente da legislação autoral (TSAI, 2008; LINDBERG, 2008) que, no caso das obras coletivas, confere direitos patrimoniais ao organizador da obra.

Na seção 1, enunciam-se as condições para que cópias em código fonte do Programa sejam distribuídas. Devem ser explícitos: os avisos de direitos autorais, quaisquer avisos que se refiram à licença e à ausência de garantias e uma cópia da licença. É permitida inclusive a cobrança de taxas pela transferência da cópia e, caso seja oferecida, pela garantia.

A seção 2 do documento trata das condições para modificação do Programa e subsequente distribuição. Primeiramente, é exigido que arquivos modificados contenham avisos anunciando as mudanças e respectivas datas: “You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change”.

A seguir, há a cláusula que determina o *copyleft* da licença. No entanto, novamente trata-se trabalho coletivo e derivado de forma similar, exigindo reciprocidade em ambos:

> “You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License”.

Ao exigir reciprocidade sob trabalhos que contenham todo ou parte do Programa, a licença reivindica direitos sob obras que, pela ótica da Lei de direitos autorais, pertencem ao organizador e não ao titular da obra original.

A próxima cláusula é um complemento da anterior e assegura que os licenciados sejam informados da reciprocidade da licença, dos direitos autorais, da não garantia e da localização da cópia da Licença:

> “If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)”

Também na seção 2, o alcance da reciprocidade da GPL é descrita com base no modo de distribuição do software:

> “These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it”.

Entende-se pelo trecho acima, que pedaços de código razoavelmente independentes não derivados do Programa sob GPL que sejam distribuídos separadamente deste não estão no escopo da licença. No entanto, não se define o que é um critério razoável para considerar a independência de um software. Além disso, o texto deixa claro que, caso haja distribuição de um trabalho como parte de um todo, configura-se trabalho baseado e o licenciamento do software sob GPL é obrigatório. Por essa razão, costuma-se sugerir que trabalhos com licenças incompatíveis sejam distribuídos separadamente e que os usuários sejam instruídos a obter as partes (SABINO; KON, 2009).

O próximo parágrafo diz que a intenção da seção é “exercer o direito de controlar a distribuição de trabalhos derivados e coletivos baseados no “Programa”, novamente usando termos que confundem a interpretação em face da lei e carecendo da definição de “trabalho coletivo baseado no Programa”.

No último parágrafo da seção 2 sinaliza-se que a mera distribuição conjunta com software GPL não estende o escopo da licença a softwares não GPL:

> “In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License”.

A seção 3 estabelece a obrigação indispensável de que o código fonte esteja disponível para os licenciados, garantindo a possibilidade de modificação futura por parte dos licenciados.

As seções 4 e 5 explicitam que a única maneira de exercer os direitos da licença é por meio da total submissão aos termos dela e que ao distribuir ou modificar o Programa aceita-se automaticamente suas condições.

A seção 6 determina que qualquer receptor do Programa recebe automaticamente uma licença do autor original. Está presente aqui uma cláusula muito importante que proíbe a imposição de quaisquer restrições aos direitos concedidos pela licença que não estejam presentes na licença. Ou seja, qualquer condição referente à modificação ou à distribuição que não seja equivalente às condições da GPL 2.0 não é compatível com esta licença.

A seção 7 determina que, caso os termos da licença entrem em conflito com decisões judiciais, ou quaisquer condições, referentes a infrações de direitos de patente ou não, o software não deve ser distribuído. É citado no texto o seguinte exemplo:

> “if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program”.

O período acima é considerado por alguns como uma licença implícita de patentes, mas há controvérsias, principalmente em relação ao seu escopo, ou seja, se essa licença de patente estende-se somente ao software original ou também a versões modificadas dele. A seção 7 é finalizada impondo que, caso a distribuição não atenda

simultaneamente aos termos da licença e de quaisquer outras obrigações do licenciado, a distribuição não é permitida.

De forma sintética, o texto da GPL 2.0 procura estender seu escopo a trabalhos que contenham ou modifiquem o original, salvo quando estes forem meras agregações em volumes de armazenamento ou meio de distribuição. Ainda que seja vislumbrada a intenção do redator da licença em distinguir “meras agregações” de “trabalhos coletivos baseados no Programa” que formem um todo coerente, na prática a distinção é intrincada e entra possivelmente em conflito com a Lei de Direitos autorais (9.610/98). Para resolver essa ambiguidade, é comum usar a distinção entre ligação estática (*static linking*) e ligação dinâmica (*dynamic linking*) para definir o escopo da GPL 2.0. Uma ligação estática incorpora dois códigos para a formação de um programa maior, um executável; uma ligação dinâmica estabelece uma relação na qual um código é referenciado por outro, mas ambos funcionam independentemente (TSAI, 2008). Nesse sentido, argumenta-se que os termos de um software GPL não incidem sob outro ligado a este de forma dinâmica. Porém, a posição da FSF é divergente e está exposta em um trecho do FAQ a respeito da GPL 2.0:

> “What is the difference between “mere aggregation” and “combining two modules into one program”?
>
> Mere aggregation of two programs means putting them side by side on the same CD-ROM or hard disk. We use this term in the case where they are separate programs, not parts of a single program. In this case, if one of the programs is covered by the GPL, it has no effect on the other program.
>
> Combining two modules means connecting them together so that they form a single larger program. If either part is covered by the GPL, the whole combination must also be released under the GPL—if you can't, or won't, do that, you may not combine them.
>
> What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).
>
> If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a

> shared address space, that almost surely means combining them into one program.
>
> By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program".

A GPL 2.0 é uma licença complexa e com incongruências entre seu texto e legislação vigente, além de dificuldades de interpretação (SABINO; KON, 2009). Isso provoca uma grande preocupação de que a simples combinação de código sob GPL com outro software torna obrigatório o licenciamento deste sob GPL, impedindo que alguns desenvolvedores utilizem código GPL em certos contextos. Essas preocupações levaram ao desenvolvimento da Library General Public License ou Lesser General Public License (LGPL).

5.1.2. General Public License 3.0

A versão GPL 3.0 (http://opensource.org/licenses/GPL-3.0), lançada em 2007, trouxe mais robustez ao texto da licença. Ao passar por um longo período de sugestões e melhorias por parte da comunidade, a GPL se tornou sólida, estando alinhada com grande número de legislações, assim como a brasileira (SABINO; KON, 2009). Com efeito, a grande vantagem da GPL 3.0 é a adequação de seu texto à legislação autoral, facilitando inclusive uma internacionalização menos problemática da GPL. A correção de ambiguidades em seu texto que geravam múltiplas interpretações também foi um avanço a ser destacado.

Uma das principais razões que levaram a FSF a alterar a General Public License foi o combate à tivoização (http://www.gnu.org/licenses/rms-why-gplv3.html). Segundo a FSF, a tivoização é a utilização de licenças de software livre, mas com restrições técnicas que impossibilitem a execução de versões modificadas, criando uma situação de monopólio do código fonte utilizável. A proteção do usuário inclui-se na seção 6, na qual é exigida, na distribuição de softwares com "produto de usuário", a presença de "Informação de Instalação" que seja suficiente para garantir que "o funcionamento contínuo do código objeto modificado não seja prevenido ou interferido somente porque

a modificação foi feita". A intenção é impedir que o hardware rejeite versões modificadas de software livre, suprimindo, na prática, a liberdade de uso dos licenciados.

