

Colaboração aberta e sua relação com a contratação de software na administração pública

Carla Rocha¹

Ricardo Augusto Poppi Martins²

Introdução

O objetivo deste texto é apresentar sucintamente o fenômeno geral da colaboração aberta e o mais específico da codificação social, que diz respeito à construção de códigos em plataformas de colaboração. A partir daí, pretende-se estabelecer uma relação sobre os efeitos desses fenômenos no setor público e na relação da sociedade com o Estado.

Uma vez delineado esse campo, vamos tratar de como esses novos paradigmas que estão surgindo afetam os mecanismos de contratação de software pelo Estado, tendo como base duas experiências atualmente em voga na administração pública federal, os Termos de Execução Descentralizada com universidades públicas federais (TEDs) e as licitações de Ateliê de Software.

Este texto não busca ser exaustivo na análise nem dos novos paradigmas, nem dos instrumentos mencionados, buscando, na verdade, apontar algumas de suas características gerais e levantar hipóteses para a pesquisa e a reflexão pela administração pública, por pesquisadores e pela sociedade em geral.

O fenômeno da colaboração aberta e da codificação social

Em respeito ao estudo da inovação, já faz alguns anos que a literatura tem apontado para o fenômeno da colaboração aberta ou inovação colaborativa (SØRENSEN & TORFING, 2011; FORTE & LAMPE, 2013; MERGEL, 2015). A inovação colaborativa se dá por meio de uma plataforma tecnologicamente mediada com baixas barreiras de entrada e saída; participa

¹ Professora de Engenharia de Software/Universidade de Brasília, Coordenadora do Laboratório Avançado de Produção, Pesquisa e Inovação em Software - Lappis.

² Mestrando no programa de pós-graduação em Ciência Política/Universidade de Brasília e Pesquisador Sênior do Laboratório Avançado de Produção, Pesquisa e Inovação em Software - Lappis.

quem quer, quando quer, como quiser. Seria, também, uma espécie de inovação produzida pela colaboração baseada em rede, diferente daquela produzida pelas hierarquias do setor público ou pela competição do mercado.

Dentro desse fenômeno mais amplo, é possível localizar um tipo específico voltado à construção e inovação em software. Chamado na literatura de “social coding” (ou *codificação social*, numa tradução livre) (MERGEL, 2015), esse tipo de colaboração se manifesta na construção de softwares de código aberto por meio de plataformas de colaboração onde redes de desenvolvedores e usuários colaboram aos milhares. Essas plataformas proporcionam ambientes colaborativos onde os desenvolvedores podem receber relatos organizados dos usuários e outros membros das comunidades de colaboração (as chamadas “issues”) podem se abrir para colaboração externa e colaborar numa miríade de projetos distintos (através dos chamados “pull requests”) e podem estudar o código de qualquer projeto através dos recursos de transparência da própria plataforma, que permite navegar pelas discussões que levaram à produção dos códigos ali presentes.

No que diz respeito ao setor público, ainda que seja incalculável o potencial de colaboração da sociedade nos softwares adotados pelo Estado através das plataformas de codificação social, a literatura demonstra que o principal uso desse fenômeno tem sido o compartilhamento de código entre uma mesma organização estatal ou entre organizações distintas (MERGEL, 2015; CRIADO, 2016). Segundo Criado,

O caso do GitHub³ demonstra o potencial para a colaboração aberta entre servidores públicos dentro de uma plataforma aberta na qual podem interagir com outros profissionais do setor para desenvolver projetos conjuntamente, tanto de software quanto de outro tipo (CRIADO, 2016, p. 264, tradução nossa).

Para que a colaboração aberta de software (“social coding”) seja viável, é fundamental que o acesso ao código seja viável por qualquer pessoa na

³ O Github é a maior plataforma de colaboração aberta em código no mundo. Mantida por uma empresa norte-americana, a plataforma permite que os desenvolvedores e organizações publiquem seus códigos e colaborem nos códigos publicados pelos outros desenvolvedores. Embora seja uma plataforma de compartilhamento e colaboração em código, ela possui funções típicas de mídias sociais, como curtidas e menções, facilitando a gestão colaborativa das propostas de melhorias e de tarefas. Link de acesso: <https://github.com/>

internet. Uma das principais definições da colaboração aberta diz respeito às baixas barreiras para participar (ou deixar de participar). Se o acesso ao código for dificultado, a colaboração não floresce. Desde algumas décadas atrás, a partir do desenvolvimento dos instrumentos de licenciamento de software, o código-fonte tem sido tratado como propriedade intelectual, um ativo comercial a se proteger. Esse cercamento moderno do conhecimento embutido nos códigos-fonte dos softwares teve uma resposta, na década de 80, pelo movimento do software livre, que entende que a liberdade de modificar e estudar o código do software que se utiliza deve ser inalienável. Para isso, esse movimento formulou uma lista de 4 liberdades que nenhum ser humano pode ser privado no contato com qualquer software, chamadas de “as 4 liberdades básicas do software livre” (executar, estudar, distribuir e modificar)⁴. Para garantir isso, esse movimento utilizou o mecanismo da propriedade intelectual para criar um tipo de licença de software que exige a distribuição do código-fonte, incluindo qualquer modificação feita no software por qualquer pessoa, mesmo que posteriormente. A licença criada pelo movimento software livre é a GPL (“General Public Licence”), que está, atualmente, na sua terceira versão (AGPLv3).