Na seção 0, são listadas as definições mais importantes do texto.

A definição de "Programa" sofreu leves alterações. Os termos "trabalho derivado" e "trabalho coletivo" foram eliminados do texto da licença, possibilitando que os termos da licença sejam analisados à luz de seu próprio texto e evitando conflitos com a legislação. O trabalho resultante de uma modificação é chamado de "versão modificada" do trabalho anterior ou trabalho "baseado" no trabalho anterior. Modificar um trabalho é definido como "copiar ou adaptar a totalidade ou parte do trabalho de forma que requeira permissão de direitos autorais, salvo a realização de uma cópia exata". A palavra "distribuição" também foi retirada das partes importantes da licença para evitar ambiguidades. São utilizados os termos "propagar" (propagate) e "transmitir" (convey). Segundo a redação da licença, propagar um trabalho significa fazer qualquer coisa com ele que, sem permissão, infringiria a lei de direitos autorais (exceto executá-lo em um computador ou modificar uma cópia privada), incluindo cópia, distribuição (com ou sem modificação) e disponibilização ao público. Transmitir é descrito como qualquer tipo de propagação que permita a outras partes fazer ou receber cópias (a mera interação com um usuário por meio de rede de computadores sem transferência de uma cópia não é considerada transmissão).

Ao "transmitir" versões modificadas de código sob GPL 3.0, é a Fonte Correspondente que deve ser disponibilizada junto com o código objeto. Na seção 1, define-se "Fonte Correspondente" como "todo código fonte necessário para gerar, instalar e (no caso de um programa executável) rodar o código objeto e modificar o trabalho, incluindo scripts que controlem essas atividades". Inclui-se uma exceção da reciprocidade da licença para bibliotecas de sistema e ferramentas de uso geral ou geralmente disponíveis de modo livre que são usados sem modificação na realização dessas atividades, mas que não fazem parte do trabalho.

É dado, ainda na seção 1, um tratamento expresso a respeito do escopo da licença e de ligações dinâmicas entre softwares, questão que gerou muita controvérsia em relação à GPL 2.0. A redação do texto esclarece que o escopo da GPL 3.0 estende-se a "arquivos de definição de interface associados com arquivos fonte do trabalho, e código

fonte de bibliotecas compartilhadas e subprogramas ligados dinamicamente, os quais o trabalho é especificamente designado para requerer, tais como pela comunicação íntima de dados ou fluxo de controle entre esses subprogramas e outras partes do trabalho". A posição da FSF é formalizada, clarificando que existe a possibilidade de ligações dinâmicas estarem no escopo da GPL. Entretanto, a imprecisão não é totalmente resolvida, pois não se define o que seria considerado "comunicação íntima" (TSAI, 2008).

A seção 5 é a versão reescrita da seção 2 da GPL 2.0. Trata das exigências para distribuição de versões modificadas. As condições são: a presença de avisos proeminentes das modificações realizadas e respectivas datas; o trabalho como um todo deve ser licenciado sob os termos da GPL 3.0 e quaisquer termos adicionais presentes na seção 7, sendo necessária a presença de aviso declarando isso; caso o trabalho contenha interfaces de usuário interativas que mostrem avisos legais, devem ser exibidos os avisos legais apropriados.

O último parágrafo, tenta esclarecer que a "contaminação" da GPL não se dá por proximidade física, mas por intimidade de comunicação entre os códigos:

> "A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate".

Outro ponto importante da licença GPL 3.0 é a inclusão de uma licença expressa de patente em seu texto, seção 11:

> "Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version".

O objetivo é impedir que o desenvolvimento de software livre seja fragilizado pela proteção de patentes de software e evitar a incerteza em relação a problemas judiciais. Ademais, seu texto proíbe a distribuição sob GPL 3.0 caso haja acordo de licença de

patentes que não inclua em seu escopo, proíba ou esteja condicionado ao não exercício de um ou mais direitos garantidos na GPL.

A seção 7 foi crucial para que a GPL se tornasse compatível com licenças não GPL, pois torna possível que permissões adicionais, definidas como “termos que suplementam os termos desta Licença, criando exceções para uma ou mais das suas condições”, sejam concedidas. Nessa seção, foram incluídas cláusulas que autorizam a adição de permissões adicionais e de condições adicionais a respeito de garantias e responsabilidades; exigência de avisos legais ou atribuições de autoria; proibição de representação falsa da origem do material ou requerimento que versões modificadas sejam marcadas como diferentes da original; limitação do uso dos nomes dos licenciantes e autorias para fins publicitários; recusa à cessão de direitos de marca; e requisição de indenização de licenciantes e autores em caso de estabelecimento de responsabilidades contratuais que se imponham a estes.

A seção 7.b permite a proteção de atribuição de autoria, de modo compatível com a proteção dos direitos morais, consumada na legislação autoral brasileira: “[...] preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it.”

No terceiro parágrafo da seção 10, exige-se que o licenciado não imponha nenhuma restrição adicional ao exercício dos direitos concedidos pela licença; por exemplo, é proibido que sejam cobradas taxas de royalty ou qualquer outro custo, assim como iniciados litígios alegando infração de direitos de patente.

Merece destaque a inclusão da MPL 2.0 e Apache 2.0 na lista de licenças compatíveis. Apesar de incompatível com a versão GPL 2.0, é possível que softwares sob versões anteriores da GPL sejam redistribuídos e/ou modificados sob os termos da GPL 3.0. Para isso é necessário que haja o seguinte aviso na distribuição do programa:

> “This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version”.

5.1.3. Lesser General Public License 2.1

Ao criar a Lesser General Public License (LGPL), cuja versão 2.1 (http://opensource.org/licenses/LGPL-2.1) data de 1999, Stallman desenvolveu uma versão mais permissiva da GPL, mais palatável para organizações e desenvolvedores e

que se ajusta melhor às bibliotecas de software que se relacionam com diversos softwares e necessitam de maior compatibilidade. Seu principal objetivo era permitir que bibliotecas fossem ligadas de forma dinâmica a aplicações que não fossem licenciadas sob GPL. De fato, seu texto permite explicitamente a junção com software sob outras licenças (inclusive proprietárias).

Assim como a GPL 2.0, as condições e restrições da LGPL 2.1 se aplicam somente às atividades de cópia, modificação e distribuição.

A seção 0 da licença contém as definições utilizadas no restante do texto. Uma “biblioteca” é definida como “uma coleção de funções de software e/ou dados preparados de forma a serem convenientemente ligados com aplicações (que usam algumas dessas funções e dados) para formar executáveis”.

A “Biblioteca” “refere-se a qualquer biblioteca de software ou trabalho que tenha sido distribuído sob os termos da LGPL”.

Um “trabalho baseado na Biblioteca” significa “tanto a Biblioteca quanto qualquer trabalho derivado sob a lei de direitos autorais: ou seja, um trabalho contendo a Biblioteca ou uma porção dela, literalmente ou com modificações e/ou traduzida diretamente para outra língua”. Similarmente ao que acontece com o texto da GPL 2.0, trabalhos que contenham ou modifiquem a Biblioteca são ambos considerados trabalhos derivados pelo texto da licença.

Uma cláusula importante é a 2(a) que permite a modificação e distribuição sob os termos da LGPL apenas caso o trabalho modificado seja uma biblioteca. No entanto, caso o trabalho derivado não seja uma biblioteca, a licença permite que este seja distribuído sob GPL, requerendo somente a alteração dos avisos que se refiram à licença do software.

Os últimos parágrafos da seção 2 são essencialmente equivalentes aos da GPL 2.0.

A seção 3 permite que o licenciado aplique os termos da GPL a uma cópia da Biblioteca, alterando todos os avisos da licença. Essa cláusula, ao admitir o licenciamento de cópias sob GPL, permite a distribuição de trabalhos derivados que não sejam bibliotecas. Importante frisar que essas mudanças realizadas sobre uma cópia da Biblioteca são irreversíveis e a GPL passa a se aplicar a ela e todos os trabalhos derivados e cópias subsequentes.

Na seção 4 está presente a exigência de disponibilização do código fonte sempre que seja distribuída a Biblioteca, versões modificadas ou partes dela:

> “You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange”.

A seção 5 exclui “trabalhos que usam a Biblioteca”, quando isolados, do escopo da LGPL:

> “A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License”.

No entanto, ligar um “trabalho que usa a Biblioteca” com a Biblioteca cria um executável que está sujeito às condições presentes na seção 6:

> “However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.”

A seção 6 informa as condições para a distribuição da combinação ou ligação entre um “trabalho que use a Biblioteca” e a Biblioteca:

> “6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications”.

O restante da seção 6 esclarece que é necessária a presença de permissão para modificação e engenharia reversa apenas da Biblioteca. Ou seja, a condição do trecho acima não se refere ao “trabalho que usa a biblioteca”. O objetivo é que o usuário possa modificar a Biblioteca livremente e recombiná-la com o código objeto do “trabalho que usa a Biblioteca”.

Além disso, deve ser explícito o aviso, junto com todas as cópias do trabalho, de que a Biblioteca é usada nele e que ela está sob os termos da LGPL. Para um executável, a

LGPL exige que este inclua dados e quaisquer componentes necessários para o funcionamento da Biblioteca a menos que esses componentes já estejam incluídos no sistema operacional sobre o qual o programa combinado é destinado para ser executado.

Na seção 7 é abordado o caso da formação de uma única biblioteca formada por um "trabalho baseado na Biblioteca" e uma biblioteca sob outra licença. As condições são: deve ser permitida a distribuição separada dos dois trabalhos; uma cópia do "trabalho baseado na Biblioteca" deve estar disponível separadamente sob a LGPL; e deve haver aviso proeminente do fato de que parte da biblioteca combinada é um "trabalho baseado na Biblioteca" e explicação de onde é possível obter a forma não combinada do mesmo trabalho.

O restante do texto se assemelha muito às seções 4 a 12 da GPL 2.0.

5.1.4. Lesser General Public License 3.0

Assim como a GPL, em 2007 a LGPL ganhou uma nova versão (http://opensource.org/licenses/LGPL-3.0). De forma geral, a licença foi reescrita e incorporou as condições da GPL 3.0, mas incluiu permissões adicionais para que o escopo do *copyleft* fosse mantido:

> "This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below".

As definições básicas estão presentes na seção 0. A primeira mudança é em relação à "Biblioteca". Da definição original foram removidos os trechos incongruentes com a legislação autoral.

"Trabalho baseado na biblioteca" é denominado, na nova versão, "Aplicação": "An 'Application' is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library.

Por sua vez, "Trabalho que usa a biblioteca" passa a ser chamado de "Trabalho Combinado":

> A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

"Código fonte" passa a ser denominado "Fonte Correspondente Mínima":

> The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

O “Código Correspondente à Aplicação” significa o “código objeto e/ou código fonte da Aplicação, incluindo quaisquer dados e programas utilitários necessários para reproduzir o “Trabalho Combinado da Aplicação”, mas excluindo as “Bibliotecas de Sistema” do “Trabalho Combinado” (tradução nossa).

De acordo com a seção 2, a distribuição de versões modificadas da biblioteca é permitida sob os termos da LGPL 3.0 ou da GPL.

A Seção 4 estabelece as condições para a distribuição de “Trabalhos Combinados”: (a) Aviso proeminente em cada cópia do Trabalho Combinado do fato que a Biblioteca é usada nele e está coberta pela LPGL; (b) Cópia da GPL e da LGPL 3.0; (c) caso o “Trabalho Combinado” exiba avisos de direitos autorais durante a execução, inclusão do aviso de direitos autorais da Biblioteca, assim como referência direcionando o usuário às cópias da GPL e a LGPL 3.0; (d) Fazer um dos dois: 1. Transmissão da “Fonte Correspondente Mínima” sob os termos da LGPL 3.0 e o “Código Correspondente à Aplicação” de forma que permita que o usuário recombine ou religue a “Aplicação” com uma versão modificada da “Versão Ligada” para produzir uma versão modificada do “Trabalho Combinado”,de maneira especificada na seção 6 da GNU GPL para transmissão de “Fonte Correspondente”; 2. Uso de um mecanismo de biblioteca compartilhada que utilize cópia de Biblioteca já presente no sistema do usuário e que opere corretamente com versão modificada da Biblioteca que seja compatível com interface da “Versão Ligada”; (e) Fornecer informações de instalação, mas só se o licenciado já seria obrigado a fornecer tais informações de acordo com a seção 6 da GPL.

A seção 5 é semelhante à seção 7 da versão anterior e esclarece a combinação da Biblioteca com outras bibliotecas. Para que seja realizada a combinação e distribuição sob termos desejados pelo licenciante é necessário que: uma cópia do “trabalho baseado na Biblioteca” esteja disponível separadamente sob a LGPL; haja aviso proeminente do fato de que parte da biblioteca combinada é um “trabalho baseado na

Biblioteca” e a explicação de onde é possível obter a forma não combinada do mesmo trabalho.

A seção 6 é correspondente à seção 13 da versão anterior. Nela, é exposta a possibilidade de se utilizar a versão da LGPL acrescida do trecho “or any later version” (ou qualquer versão posterior). Isso significa que licenciados podem escolher entre seguir os termos de versões mais atuais da LGPL sem necessidade de obter outra licença.

Os termos utilizados na redação seguem a mesma linha da GPL 3.0. O termo “distribuição” foi retirado do texto em favor do verbo “transmitir” (convey). A versão 3.0 da LGPL se mantém como recíproca parcial e, pelo fato de absorver os termos da GPL 3.0, é mais enxuta que a versão anterior.

6. Compatibilidade entre Licenças de Software

Durante o desenvolvimento de um software, é possível que se pretenda combinar partes de dois ou mais programas em um só. Caso esses programas estejam sob licenciamentos distintos, é necessário checar se a fusão e distribuição não infringem os termos das licenças envolvidas. Caso nenhuma violação seja desencadeada, diz-se que as licenças são compatíveis (LAURENT, 2004). Exemplificando, se uma licença X é compatível com uma licença Y, isso significa que é possível um software sob licença X e um software sob licença Y serem incorporados e distribuídos de alguma forma sem que os termos de quaisquer licenças envolvidas sejam transgredidos.