Atualmente, a disponibilidade de licenças que viabilizam a colaboração aberta em software é maior, graças a movimentos dissidentes que surgiram a partir do movimento software livre, como a turma do “open source”, ampliando a possibilidade de abertura de código. As licenças ligadas ao movimento “open source” desobrigam os futuros contribuidores a licenciar suas modificações em formato aberto, atraindo outros modelos de negócio para o ecossistema da colaboração aberta em software. Kon et al (2012) propuseram uma tipologia que organiza essa grande variedade de licenças em três tipos: (1) Licenças permissivas – que são aquelas onde “não é feita nenhuma restrição ao licenciamento de trabalhos derivados, podendo estes, inclusive, ser distribuídos sob uma licença restrita” (Kon et al, 2012); (2) Licenças recíprocas totais – que “determinam que qualquer trabalho derivado precisa ser distribuído sob os mesmos termos da licença original” (Kon et al, 2012); e, finalmente, as (3) Recíprocas parciais – que, embora exijam os mesmos termos para os trabalhos derivados, “quando o trabalho é utilizado apenas como um componente de outro projeto, esse projeto não precisa estar sob a mesma licença” (Kon et al, 2012).

⁴ (1) A liberdade de executar o programa, para qualquer propósito; (2) A liberdade de estudar o programa, e adaptá-lo para as suas necessidades; (3) A liberdade de redistribuir cópias do programa de modo que você possa ajudar ao seu próximo; (4) A liberdade de modificar (aperfeiçoar) o programa e distribuir estas modificações, de modo que toda a comunidade se beneficie.

O nosso objetivo aqui não é analisar as vantagens e desvantagens de cada abordagem, mas sim apontar que todas permitem que o fenômeno da colaboração aberta em software aconteça. Para as licenças ligadas ao movimento do software livre, a obrigação de manter abertos os códigos vem do próprio instrumento da propriedade intelectual (o texto da própria licença), enquanto para as ligadas ao movimento “open source”, a manutenção da abertura não é uma obrigação, mas algo que é incentivado pelo nível de atividade do próprio projeto. Quando um projeto de software “open source” é muito ativo, o custo em não publicar uma modificação está em se distanciar da base de código principal do projeto (árvore principal) e perder as próprias modificações no longo prazo, já que sempre será muito grande o incentivo para obter as outras atualizações feitas pelos demais desenvolvedores do projeto. Para fins de definição, sempre que nesse texto nos referirmos ao código construído nas plataformas de codificação social usaremos o termo FLOSS, que é uma junção das iniciais de “free”, “libre” e de “open source software”⁵. O termo FLOSS será utilizado aqui sempre para a referência ao software livre ou ao software de código aberto (“open source”).

Efeitos no setor público e na relação Sociedade/Estado

Para além dos efeitos na eficiência estatal pelo compartilhamento de códigos entre servidores e órgãos, a literatura de governança digital aponta para um ganho de participação e co-construção de políticas por meio das plataformas de colaboração aberta. Vaz (2017) alerta que existe um processo global de superação do modo “broadcasting” da governança eletrônica, que adquire um caráter cada vez mais relacional. Segundo o autor,

O quadro de transformações tecnológicas e a sua interação com os processos sociais permitem que se considere a emergência de uma segunda geração da governança eletrônica. Supera-se o modo broadcasting de governança eletrônica: quebra-se o monopólio do Estado sobre as decisões e iniciativas de transparência e participação nas políticas públicas. Surgem outras formas de promover a participação, a transparência e o controle social das políticas públicas. As práticas de desenvolvimento compar-

⁵ Para uma explicação (em inglês) do conceito FLOSS ver: <https://www.gnu.org/philosophy/floss-and-foss.html>

tilhado e os dados governamentais abertos permitem a coprodução e a produção descentralizadas de aplicações e serviços de base tecnológica [...]. Isso significa que podem emergir, desvinculadamente dos governos, formas de participação e intervenção nas decisões das políticas públicas baseadas na tecnologia (VAZ, 2017, p. 91).

A visão aguçada sobre o potencial da colaboração aberta na construção de softwares no setor público já se manifesta há alguns anos nas experiências brasileiras. Uma das iniciativas precursoras para o aproveitamento desse potencial foi a do Portal do Software Público Brasileiro (SPB), lançado em 2007, que, em apenas três anos de vida, já “disponibilizava à sociedade trinta e seis soluções – ou softwares públicos. Tais soluções eram utilizadas por 66 mil usuários cadastrados à época” (FREITAS & MEFFE, 2010, p. 535). O portal era utilizado por vários atores diferentes, como pessoas físicas, empresas e órgãos das diversas esferas de governo, que cooperavam numa mesma lógica de “disponibilização de bens e serviços públicos em uma rede virtual interorganizacional de produção e difusão de conhecimento tecnológico” (FREITAS, 2012, p. 111). Pelo portal, que ainda segue ativo, qualquer órgão público poderia aderir a qualquer uma das soluções disponibilizadas, se beneficiando dos seus arranjos de desenvolvimento, fossem eles realizados pelos próprios órgãos, ou abrindo procedimentos licitatórios para que as empresas envolvidas no ecossistema (o portal chamava de “comunidades”) pudessem prestar serviços relacionados ao software adotado.

A iniciativa do Portal do Software Público Brasileiro atuou em consonância com a recomendação de Sørensen & Torfing (2011) quando dizem que “o papel dos gestores públicos não é o de produzir a inovação pública, mas o de criar, institucionalizar e gerir arenas abertas e flexíveis para a interação colaborativa com outros atores relevantes ou afetados” (p. 857). Por outro lado, o SPB sempre enfrentou dificuldades para se manter como uma iniciativa vibrante. A baixa capacidade de produzir impacto na administração pública pode estar relacionada com sua excessiva institucionalização, o que acabou criando uma barreira burocrática para a adesão e participação no portal. Isso teria causado o efeito de distanciar as práticas do portal das práticas mais comuns das comunidades de FLOSS nas demais plataformas de colaboração aberta, como GitHub etc. Segundo O'Maley (2013), esse diagnóstico é confirmado por um grande número de desenvolvedores brasileiros de FLOSS “que acreditam faltar à iniciativa tanto uma comunidade de programadores vibrante e inovadora quanto a oferta de software de ponta” (p. 5).

A adoção de softwares pelo Estado é relevante porque impacta diretamente a oferta dos serviços públicos aos cidadãos. A estimativa é que existam, atualmente, aproximadamente 2 mil serviços⁶ públicos só na administração pública federal. Diante do fato de que esses serviços estão em processo de digitalização, diversos tipos de softwares estão sendo adotados para viabilizar a sua oferta e, conseqüentemente, dar suporte às políticas públicas. Segundo Germani (2016),

Serviços digitais se concretizam na forma de software que, acessíveis a partir de dispositivos conectados à rede, podem atender milhares – ou milhões – de pessoas simultaneamente. Softwares, por sua vez, são conjuntos de instruções, escritos em forma de texto, que indicam para o computador o que ele deve fazer. Logo, serviços digitais são, também, produtos de produção intelectual – e, portanto, conhecimento – assim como textos, leis, partituras etc. Este entendimento é importante, pois, ao analisar as novas práticas e metodologias de desenvolvimento de serviços digitais, analisaremos, sobretudo, sob a perspectiva das mudanças estruturais trazidas pelos meios digitais de comunicação, em especial a Internet, sobre a forma como a humanidade produz conhecimento (p. 5).

Entre os exemplos de softwares que suportam os serviços públicos digitais está o Sistema de Apoio às Leis de Incentivo à Cultura (SALIC)⁷. Esse sistema suporta a política nacional de incentivo à cultura, amparada pela popular Lei Rouanet. A maneira com que os diversos atores do setor cultural têm acesso a essa política é através desse serviço digital. Dessa forma, uma importante política pública no Brasil tem nesse serviço digital um de seus principais componentes. Muitas das decisões ligadas ao desenvolvimento dos softwares que viabilizam a oferta de um serviço desse tipo passam por decisões políticas. Decisões sobre essas tecnologias acabam, portanto, refletindo ou condicionando as decisões sobre as políticas públicas que esses serviços suportam.

⁶ Reportagem: <http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infolid=46522&csid=154>

⁷ Link de acesso ao SALIC, mantido pelo Ministério da Cultura: <http://salic.cultura.gov.br>

Arranjo de comunidade

O politólogo Steven Weber (2004) explorou o tema das comunidades de desenvolvimento colaborativo de software de código aberto. Segundo Weber (2004), o processo do código aberto é uma organização política que opera a gestão de conflitos, poder, interesses, regras, normas comportamentais, procedimentos de tomada de decisão e mecanismos de sanção, porém, de uma maneira diferente da lógica da economia política da era industrial (ibidem, p. 3). Não é um processo “caótico e desregrado no qual todos têm igual poder e influência” (ibidem, p. 3), nem uma comunidade idílica de amigos com a mesma opinião, onde reinam o consenso e os acordos fáceis. De fato, o conflito é inerente ao processo do FLOSS (ibidem, p. 3). Ainda segundo Weber, o processo do FLOSS demonstra a “viabilidade de um sistema de inovação massivamente distribuído que estende as fronteiras das noções convencionais sobre os limites da divisão de trabalho” (ibidem, p. 14). Essas características ficam claras quando o autor diz que as comunidades de código aberto “descrevem uma estrutura nascente de cultura e comunidade, [incluindo critérios] para entrar (e sair), papéis de liderança, relações de poder, questões distributivistas, formas de educação e socialização” (ibidem, p. 15).

O diagnóstico político de Weber é corroborado por estudos dentro da Computação e Engenharia de Software a respeito do desenvolvimento colaborativo. Esses estudos analisam a atuação dos desenvolvedores nas plataformas de colaboração aberta. A partir da análise em ambientes de codificação social, Dabbish & Stuart (2012) demonstram que os mecanismos de transparência presentes na plataforma do GitHub viabilizam o trabalho cooperativo, diminuindo a necessidade de encontros sincronizados. Segundo as autoras, seus achados trazem informações relevantes sobre o “desenho desse tipo de mídia social para colaboração em larga escala e implicam uma variedade de maneiras pelas quais a transparência pode apoiar inovação, compartilhamento de conhecimento e construção de comunidade” (DABBISH & STUART, 2012, p. 1286). McDonald & Goggins (2013) também perceberam o valor da transparência e das ferramentas sociais de colaboração no dia a dia dos desenvolvedores que operam nesses ambientes. Segundo alguns resultados de suas pesquisas, também com base no GitHub, os desenvolvedores associam a experiência de uso da plataforma com o desenvolvimento de práticas mais democráticas e transparentes de trabalho. O ambiente de discussão em torno das contribuições permite que haja um diálogo público atrelado ao processo decisório sobre novas incorporações de código, permitindo que quem

proponha a alteração tenha acesso e exponha todos os argumentos e todos os interessados naquela mudança possam se manifestar (MCDONALD & GOGGINS, 2013, p. 142). O uso do mecanismo das menções (@usuário), típico dos ambientes de mídias sociais, também é utilizado como instrumento de trabalho para aumentar a agilidade na resolução dos problemas e melhorar a comunicação entre os desenvolvedores e usuários (ZHANG, WANG & YIN, 2015, p. 4). O histórico público desse processo (transparência) também permite que ele seja posteriormente conhecido por qualquer desenvolvedor ou usuário da comunidade.

O Estado e as comunidades

Podemos dizer que um arranjo FLOSS é uma instituição social regida pela inovação distribuída, ágil, transparente e com altos níveis de qualidade técnica, que já tem funcionado com a participação de agentes estatais. O livre acesso aos códigos e à documentação da comunidade permite que diferentes atores possam investir em desenvolvimento de forma complementar e articulada com as outras partes, potencializando a entrega de resultados e, a partir da transparência, aumentando a confiança entre os membros do arranjo. Esse tipo de arranjo também fortalece a autonomia da sociedade na construção de tecnologias que dão suporte às políticas públicas ao mesmo tempo que preserva a soberania estatal na execução das políticas, eliminando a dependência das instituições estatais de um único agente privado. Porém, efetivar a adesão do Estado a esses arranjos não é uma operação simples, exigindo outros tipos de capacidades de gestão, além de instrumentos de contratação e financiamento de desenvolvimento de softwares que possam ser operados nesses novos paradigmas.