A compatibilidade deve ser analisada em duas situações: No caso de trabalhos coletivos e no caso de trabalhos derivados.

De acordo com a Lei de direitos autorais (9.610/98) cabe ao autor de um trabalho “utilizar, fruir e dispor” da sua obra. Como o regime do software é o mesmo aplicado às obras literárias, conclui-se que os direitos enumerados nos artigos 28 e 29 da Lei se aplicam igualmente aos detentores de direitos autorais de programas de computador. Modificações, portanto, do software original, dependem de autorização, pois caracterizam obras derivadas. Direitos patrimoniais de obras coletivas, por outro lado, pertencem ao organizador da obra segundo o § 2º do artigo 17 da Lei: “cabe ao organizador a titularidade dos direitos patrimoniais sobre o conjunto da obra coletiva”. Exemplificando, caso dois softwares independentes sejam apenas agrupados e

distribuídos, mas não constituam um único software, a licença de um software não afeta o outro. Ainda assim, o texto da GPL 2.0, por exemplo, não deixa claro o que considera como trabalho coletivo ou derivado, problema que foi em parte corrigido na versão GPL 3.0. No caso de dúvida, é mais seguro, se factível, contactar o(s) detentor(es) dos direitos autorais e pedir permissão para a distribuição.

Ao passo que trabalhos coletivos têm uma compatibilidade de análise relativamente simples, quando o trabalho é derivado de um ou mais softwares, a compatibilidade é mais delicada, pois depende das permissões das licenças.

Antes de analisar a compatibilidade de licenças envolvendo combinações de software, são necessários alguns esclarecimentos. Primeiramente, ainda que a presença de *copyleft* e sua força seja o maior entrave à compatibilidade entre licenças, todas as condições impostas pelas licenças devem ser analisadas para se ter uma resposta segura quanto validade da conciliação entre duas ou mais licenças. Segundo, a compatibilidade é direcional (MORIN *et al*, 2012), ou seja, depende do sentido da combinação de dois ou mais softwares. Caso o software da licença analisada esteja incorporando código de outra licença, o sentido da compatibilidade é montante ou para trás. Caso o software esteja migrando para licença de outro código, o sentido é jusante ou para frente. Em outras palavras, no sentido montante, avalia-se se o código recebido pode ser licenciado sob os termos do software em questão; no sentido jusante, é considerada a possibilidade do software em questão ser licenciado sob os termos de outro software com o qual será combinado. Por exemplo, a GPL 2.0 é compatível com BSD 3-cláusulas apenas no sentido montante, porque apesar de ser possível código BSD 3-cláusulas tornar-se código GPL 2.0, o inverso não é possível.

Figura 5, Fonte: Elaborada pelo autor.

No caso tradicional de software proprietário os direitos oferecidos aos licenciados não permitem que o software seja modificado nem distribuído. Nas licenças de software livre o tratamento é distinto e o código fonte é disponibilizado para que o usuário

modifique-o e distribua-o. Ainda assim cada licença exige certas condições para que os direitos sejam concedidos aos licenciados.

Licenças permissivas geralmente não enfrentam graves problemas de compatibilidade a jusante. Um software sob BSD ou MIT, por exemplo, pode ser modificado e distribuído sob várias licenças (inclusive proprietárias). Todavia, deve-se atentar aos detalhes, já que as condições impostas por essas licenças (normalmente a presença de avisos de direitos autorais ou proteção de marcas e nomes) devem ser atendidas e não podem entrar em conflito com a licença do trabalho compilado. No sentido montante, a priori, não existe compatibilidade de licenças permissivas com licenças *copyleft*, devido à reciprocidade destas.

No caso de licenças *copyleft* ou recíprocas, trabalhos derivados devem preservar a licença do trabalho original, salvo na presença de cláusulas que autorizem o contrário, como no caso da MPL 2.0 que permite um duplo licenciamento de combinações envolvendo softwares sob outra licença previamente determinada; ou no caso da LGPL 2.1 que permite sublicenciamento sob GPL. Isto é, de forma generalista, licenças recíprocas são compatíveis apenas com si mesmas a jusante. A diferença entre reciprocidade parcial e total se dá na possibilidade de ligação entre softwares de diferentes licenças. Licenças recíprocas parciais não estendem seu escopo sobre todo o software no caso de combinações. Exemplificando, ao ligar uma biblioteca LGPL (recíproca parcial) com um software proprietário, somente o código proveniente da biblioteca precisa estar licenciado sob LGPL. Licenças recíprocas totais, por outro lado, “infectam” o software na íntegra. Ou seja, ao incorporar uma biblioteca GPL (recíproca total) a um software sob outra licença, a distribuição só pode ser realizada sob GPL, o que, muitas vezes, acarreta na impossibilidade de cumprimento de ambas as licenças. Por esse motivo, a compatibilidade envolvendo GPL é mais complicada. Enquanto que, a jusante, licenças recíprocas são compatíveis com a maioria das soluções permissivas e algumas das recíprocas, a montante não há compatibilidade com alternativas de licenciamento, exceto em casos especiais.

6.1. Compatibilidade envolvendo GPL 2.0

A compatibilidade da GPL 2.0 é obstruída em muitos casos pela cláusula contida na seção 6: “You may not impose any further restrictions on the recipients' exercise of the rights granted herein”.

Essa condição impede que uma licença que contenha qualquer condição não presente na GPL 2.0 que restrinja direitos garantidos por ela se torne automaticamente incompatível. O efeito é negativo, pois muitas vezes as restrições que causam incompatibilidade não vão de encontro ao *copyleft* da GPL. Por exemplo, a versão original da BSD de 4 cláusulas é incompatível somente devido à cláusula de propaganda; A Apache 2.0 é inconciliável devido às cláusulas de indenização e de terminação da licença em caso de processo judicial alegando infração de direitos de patente. Das licenças permissivas mais utilizadas, a GPL é compatível com a MIT e as versões BSD 3-cláusulas e BSD 2-cláusulas (http://www.gnu.org/licenses/license-list).

Das licenças recíprocas parciais, a mais fácil de compatibilizar com GPL 2.0 é a LGPL 2.1, pois está expressa em sua seção 3 uma autorização para que trabalhos sob seus termos sejam distribuídos sob qualquer versão da GPL. A LGPL 3.0, no entanto, não possui essa cláusula e é incompatível com GPL 2.0. A Mozilla Public License 2.0 é compatível devido à cláusula de sua seção 3.3 que permite combinação com softwares GPL desde que uma cópia dos códigos MPL 2.0 sejam disponibilizada sob os termos originais, possibilitando que o receptor do programa escolha entre as duas licenças (http://opensource.org/licenses/MPL-2.0).

Sempre houve muita controvérsia a respeito do escopo da GPL no caso de combinações entre softwares GPL e não GPL. Sabino (2009) define que os dois pontos críticos para analisar o alcance da GPL são:

> “o quanto um software é derivado do outro e como eles são distribuídos. Se um software é razoavelmente independente do outro que está sob a GPL e se eles são distribuídos separadamente, é seguro afirmar que ele não estará sujeito às condições da GPL, e, portanto não há preocupações referentes à compatibilidade de licenças”.

Como já apontado, não há uma interpretação definitiva em relação ao que seria software razoavelmente independente.

6.2. Compatibilidade envolvendo GPL 3.0

A seção 6 da versão 2.0 foi reescrita e aparece na seção 10:

> “You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it”.

Além de proibir restrições adicionais aos direitos concedidos aos licenciados, é incluída uma licença expressa de patente, o que proporciona maior proteção aos usuários e compatibilidade com licenças como Apache 2.0 e MPL 2.0. Apesar de continuar vedando restrições que não estejam presentes na GPL, a inclusão da seção 7 abriu uma série de exceções visando à conciliação com a Apache. De fato, as subseções 7(e) e 7(f) autorizam, respectivamente, cláusulas de proteção à marca e de indenização presentes na Apache 2.0 (http://opensource.org/licenses/Apache-2.0), tornando a GPL 3.0 compatível com Apache 2.0. Além da Apache 2.0, estão na lista de licenças permissivas compatíveis a montante, por exemplo, as licenças MIT, BSD 3-cláusulas e BSD 2-cláusulas (http://www.gnu.org/licenses/license-list). Dentro da família GPL, há compatibilidade a montante com LGPL 2.1, devido à cláusula que permite sublicenciamento sob qualquer versão GPL e com LGPL 3.0. Reafirma-se que GPL 2.0 não é compatível com GPL 3.0 em razão da presença de condições presentes apenas na versão anterior, o que viola a seção 6 da versão 2.0. No entanto, a versão 3.0 tem compatibilidade a montante com softwares licenciados sob a versão “GPL 2.0 ou versão posterior”.

Por parte da MPL 2.0, a compatibilidade se dá em sua cláusula da seção 3.3 que permite licenciar adicionalmente um trabalho gerado a partir da combinação entre software MPL 2.0 e software sob uma licença secundária (GPL versão 2.0, LGPL 2.1, AGPL 3.0 ou qualquer versão posterior dessas licenças).

Os dois diagramas abaixo (Figura 6) mostram a compatibilidade das principais licenças de software livre com as versões 2.0 e 3.0 da GPL no caso de combinação.

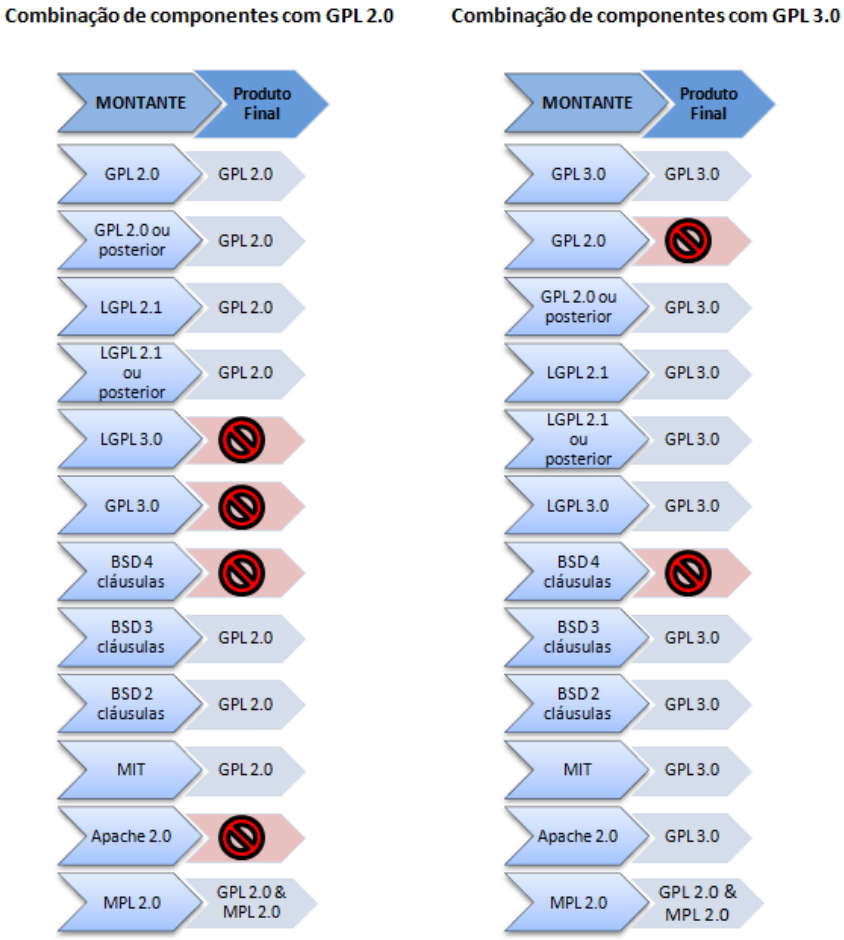

Figura 6, Fonte: Elaborada pelo autor

7. Diagnóstico da Instrução Normativa 01/2011

A Instrução Normativa 01/2011 dispõe sobre o SPB e estabelece as condições para seu funcionamento. A análise preliminar do Software Público Brasileiro apontou, na Instrução Normativa 01/2011, a presença de imprecisões que dificultam a sua interpretação e de barreiras que prejudicam a expansão do impacto do SPB.

O software público é definido em seu artigo 2º como um software livre, com licença livre aceita pelo Órgão Central do SISP, que tem sua identidade original entre o seu nome, marca, código-fonte, documentação e outros artefatos relacionados protegidos pela Licença Pública de Marca (LPM) e é disponibilizado publicamente em ambiente virtual. A LPM é definida como (art.3º, inciso V):

> "licença de uso de marca que preserva a identidade original entre o nome, a marca, o código-fonte, a documentação e outros artefatos relacionados ao Software Público Brasileiro e na qual o titular do registro consente genericamente, sem necessidade de qualquer tipo de autorização prévia e/ou

> específica, que outros utilizem gratuitamente a marca para fins de cópia, distribuição, compartilhamento e transmissão em qualquer dispositivo físico ou virtual, inclusive com propósitos comerciais, desde que respeitada as regras e requisitos previstos no Capítulo IV desta Instrução Normativa".

Dentre as definições da IN, percebe-se incoerência na definição de "tecnologia proprietária" presente no inciso III do artigo 3º: "tecnologia proprietária: aquela cuja cópia, uso, redistribuição ou modificação são, em alguma medida, restringidas ou liberadas mediante contrato". A definição utilizada abrange qualquer obra autoral que seja objeto de contrato ou licença e, portanto, não tem sucesso em expor as diferenças legais entre software proprietário e software livre. Uma solução seria utilizar uma definição negativa, ou seja, definir tecnologia proprietária como o que não é tecnologia livre. Desse modo, a característica que distingue a tecnologia proprietária é a distribuição sem código fonte e/ou que não permite "ver, tentar ver ou recriar o código" (KLANG, 2005) nem distribuir trabalhos originais e derivados (LAURENT, 2004).

É vedado ao software público (art. 5º da IN 01/2011):

- Utilizar bibliotecas, componentes, ferramentas, códigos-fonte e utilitários proprietários.
- Depender somente de plataformas proprietárias;
- Depender de único fornecedor.