Quando se trata de instituições estatais, o principal apelo deve estar relacionado ao seu caráter público e responsivo à participação da cidadania. Segundo Germani (2016), os processos de desenvolvimento de software baseados em códigos abertos, para além do licenciamento do código-fonte, promovem a abertura do processo de desenvolvimento para os cidadãos, “que serão futuros usuários do serviço, para participar da sua construção, e permitindo com que outros atores, de dentro e de fora do governo, possam acompanhar, opinar e colaborar com o processo” (GERMANI, 2016). Segundo Meireles (2015), o FLOSS permite que os “softwares sejam modificados pelas próprias pessoas, com base em suas necessidades funcionais, diferenças culturais, personalidade e estilo” (MEIRELES, 2015). Além disso, ele torna “o desenvolvimento mais

ágil, como também promove ciclos de inovação aberta, ao contar com a participação de comunidades interessadas” (MEIRELES, 2015).

Além desses argumentos, vale mencionar o trabalho realizado pelo Grupo de Trabalho que investigou a possibilidade de migrações para FLOSS na Universidade Federal do ABC (UFABC) (RATCOV, et al., 2018), que produziu um relatório que elenca os seguintes argumentos pró-software livre:

1. **Economicidade:** possuem custos mais baixos, principalmente quando operados em escala, mesmo considerando os investimentos necessários para a saúde do arranjo de comunidade;
2. **Segurança:** software auditável é sempre mais seguro;
3. **Autonomia tecnológica:** permite que cidadãos brasileiros e arranjos locais tenham completo conhecimento dos códigos e tecnologias, afastando dependências externas;
4. **Independência de fornecedores:** os arranjos de comunidade são diversos, de forma que o órgão público não fica refém de um único fornecedor;
5. **Argumento democrático:** já explicitado acima, sua construção é naturalmente participativa;
6. **Estimula um mercado interno de desenvolvimento:** permite que diversos empreendimentos locais se capacitem nas tecnologias devido à baixa barreira de entrada;
7. **Promove inclusão digital:** diminuindo barreiras de custo e acesso a conhecimento;
8. **Softwares livres naturalmente suportam formatos abertos,** pois são construídos de forma a atender uma ampla gama de iniciativas e organizações.

Quanto melhor uma tecnologia é desenhada, tendo em vista os objetivos e sujeitos que farão uso dela, mais sucesso e impacto essa tecnologia irá gerar. Pequenos detalhes, muitas vezes, são responsáveis por diferenças expressivas nos resultados. Dessa forma, optar por uma metodologia participativa na construção de tecnologias que suportam os serviços públicos digitais parece ser uma decisão acertada. Além do aumento de eficiência e qualidade na construção dos códigos informáticos, as decisões de desenho tomadas em conjunto

com o público sujeito e interessado levarão à prestação de melhores serviços públicos. Atualmente, esse processo é incentivado por duas das maiores economias mundiais, Estados Unidos⁸ e Reino Unido⁹, além de diversas cidades referência no mundo, como o caso de Barcelona¹⁰. Como já falamos anteriormente, o Brasil também tem uma iniciativa consolidada nessa mesma linha, o portal do Software Público Brasileiro, tendo obtido resultados importantes na última década¹¹.

Refletindo sobre instrumentos de contratação

Para essa reflexão, vamos considerar dois instrumentos pelos quais um órgão estatal pode aderir a um arranjo FLOSS. A ideia de adesão ao arranjo FLOSS é definida aqui como a utilização, por parte de uma instituição estatal, de instrumentos para adotar e financiar o desenvolvimento de infraestruturas e aplicações digitais FLOSS, com vistas a implementar um determinado serviço e política pública.

Os exemplos foram trabalhados considerando dois instrumentos distintos, o TED com universidades públicas federais e o Ateliê de Software via licitação com empresas privadas. Serão analisados alguns aspectos transversais à gestão de Tecnologia da Informação, como Efetividade, Sustentabilidade e Inovação, além da adequação de cada instrumento para operar a adesão do Estado aos arranjos de comunidades FLOSS.

O instrumento do TED

O instrumento do termo de Execução Descentralizada é definido pelo Decreto nº 8.180, de 30 de dezembro de 2013, como “instrumento por meio do qual é ajustada a descentralização de crédito entre órgãos e/ou entidades integrantes dos Orçamentos Fiscal e da Seguridade Social da União, para execução de ações de interesse da unidade orçamentária descentralizadora e consecução do objeto previsto no programa de trabalho, respeitada fielmente a classificação funcional programática”¹². Esse instrumento é utilizado para a

⁸ <https://code.gov/#/>

⁹ <https://www.gov.uk/guidance/be-open-and-use-open-source>

¹⁰ <http://ajuntament.barcelona.cat/digital/es/transformacion-digital/tecnologia-para-un-gobierno-mejor/transformacion-con-metodologia-agile>

¹¹ <https://softwarepublico.gov.br/social/>

¹² http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/decreto/D8180.htm

realização de parceria entre órgãos vinculados ao Orçamento Geral da União (OGU), podendo parte do orçamento ser transferido para ser empenhado e executado por outros órgãos, desde que cumprindo a função original para a qual aquele orçamento foi destinado. É um instrumento amplamente utilizado para que órgãos da administração direta possam financiar pesquisa e desenvolvimento nas universidades públicas e assim auxiliar e qualificar a entrega dos objetivos pretendidos pelo órgão.

Alguns órgãos públicos têm usado o instrumento do TED com universidades públicas federais como uma alternativa ao modelo das fábricas de software, que entregam soluções de baixa qualidade a custos altos. A parceria com as universidades públicas permite aos órgãos operar a inovação em tecnologia fora da métrica por ponto de função. Essa métrica não considera o software como um produto de criatividade intelectual que busca solucionar um problema finalístico, buscando quantificar o seu desenvolvimento pelos seus próprios componentes tecnológicos. No modelo da fábrica de software, a empresa tem o incentivo a desenhar uma solução mais difícil de usar, desde que tenha mais componentes pelos quais possa fazer cobranças maiores. No final das contas, a relação entre a necessidade do órgão de contratar software de qualidade ao menor custo possível e a necessidade da empresa de faturar mais com menos trabalho empenhado fica desequilibrada para o lado do órgão, que acaba recebendo um software ruim e pagando caro por isso.