Não é clarificada no inciso III do artigo supracitado qual a definição de dependência no contexto do SPB nem a qual tipo de fornecedor se está referindo.

Além da disponibilização do SPB dever ser gratuita (art. 9º), requisitos técnicos e jurídicos devem ser observados. Os requisitos técnicos obrigatórios do SPB são (art.4º):

- Uma versão suficientemente estável e madura do software que possibilite a sua instalação e utilização em um ambiente de produção;
- Manual de instalação de acordo com o Anexo I da IN 01/2011 que permita ao usuário instalar o software sem o auxílio do ofertante de SPB;
- Fornecimento do código fonte do software;
- Fornecimento de todos os scripts necessários à correta instalação e utilização do software.

Softwares livres necessitam de colaboração intensiva, principalmente nas versões pré-estáveis, quando não há garantias de continuidade do desenvolvimento. A exigência

de versões estáveis e maduras para que um software se torne público coloca todo custo e risco de desenvolvimento na iniciativa privada; impede que sejam explorados, com antecedência, os benefícios do compartilhamento e revisão intensa entre pares; e reduz potencialmente a oferta de softwares no Portal SPB.

Também são elencados na IN 01/2011 os requisitos técnicos facultativos (art. 6º):

- A existência de um manual de uso que contenha, no mínimo, as informações elencadas no Anexo II da IN e que descreva todas as funções disponibilizadas pelo software;
- Fornecimento da documentação de desenvolvimento do software.

São requisitos jurídicos obrigatórios do SPB (art. 7º):

- Registro do software no Instituto Nacional de Propriedade Industrial - INPI;
- Utilização do modelo de licença CC-GPL 2.0 em português, ou algum outro modelo de licença livre que venha a ser aprovado pelo Órgão Central do SISP;
- Utilização do modelo de LPM em relação à proteção da marca do software.

SPBs podem ser ofertados tanto por parte de órgãos e entidades públicos como pela iniciativa privada (art.15). Para tal, deve-se primeiramente completar um cadastro no Portal, aceitar os termos da IN 01/2011 e preencher um formulário de envio de software. Um ambiente de avaliação é criado e o ofertante envia os artefatos do software para avaliação da adequação aos requisitos da IN 01/2011. Caso haja alguma inconsistência, os ajustes são solicitados pelo Órgão Central do SISP (art.19). Em caso de conformidade, o software e artefatos são testados e verificados. Após aprovação técnica, o registro no INPI é providenciado pelo ofertante (art.20) para que seja criada uma comunidade para o software e a solução seja disponibilizada (art.21).

De acordo com relatório preliminar da CGU, de agosto de 2014, muitas unidades da administração pública não disponibilizam suas soluções no Portal SPB por conta da obrigatoriedade de alguns requisitos técnicos e jurídicos. Para aumentar a oferta de soluções, deve-se considerar a possibilidade de que softwares sem documentação completa ou registro no INPI possam ser aceitos de forma temporária e desfrutem das vantagens do ambiente enquanto cumprem tais requisitos. É uma maneira trazer

agilidade à inclusão de soluções no Portal, além de suscitar nas unidades a vontade de disponibilizá-las, passando por um período experimental.

Um aspecto de incerteza encontra-se no artigo 14, que abre uma brecha para disponibilização de softwares não públicos e/ou que não atendam aos requisitos técnicos e jurídicos da IN 01/2011:

> “Art. 14. Também poderão ser disponibilizados no Portal SPB softwares que não adotem o modelo público de disponibilização, tendo em vista a necessidade de se compartilharem soluções de software da administração pública federal que não satisfazem a todos os requisitos técnicos e jurídicos necessários para que o software se torne um SPB.”
>
> “Parágrafo único. As soluções de software de que trata o caput deste artigo devem ser compartilhadas por meio de comunidades fechadas, acessíveis a um conjunto restrito de usuários, que devem ser adicionados a critério do administrador da comunidade e de acordo com regras de acesso previamente definidas”.

Isso cria possibilidade de que softwares sem licença sejam disponibilizados no Portal, ainda que em comunidades fechadas, criando um risco para a governança do software.

Cada SPB tem uma comunidade, que simplifica a comunicação entre governo e usuários e facilita a evolução da solução (art. 11). O ofertante de SPB é obrigado a nomear ao menos um coordenador institucional e um coordenador técnico para a comunidade (art.27). A CGU cita essa responsabilidade como uma das causas da falta de engajamento das unidades de TI no compartilhamento de software.

Considerando que, no relatório da CGU, (a) 92% das unidades pesquisadas consideram que as soluções do Portal SPB não atendem completamente às demandas internas e (b) diversas soluções não contém documentação existente ou atualizada e que (c) foi relatado pouco apoio e suporte das comunidades, é recomendado o desenvolvimento de critérios de descredenciamento de soluções, para fazer com que softwares que descumpram condições ou que sejam comprovados como de má qualidade sejam afastados do Portal SPB.

Finalmente, há um ponto conflitante entre a IN 01/2011 e a licença GPL 2.0. A IN 01/2011, no art.33, exige o cadastro prévio para que um usuário tenha acesso a um software público. Essa condição é conflitante com uma cláusula da GPL, que proíbe

que direitos garantidos por ela sejam restringidos por quaisquer outras condições. Ela está presente na seção 6 da versão 2.0:

> "Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein".

Também se encontra na versão 3.0, seção 10: "You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License".

8. Conclusões

O desenvolvimento de um software livre se diferencia do desenvolvimento de software tradicional porque frequentemente depende de trabalho coordenado sem a existência de uma estrutura organizacional e o produto final geralmente é fornecido abertamente. (STEWART *et al*., 2006). Com efeito, um desafio para muitos projetos de software livre é atrair e reter colaboradores, pois é necessário motivá-los (COLAZO; FANG, 2009; SANTOS JR., 2013). Variáveis que influenciam a motivação incluem ganho de reputação, possibilidades de ascensão na carreira, aprendizado, necessidade de uso do software a ser desenvolvido e número de usuários interessados no projeto (LERNER; TIROLE, 2002; STEWART, 2006). O relatório da CGU revelou que o Portal SPB enfrenta obstáculos que inibiram a difusão e evolução do software público. No entanto, parte dos problemas encontrados pode ser corrigida com medidas de impacto positivo na atratividade do ecossistema.

Sabendo que cada licença impacta a atração e retenção de usuários e colaboradores de modo distinto (COLAZO; FANG, 2009; SANTOS JR., 2010) e na evolução do software (HASTENREITER, 2013), a aceitação de diferentes opções de licença no Portal SPB aumentaria a oferta de softwares públicos e o número de atores interessados, sejam eles colaboradores, usuários ativos ou usuários passivos.

Preliminarmente, pode-se dizer que é oportuno que mais modelos de licenças sejam aceitos. O diagnóstico preliminar do SPB revelou a falta de critérios para a aprovação de licenças pelo Órgão Central do SISP (Secretaria de Logística e Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão). É ideal que critérios sejam adotados para a aprovação de licenças, impedindo decisões arbitrárias e

subjetivas que prejudiquem a governança do software público. Também é importante que não seja desenvolvido um rol taxativo de licenças, pois o ambiente tecnológico, de desenvolvimento e de mercado evoluem, sendo necessário que se consiga acompanhar essas mudanças sem a necessidade de revisões constantes das instruções normativas referentes ao software público.