Já o modelo de TED com universidades públicas federais tem na inovação e no software de qualidade uma das premissas da parceria. Isso porque a parceria, além de envolver a entrega da solução de software, contempla, também, a realização de pesquisa e a entrega de outros produtos e atividades ligadas à produção de conhecimento, como formação de novos pesquisadores, publicação de artigos acadêmicos e participação/promoção de seminários. A parceria é planejada através de um plano de trabalho acordado entre as partes que prevê as diversas fases do projeto e um cronograma de desembolso que considere as necessidades reais do projeto em cada etapa. A partir da realização desse planejamento e firmado o instrumento, as partes podem focar a energia no que realmente traz valor para ambas: a pesquisa e desenvolvimento da solução. Nesse modelo, a forma de aferição de resultados é mais flexível por se tratar de dois órgãos federais. A Universidade está sujeita à legislação de licitações e presta contas aos órgãos de controle da mesma forma que o órgão descentralizador, com responsabilidade dividida.

Por outro lado, essa flexibilidade também pode ser uma fragilidade do instrumento. A falta de acompanhamento da parceria pelo órgão descentralizador pode gerar um desequilíbrio em relação aos interesses das partes, fazendo com que a universidade se concentre mais nos resultados acadêmicos de formação e publicação de artigos e menos na entrega da solução para o órgão. Nesse caso, a inovação até seria documentada, mas não seria aplicada no caso concreto, gerando prejuízos para os objetivos que foram estabelecidos para a ação orçamentária que originou a parceria. Alguns gestores têm dúvidas sobre como garantir o cumprimento dos objetivos no contexto de uma parceria em que o órgão não pode impor sanções à universidade. A parceria ficaria muito dependente dos aspectos macros, relacionados à capacidade que a universidade tem de cumprir os objetivos da parceria, mas faltariam instrumentos concretos para garantir qualidade e alinhamento das entregas específicas com as necessidades do órgão.

O arcabouço do processo de desenvolvimento das comunidades FLOSS está disponível para ser aplicado nesse tipo de parceria, elevando a qualidade do trabalho para outros patamares, principalmente o uso de metodologia ágil, comunicação instantânea entre servidores do órgão e pesquisadores da universidade, além das práticas simplificadas de documentação em tempo real (através de ferramenta de wiki). Há relatos de alguns aperfeiçoamentos recentes que foram realizados no acompanhamento desse tipo de instrumento de parceria, como a exigência à universidade de confecção de relatórios trimestrais com indicadores, além do acompanhamento constante dos repositórios de código e demais produtos da parceria (comunicação instantânea e wikis).

Se, do ponto de vista da inovação, são inegáveis as vantagens trazidas pelo instrumento TED com universidades públicas federais, vale fazer a mesma reflexão em relação aos aspectos de sua efetividade e sustentabilidade. Quanto mais consolidado o tipo de infraestrutura de TI, menos esse instrumento pode agregar a ela. Por exemplo, para o órgão manter infraestruturas básicas como link de internet ou impressoras, esse instrumento não é adequado, pois a conexão entre a inovação e a entrega de valor é mais fraca. Para esse tipo de infraestrutura, os contratos são o instrumento mais adequado. Por outro lado, para a entrega de serviços digitais com agilidade e qualidade, como sites e serviços, aos cidadãos, o instrumento do TED passa a fazer mais sentido. O legado deixado pelos TEDs nos órgãos tende a se integrar de forma permanente nas equipes, contribuindo para aumentar a efetividade do órgão na entrega de serviços digitais para as áreas finalísticas e, consequentemente,

para os cidadãos. Siqueira et al (2018) demonstrou como a capacidade de pesquisa e adaptação do arranjo de universidade foi fundamental para o sucesso de uma parceria de dois anos, de 2014 a 2016, entre a Universidade de Brasília e o Ministério do Planejamento. Os autores que atuaram nessa parceria

(...) acreditavam que a entrega constante de software era melhor para o sucesso do projeto. Em oposição a isso, o Ministério do Planejamento estava habituado à ideia de uma única entrega ao final do projeto, e nem sua estrutura burocrática, nem sua experiência técnica estavam alinhadas a esse estilo de trabalho. Isso estava prejudicando os benefícios da ferramenta (desenvolvida) e impedindo o time da UnB de mostrar os frutos do projeto para aqueles responsáveis por avaliarem o trabalho (SIQUEIRA, et al., 2018).

Dessa forma, a capacidade do time do projeto de pesquisar e implementar a técnica da engenharia de software conhecida como entrega contínua¹³ foi decisiva na transformação cultural necessária para o órgão absorver o trabalho que estava sendo realizado no contexto do projeto. Segundo os autores,

(...) a entrega contínua ajudou a fortalecer a confiança entre desenvolvedores e a equipe do ministério. Antes de utilizar a entrega contínua, eles só conseguiam validar as novas funcionalidades desenvolvidas apenas no final do ciclo de release, normalmente a cada quatro meses. Com a implementação da entrega contínua, versões intermediárias ficavam disponíveis permitindo que eles realizassem pequenas validações ao longo do tempo. O monitoramento constante do trabalho de desenvolvimento trouxe maior segurança aos líderes do ministério e melhorou as interações com nossa equipe (SIQUEIRA, et al., 2018).

Em suma, a construção e utilização da entrega contínua no TED descrito pelo artigo foi responsável por estabelecer três novas dinâmicas, fundamentais para o sucesso da parceria e para a garantia da entrega de valor para o órgão: (1) Demonstrar resultados concretos ao invés de apenas reportá-los; (2) Tornar o gerenciamento do projeto transparente e colaborativo para a equipe do ministério; e (3) Ganhar a confiança da equipe do governo (SIQUEIRA et al., 2018).

¹³ A entrega contínua é uma técnica da engenharia de software que permite que atualizações do software sejam rapidamente disponibilizadas para os usuários.

Do ponto de vista da sustentabilidade, as vantagens desse instrumento dependem do cultivo de um ambiente propício no próprio órgão. Via de regra, os servidores dos órgãos de TI se envolvem pouco com as atividades mais centrais das áreas de tecnologia da informação como desenvolvimento e gestão de projetos, que ficam mais a cargo da gestão de contratos. Essa característica da gestão das equipes prejudica o processo de apropriação e sustentabilidade, ficando todo o conhecimento apenas nos fornecedores passageiros. Por outro lado, um ambiente onde os servidores se envolvem nas discussões e decisões sobre os sistemas e soluções de software do órgão é o mais propício para aproveitar os aportes que as parcerias com as universidades públicas podem trazer. O uso do arcabouço de trabalho das comunidades FLOSS já mencionadas aqui, aliado ao comprometimento verdadeiro dos servidores com os códigos e comunidades das soluções adotadas e desenvolvidas pelo órgão, gera um ambiente seguro e duradouro, cuja sustentabilidade das soluções evita a interrupção na oferta de serviços e políticas digitais pelos órgãos estatais.

O instrumento do Ateliê de Software

O instrumento da licitação para ateliê de software usa o mesmo instrumento legal de contratação das atuais fábricas de software (que é a lei de licitações) mas com mudanças substanciais nas exigências e nas métricas utilizadas. A base filosófica do instrumento do ateliê é o manifesto do “kraftmanship” ou manifesto para a artesanaria de software¹⁴, que parte da premissa de que a construção de software está mais próxima de um processo de criação artística do que do processo fabril, como é tratado atualmente pelas fábricas. Isso significa que o trabalho de criação de software deve estar orientado a criar valor em suas soluções e valorizar o trabalho das pessoas envolvidas, recompensando esforço e qualidade. Assim como o TED, esse instrumento se viabiliza a partir de uma crítica frontal à metodologia de trabalho e gestão e à métrica mais utilizada hoje nos processos governamentais de contratação de software, a contagem por pontos de função¹⁵. A métrica de pontos de função gera incentivos para que as empresas construam softwares piores e mais complicados para os usuários, mas que tenham remunerações mais atraentes para

¹⁴ <http://manifesto.softwarecraftsmanship.org/#/pt-br>

¹⁵ Análise de Pontos de Função (APF) é uma técnica para a medição de projetos de desenvolvimento de software, visando projetar uma medida de tamanho com base nas funções realizadas pelo software. É a métrica mais utilizada nas licitações para desenvolvimento de software do governo federal, e muito pouco usada fora desse contexto.

os seus negócios. A busca por soluções inovadoras e o esforço na construção de entregas com mais qualidade não são captados pela métrica. Esse incentivo implícito na métrica gera um desequilíbrio entre a qualidade do produto e o uso eficiente do dinheiro público, em favor do lucro das empresas.

Outra mudança que a lógica do ateliê de software promove em relação às fábricas de software é a mudança da metodologia. No conceito fabril, os processos são mais importantes que as pessoas e toda falha é analisada como uma falta de aderência a eles. Na lógica do ateliê as pessoas estão no centro do método, cuja preocupação começa desde as exigências da pirâmide invertida na montagem da equipe (a quantidade de juniores não pode ser maior que a de desenvolvedores plenos que não pode ser maior que a de desenvolvedores seniores) e a obrigatoriedade de manter a equipe presente no órgão. Com uma equipe formada majoritariamente de profissionais experientes e *“in-loco”*, o gestor passa a ter incidência sobre o clima organizacional que será criado e a metodologia de trabalho que será utilizada. A métrica utilizada pelo ateliê é a UST (Unidade de Serviço Técnico) que estima o nível de esforço necessário para produzir os principais componentes e etapas de um processo de desenvolvimento. Essa estimativa é feita a partir da criação de um repertório acordado entre o órgão e a empresa, que passa por revisões periódicas de comum acordo entre as partes. O fato de o repertório poder passar por revisões periódicas permite que ele seja capaz de reequilibrar constantemente a relação, a partir dos requisitos de qualidade exigidos pelo órgão. Parece complexo, mas não é: enquanto o manual de contagem de pontos de função tem aproximadamente 80 páginas, o repertório utilizado pelo Ministério das Relações Exteriores na contagem dos pontos de função tem, no máximo, 5 páginas¹⁶. Essa simplificação também permite que a gestão seja feita com mais qualidade. Um ponto importante para o funcionamento correto da métrica é que nada deve ser pago fora do previsto. Toda entrega é quantificada em UST antes de o trabalho começar e acordada entre as partes. Quando concluído, o valor cobrado deverá ser aquele que foi acordado anteriormente. Qualquer adaptação no repertório só poderá acontecer após a entrega do produto.

Do ponto de vista da inovação, como o repertório de contagem é gerido pelo próprio órgão, ele tem liberdade de remunerar trabalhos que contribuam para a inovação, como estudos e pesquisas. Por outro lado, uma gestão que usa como base o manifesto da artesanania de software percebe a inovação

¹⁶ Dado obtido em entrevista realizada no ano de 2018, com técnicos do setor.

como um vetor para melhorar a qualidade do trabalho das pessoas, resolvendo problemas da organização. Não se trata, portanto, de inovar por inovar. Em relação à sustentabilidade, é evidente que o instrumento do ateliê, associado ao uso de metodologia ágil, é fundamental para que os projetos sejam entregues com qualidade e atendam às necessidades dos usuários, contribuindo para a continuidade das soluções. Por fim, em relação à efetividade, o instrumento do ateliê contribui com entregas mais rápidas e melhores, aprimorando a capacidade que a TI tem de atender às necessidades dos usuários e das áreas da organização que dependem dela para desempenhar o seu trabalho.