Até o momento, a única licença aceita no âmbito do software público é a CC-GPL 2.0, versão traduzida e não oficial da GPL 2.0. A análise detalhada da GPL 2.0 expôs a fragilidade desta licença em diversos aspectos. Primeiramente, sua redação possui passagens ambíguas e trechos incoerentes com diversas legislações autorais, inclusive a brasileira (SABINO; KON, 2009). Segundo, é ausente na GPL 2.0 uma licença explícita de patente. Isso significa que caso um titular de patente distribua código sob uma licença livre sem licença explícita de patente, como GPL 2.0, "o escopo e cobertura de tal licença implícita seria difícil de estabelecer" (FONTANA *et al*, 2008, p.35), trazendo incerteza e risco de que possa haver patenteamento de um software livre e subsequente litígio contra licenciados. Terceiro, ela é tecnologicamente defasada em relação ao problema da tivoização (TSAI, 2008). Para superar essas ameaças, é necessário que existam outros modelos de licença.

É crucial também que softwares não maduros sejam incluídos, já que os estágios pré-estáveis (e.g. *alpha*, *beta*) são fases críticas para o desenvolvimento de um software, nas quais os feedbacks dos usuários e a revisão do código por pares definirão o futuro do software. Os usuários também são stakeholders relevantes. Uma solução que, nos períodos iniciais de seu projeto, tem acesso à comunidade de potenciais usuários, consegue atender às demandas destes ao mesmo tempo em que recebe sugestão de correções e mantém os colaboradores motivados com o aumento da base de usuários (STEWART *et al*., 2006). Além disso, ao barrar a entrada de softwares não maduros, alocam-se todos os custos e riscos do desenvolvimento na iniciativa privada, diminuindo, por consequência, a motivação de uma futura disponibilização do software estável no Portal. Por esses motivos, recomenda-se a alteração dos requisitos técnicos do inciso I do artigo 4º da IN 01/2011 para que sejam aceitos softwares não estáveis.

Ainda é prematuro tirar conclusões definitivas. O atual estudo pretende agregar insumos que forneçam um parecer seguro contendo sugestões de mudanças no Portal

SPB e seus respectivos impactos. Será planejada a coleta de dados com a comunidade e já estão em andamento consultorias jurídicas junto a um conceituado escritório de advocacia da iniciativa privada e à Advocacia-Geral da União (AGU). Às entidades citadas foram realizadas perguntas a respeito da base jurídica que sustenta o software livre (podem ser encontradas no Apêndice A deste documento).

REFERÊNCIAS

ALVES, A. M.; STEFANUTO, G. N.; DE CASTRO, P. F. D.; ELEUTÉRIO, S. A. V. Software Público Brasileiro: muito além do compartilhamento de software. **Revista InfoBrasil Especial**, p.19-21, jun./ago. 2009.

APACHE LICENSE, VERSION 2.0. **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/Apache-2.0>. Acesso em: ago. 2014.

ARPIT, B. E-Government and Social Media as Openness and Anti-Corruption Strategy. **Research Journal of Management Sciences**, v.1, p.48-52, Ago, 2012.

BLACK DUCK . Top 20 Open Source Licenses. Disponível em: <http://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>. Acesso em: out. 2014.

BRASIL, Lei nº 9609, de 19 de fevereiro de 1998.

BRASIL, Lei nº 9610, de 19 de fevereiro de 1998.

BRASIL. **Ministério do Planejamento, Orçamento e Gestão**. Secretaria de Logística e Tecnologia da Informação. Planejamento estratégico 2011-2015 / Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. Brasília: MP, 2012a. 123 p.: il. Disponível em: <http://www.planejamento.gov.br/secretarias/upload/Arquivos/publicacao/slti/plano_estrategico_SLTI_2011-2015.pdf>. Acesso em: out. 2014.

BRASIL. **Ministério do Planejamento, Orçamento e Gestão**. Secretaria de Logística e Tecnologia da Informação. Estratégia Geral de Tecnologia da Informação e Comunicações 2014-2015 / Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. -Brasília: MP, 2014. 55 p.: il. Disponível em: <http://www.sisp.gov.br/egtic/wiki/download/file/EGTIC>. Acesso em: out. 2014.

BRASIL. Software Público Brasileiro: Perspectiva Sistêmica. **Centro de Tecnologia da Informação Renato Archer**, 2012b.

BURTON, M. A Peace Corps for Programmers. In:_____. LATHROP, D.; RUMA, L. **Open Government**. 2010. Cap.1, p.1-39.

CINCOM. To Build or Buy? A Question of Application Development for Compliance and Quality Systems. 2008.

COLAZO, J.; FANG, Y. Impact of license choice on Open Source Software development activity. **Journal of The American Society for Information Science and Technology**, v.60, n.5, p.997-1011, jan. 2009.

DAFFARA, C.; GONZALEZ-BARAHONA, J. M. Open Source Software for Open Government Agencies. In:_____. LATHROP, D.; RUMA, L. **Open Government**. 2010. Cap.32, p.353-370.

DE LAAT, P. B. Copyright or copyleft? An analysis of property regimes for software development. **Research Policy**, v.34, n.10, p.1511-1532, dez. 2005.

EAVES, D. After the Collapse: Open Government and the Future of Civil Service. In:_____. LATHROP, D.; RUMA, L. **Open Government**. 2010. Cap.12, p.141-154.

FALCÃO, J. *et al*. Estudo sobre o software livre comissionado pelo Instituto Nacional da Tecnologia da Informação (ITI). 2009.

FANG, Y.; NEUFELD, D. Understanding Sustained Participation in Open Source Software Projects. **Journal of Management Information Systems**, v.25, n.4, p.9-50, 2009.

FERSHTMAN, C.; GANDAL, N. Open Source Software: Motivation and Restrictive Licensing. **International Economics and Economic Policy**, v.4, n.2, p.209-225, ago. 2007.

FONTANA, R.; KUHN, B. M.; MOGLEN, E.; NORWOOD, M.; RAVICHER, D. B.; SANDLER, K.; VASILE, J.; WILLIAMSON, A. A Legal Issues Primer for Open Source and Free Software Projects. 2008. Disponível em: <http://www.softwarefreedom.org/resources/2008/foss-primer.pdf>. Acesso em: out. 2014.

FUGGETTA, A. Open source software – an evaluation. **The Journal of Systems and Software**, v.66, n.1, p.77-90, 2003.

GNU GENERAL PUBLIC LICENSE, VERSION 2 (GPL-2.0). **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/GPL-2.0>. Acesso em: ago. 2014.

GNU GENERAL PUBLIC LICENSE, VERSION 2. **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/GPL-2.0> Acesso em: ago. 2014.

GNU GENERAL PUBLIC LICENSE, VERSION 3 (GPL-3.0). **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/GPL-3.0>. Acesso em: ago. 2014.

GONZALEZ-BARAHONA, J. M.; ROBLES, G. Libre Software in Europe. In: DIBONA, Chris; STONE, Mark; COOPER, Danese. **Open sources 2.0: The continuing evolution**. O'Reilly Media, Inc., 2005. p.161-188)

GURBAXANI, V.; WHANG, S. The Impact of Information Systems on Organizations and Markets. **Communications of the ACM**, v.34, n.1, p.59-73, jan. 1991.