Adequação dos instrumentos em relação à arranjos de comunidades FLOSS

O diagnóstico dos dois instrumentos (TED e Ateliê de Software) revela que, apesar de suas especificidades, eles estão mais capacitados para lidar com os desafios de Efetividade, Sustentabilidade e Inovação inerentes à TI das organizações públicas do que o modelo atual baseado em fábricas de software regidas pela métrica dos pontos de função. Por outro lado, ambos os arranjos se diferem em relação a seus pontos fortes e suas fraquezas, tornando necessária uma avaliação minuciosa do cenário de TI do órgão antes da decisão pela adoção de um ou outro instrumento. Abaixo, apresentamos uma sistematização básica desses achados em formato de tabela (Tabela 1).

Tabela 1 . Sistematização do diagnóstico realizado nos dois instrumentos

Eixo/Aspecto	Termo de Descentralização com Universidades Públicas Federais	Licitação de Ateliê de Software
Inovação	É premissa do acordo. Com uma boa prospecção de universidades com histórico de pesquisa nas áreas de desafio pretendidas pelo órgão, é uma forma eficiente de utilizar a pesquisa aplicada nos desafios e problemas do órgão. Requer acompanhamento próximo da gestão para que a inovação não seja aplicada apenas nos produtos acadêmicos da parceria.	Depende da gestão de TI do órgão. A flexibilidade dos repertórios permite à gestão remunerar inovação. Isso depende da gestão, pois o incentivo dificilmente virá da empresa. Uma gestão orientada para que as soluções de TI entreguem valor concreto para as áreas finalísticas poderá utilizar bem esse instrumento.

Efetividade	Aumenta a efetividade na oferta de serviços complexos. O legado deixado pelas universidades instrumentaliza o órgão a fazer entregas melhores e mais efetivas. Para isso, é necessário que a gestão tenha sucesso em envolver a equipe permanente (sejam servidores, sejam terceirizados) nas dinâmicas de trabalho ao longo do tempo de execução do TED.	A flexibilidade do instrumento associado a equipes mais experientes trabalhando em metodologia ágil tende a aumentar a efetividade de todas as entregas.
Sustentabilidade	O instrumento ajuda se o ambiente de trabalho compartilhado for criado. O modo de trabalho das comunidades FLOSS ajuda no processo de criação de capacidades na equipe de servidores e terceirizados.	O instrumento ajuda, mas o peso está na gestão. Entregas melhores fazem com que os projetos durem mais. O clima organizacional faz com que o conhecimento se dissemine na equipe. Por outro lado, é papel da gestão proteger a área de TI de projetos inúteis.
Pontos fortes	Não depende, para funcionar, do conhecimento prévio da gestão em relação à metodologia de trabalho.	Tem mecanismos de sanção que podem ser utilizados para melhorar a qualidade dos produtos entregues.
Fragilidades	Sem acompanhamento próximo da gestão, o foco da universidade tende a ser em produtos acadêmicos ou experimentos sem conexão com os problemas da organização.	Depende mais do conhecimento prévio da gestão em relação à metodologia de trabalho e à cultura organizacional.

Para o recorte específico que nos propusemos a desenvolver neste texto, relacionado à capacidade do Estado em aderir a arranjos de colaboração aberta e codificação social no formato de comunidades FLOSS, percebemos que o instrumento do Ateliê de Software possui inúmeras lacunas e fragilidades nesse sentido. Isso se deve ao fato de que esse instrumento depende de uma cultura organizacional pró-software livre nas áreas de TI dos órgãos para que a adesão ao FLOSS aconteça. A licitação de Ateliê de Software não possui nenhum mecanismo incentivador ou facilitador para isso. Regido por um contrato privado, um arranjo desse tipo não garante as vantagens difusas oriundas da adesão à FLOSS, como autonomia tecnológica e desenvolvimento de mercado local, entre outras.

Por outro lado, o instrumento do TED com universidades públicas se mostrou mais adequado. Isso acontece pelo fato de o TED ser operado como um instrumento baseado em uma parceria de pesquisa aplicada e não nos moldes de um contrato cliente/fornecedor. Como a universidade não pode – nem deve – operar no mesmo ritmo do mercado, ela precisa aliar a entrega de valor para o parceiro com a construção de um ambiente saudável de pesquisa e formação. O objetivo institucional da universidade de atuar na formação de pesquisadores e profissionais requer a construção de pontes amplas de colaboração, sob o risco de oferecer um ambiente medíocre e pouco desafiador para os alunos e bolsistas. Essas especificidades fazem com que os laboratórios de universidades, de certa forma, dependam da articulação com o conhecimento disponível nas comunidades para efetuar a entrega de valor ao órgão parceiro. Seria hipoteticamente possível a existência de TEDs para construir soluções fechadas e autorais, porém, com o alto custo da universidade não cumprir o seu papel institucional, afetando a qualidade e a agilidade das entregas em médio e longo prazos. Além disso, soluções fechadas possuem níveis muito mais baixos de transparência se comparadas com a transparência inerente aos arranjos de colaboração aberta, prejudicando os trabalhos de monitoramento e controle da parceria realizados tanto pelos órgãos parceiros quanto pelos órgãos de controle. Explicando melhor, a articulação com as comunidades potencializa a entrega de inovação e valor da solução ao parceiro, ao passo que a transparência das práticas de colaboração aberta mitiga a fragilidade de controle do instrumento que foi detectada no nosso diagnóstico. Em suma, o instrumento do TED acaba sendo afetado por uma força centrípeta na direção do arranjo de comunidade, tanto pelas necessidades intrínsecas da pesquisa aplicada (inovação com entrega de valor e requisitos de transparência) quanto pelo caráter aberto da cultura da boa ciência como um empreendimento coletivo e colaborativo.

Considerações Finais

Passamos por uma visão geral do fenômeno da colaboração aberta, a partir de sua vertente mais específica da codificação social. A codificação social tem sido pesquisada como um fenômeno em crescimento e cada vez mais adotado pelas instituições estatais. As potencialidades da codificação social são viabilizadas pelo modelo de licenciamento dos softwares, geridos como um bem comum. Do software livre ao “open source”, os diversos modelos permitem a realização da codificação social, ainda que com visões de mundo e incentivos diferentes.

De forma geral, os arranjos de comunidade FLOSS baseados nas plataformas de codificação social atuam com altos níveis de transparência, utilizando-se de instrumentos específicos dessas plataformas para fomentar a comunicação e construir práticas de colaboração. Do ponto de vista do Estado, a adoção de softwares produzidos dentro desses paradigmas pode aumentar a permeabilidade e a transparência na construção dos serviços digitais disponibilizados para a população, aprimorando a participação social, além de outros benefícios mais difusos para a sociedade.

Por fim, analisamos dois instrumentos de contratação e investimento público em softwares: o TED com universidades públicas e a licitação de Ateliê de Software com empresas privadas. Ambos os instrumentos têm a vantagem de superar a mediocridade da métrica de pontos por função, aumentando a entrega de valor para os órgãos e a efetividade no gasto do recurso público. Por outro lado, o TED é mais recomendado para contextos de inovação, além de ser mais adequado para a adesão do Estado a arranjos de comunidades FLOSS. O instrumento do TED com as universidades públicas, se bem manejado, traz enormes benefícios relacionados à inovação e à sustentabilidade, além dos demais benefícios difusos trazidos pela adesão a esse tipo de arranjo.

Referências Bibliográficas

CRIADO, J. Ignacio. Las administraciones publicas en la era del gobierno abierto. Gobernanza inteligente para un cambio de paradigma en la gestion publica. **Revista de Estudios Políticos**, n. 173, 2016. Disponível em: <http://www.cepc.gob.es/Publicaciones/Revistas/revistaselectronicas?IDR=3&ID-N=1361&IDA=37798> Acesso em: 25 out. 2018.

DABBISH, Laura; STUART, Colleen; TSAY, Jason; et al. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In: **Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 2012, p. 1277–1286. (CSCW'12). Disponível em: <<http://doi.acm.org/10.1145/2145204.2145396>>. Acesso em: 24 out. 2018.

FORTE, Andrea; LAMPE, Cliff, Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature, **American Behavioral Scientist**, v. 57, n. 5, p. 535–547, 2013.

FREITAS, Christiana Soares de; MEFFE, Corinto. Redes de produção de conhecimento tecnológico: um projeto governamental brasileiro. **Estudos de Sociologia**, v. 15, n. 29, 2010. Disponível em: <<http://piwik.seer.fclar.unesp.br/estudos/article/view/2978>>. Acesso em: 1 nov. 2018.

FREITAS, Christiana Soares de. O Software Público Brasileiro: novos modelos de cooperação econômica entre Estado e Sociedade Civil. **Informação & Sociedade: Estudos**, v. 22, n. 2, 2012. Disponível em: <<http://periodicos.ufpb.br/ojs2/index.php/ies/article/view/12231>>. Acesso em: 1 nov. 2018.

GERMANI, Leonardo Barbosa. **Desafios para o desenvolvimento de serviços digitais pelo governo federal brasileiro**. 2016. Disponível em: tede2.pucsp.br/tede/handle/handle/18772. Acesso em: 1 nov. 2018.

KON, Fabio et al. **Software Livre e Propriedade Intelectual**: Aspectos Jurídicos, Licenças e Modelos de Negócio. <http://ccsl.ime.usp.br/files/slpi.pdf>. Acesso em 05/nov/2018, v. 2, p. 12, 2012.

MCDONALD, Nora; GOGGINS, Sean. Performance and Participation in Open Source Software on GitHub. In: **CHI '13 Extended Abstracts on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2013, p. 139–144. (CHI EA '13). Disponível em: <<http://doi.acm.org/10.1145/2468356.2468382>>. Acesso em: 25 out. 2018.

MEIRELES, Adriana Veloso. **Democracia 3.0**: interação entre governo e cidadãos mediada por tecnologias digitais. 2015. Disponível em: <<http://repositorio.unb.br/handle/10482/19044>>. Acesso em: 26 nov. 2017.

MERGEL, Ines. **Open collaboration in the public sector**: The case of social coding on GitHub. *Government Information Quarterly*, v. 32, n. 4, p. 464–472, 2015.

O'MALEY, Daniel. Software Público Brasileiro (SPB): The State in the Commons. In: Workshop Sobre Software Livre. **Anais do WSL2013**. Porto Alegre-RS: SBC. [s.l.: s.n.], 2013, p. 1–10.

RATCOV, David; PELLEGRINI, Jerônimo Cordoni; VIEIRA, Miguel Said; da SILVA, Silas Justiniano. **Relatório - GT Software Livre**. Santo André, 2018. Disponível em: <http://nti.ufabc.edu.br/cetic-2/relatorios-de-grupos-de-trabalho>

SIQUEIRA, Rodrigo. et al, Continuous Delivery: Building Trust in a Large-Scale, Complex Government Organization, **IEEE Software**, v. 35, n. 2, p. 38–43, 2018. Disponível: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8255783>

SØRENSEN, Eva; TORFING, Jacob. Enhancing Collaborative Innovation in the Public Sector. **Administration & Society**, v. 43, n. 8, p. 842–868, 2011.

VAZ, José Carlos. Transformações tecnológicas e perspectivas para a gestão democrática das políticas culturais. **Cadernos Gestão Pública e Cidadania**, v. 22, n. 71, 2017.

WEBER, Steven. **The Success of Open Source**. Harvard University Press, 2004.

ZHANG, Yang; WANG, Huaimin; YIN, Gang; et al. Exploring the Use of @-mention to Assist Software Development in GitHub. In: **Proceedings of the 7th Asia-Pacific Symposium on Internetware**. New York, NY, USA: ACM, 2015, p. 83–92. (Internetware '15). Disponível em: <<http://doi.acm.org/10.1145/2875913.2875914>>. Acesso em: 25 out. 2018.