HASTENREITER, D. M. Impactos Da Escolha Da Licença Na Dinâmica De Desenvolvimento De Software Livre. 2013.

KLANG, M. Free software and open source: The freedom debate and its consequences. **First Monday**, v.10, n.3, p., 2005. Disponível em: <http://journals.uic.edu/ojs/index.php/fm/article/view/1211/1131>. Acesso em: out. 2014.

KON, F.; LAGO, N.; MEIRELLES, P.; SABINO, V. Software Livre e Propriedade Intelectual: Aspectos Jurídicos, Licenças e Modelos de Negócio. In: **Congresso da**

Sociedade Brasileira de Computação, jul. 2011, Natal. Disponível em: <http://ccsl.ime.usp.br/files/slpi.pdf>. Acesso em: out. 2014.

LAURENT, A. M. ST. Legal Impacts of Open Source and Free Software Licensing. In:_____. **Understanding Open Source and Free Software Licensing**. O'Reilly Media, 2004. Cap. 6, p.147-163.

LERNER, J.; TIROLE, J. The Scope of Open Source Licensing. **Journal of Law, Economics and Organization**, v.21, n.1, p.20-56, 2005.

LINDBERG, V. Choosing a License. In:_____. **Intellectual Property and Open Source: A Practical Guide to Protecting Code**. O'Reilly Media, 2008. Cap. 10, p.197-214.

MORIN A.; Urban J.; Sliz P. A Quick Guide to Software Licensing for the Scientist-Programmer. **PLoS Computational Biology**, 2012. Disponível em: http://dash.harvard.edu/handle/1/10445641. Acesso em: ago. 2014

MOZILLA PUBLIC LICENSE 2.0 (MPL-2.0). **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/MPL-2.0>. Acesso em ago. 2014.

PHIPPS, S. Is GPL licensing in decline? 2012. **InfoWorld**. Disponível em:<http://www.infoworld.com/article/2617113/open-source-software/is-gpl-licensing-in-decline-.html>. Acesso em: set. 2014.

RAYMOND, E. S. The Cathedral and The Bazaar. 1999. Disponível em: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar>. Acesso em: out. 2014.

ROSEN, L. **Open Source Licensing: Software Freedom and Intellectual Property Law**. Prentice Hall, 2004

SABINO, V.; KON, F. Licenças de Software Livre: História e Características. **Relatório Técnico RT-MAC-IME-USP**, 2009. Disponível em: <http://ccsl.ime.usp.br/files/relatorio-licencas.pdf>. Acesso em: ago. 2014

SANTOS JR., C. D. Atratividade de Projetos de Software Livre: Importância Teórica e Estratégias para Administração. **Revista de Administração de Empresas**, São Paulo, v.50, n.4, p.424-438, out./dez. 2010.

SANTOS JR., C. D.; CAVALCA, M.; KON, F.; SINGER, J.; RITTER, V.; REGINA, D.; TSUJIMOTO, T. Intellectual Property Policy and Attractiveness: A longitudinal Study of Free and Open Source Software Projects. In: **CSCW**, Hangzhou, mar. 2011.

SANTOS JR., C. D.; KUK, G.; KON, F.; PEARSON, J. The Attraction of Contributors in Free and Open Source Software Projects. **Journal of Strategic Information Systems**, 2013.

SEN, R.; SUBRAMANIAM, C.; NELSON, M. L. Determinants of the Choice of Open Source Software License. **Journal of Management Information Systems**, v.25, n.3, p.207-239, 2008.

STALLMAN, R. The GNU operating system and the free software movement. 1999. Disponível em: < http://oreilly.com/catalog/opensources/book/stallman.html>. Acesso em: ago. 2014.

STEWART, K. J.; AMMETER, A. P.; MARUPING, L. M. Impacts of license choice and organizational sponsorship on success in open source software development projects. **Information Systems Research**, v.17, n.2, p.126-144, 2006.

THE GNU LESSER GENERAL PUBLIC LICENSE, VERSION 2.1 (LGPL-2.1). **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/LGPL-2.1>. Acesso em: ago. 2014.

THE GNU LESSER GENERAL PUBLIC LICENSE, VERSION 3.0 (LGPL-3.0). **Open Source Initiative**. Disponível em: <http://opensource.org/licenses/LGPL-3.0>. Acesso em ago. 2014.

THE OPEN SOURCE DEFINITION. Open Source Initiative. Disponível em: <http://opensource.org/osd>. Acesso em: ago. 2014.

TSAI, J. For Better or Worse: Introducing the GNU General Public License Version 3. **Berkeley Technology Law Journal**, v.23, p.547-581, 2008.

VARIOUS LICENSES AND COMMENTS ABOUT THEM. Free **Software Foundation**. Disponível em: <http://www.gnu.org/licenses/license-list>. Acesso em: ago. 2014.

VON HIPPEL, E. Innovation Communities. In: _____**Democratizing Innovation**. Boston, MA, The MIT Press, 2005. Cap. 7, p.93-106.

WHAT'S FREE SOFTWARE. **Free Software Foundation**. Disponível em: <www.gnu.org/philosophy/free-sw>. Acesso em: ago. 2014.

WHY UPGRADE TO GPLV3. **Free Software Foundation**. Disponível em: <http://www.gnu.org/licenses/rms-why-gplv3>. Acesso em: ago. 2014.

APÊNDICE A – Perguntas aos Especialistas

1. Quais são os requisitos para que uma licença de software seja reconhecida juridicamente no Brasil?
2. Quais licenças livres são reconhecidas no Brasil? Elas já foram citadas ou usadas em embasamentos de decisões judiciais? Quais por exemplo?
3. Existem decisões judiciais brasileiras ou, se não existirem, estrangeiras, que estabelecem quando um código-fonte é razoavelmente independente de outro ou não a ponto de se definir se “a contaminação” da GPL é necessária?
4. Dentro do contexto de licitação para contração de serviços, é possível o gestor público federal contratar mão-de-obra no mercado para aprimorar software livre? O software deve possuir alguma licença específica? Deve constar no SPB? É possível contratar pessoal física e jurídica nesses casos?
5. Duplo licenciamento: É possível, no âmbito da administração pública federal, um órgão disponibilizar um software como GPL para outros órgãos públicos, e cobrar caso o usuário seja a iniciativa privada (ou seja, o software seria livre para a administração pública e proprietário para iniciativa privada)?
6. Na literatura científica, é possível encontrar afirmações como a seguinte: “Ao exigir reciprocidade sob trabalhos que contenham todo ou parte do Programa, a licença reivindica direitos sob obras que, pela ótica da Lei de direitos autorais, pertencem ao organizador e não ao titular da obra original.” Tais afirmações dão indicativos de que existem contradições entre a GPL e a lei de direitos autorais. Quais são essas contradições?
7. Juridicamente falando, pode-se citar um critério em normativo brasileiro dizendo que para uma licença ser aceita é necessário que ela seja reconhecida por um órgão estrangeiro como a Open Source Initiative e a Free Software Foundation?