

8

INTRODUZINDO A CULTURA DE INOVAÇÃO TECNOLÓGICA NO PODER JUDICIÁRIO:

A EXPERIÊNCIA DO IJUSPLAB

PARTE 2 – A INCUBADORA DE SOLUÇÕES TECNOLÓGICAS

INTRODUCING THE CULTURE OF TECHNOLOGICAL INNOVATION IN THE JUDICIARY:

THE IJUSPLAB EXPERIENCE

PART 2 – THE TECHNOLOGICAL SOLUTIONS INCUBATOR

*Caio Moysés de Lima**

RESUMO

Este capítulo aborda alguns dos desafios inerentes à disseminação de uma cultura de inovação tecnológica no Poder Judiciário e mostra como o iJusLab tem auxiliado a administração da Justiça Federal de São Paulo a enfrentar esses desafios por meio da Incubadora de Soluções Tecnológicas. O tema é abordado a partir do fenômeno denominado *Shadow IT* e de como ele pode ser utilizado em favor da instituição, mediante a mudança da cultura organizacional e a criação de meios adequados de reunir talentos e estimular a prototipagem de soluções tecnológicas sem comprometer a segurança dos serviços de TI. O capítulo também aborda alguns aspectos importantes do funcionamento da incubadora, especialmente os que dizem respeito a sua estrutura, métodos de trabalho e relacionamento com o departamento de TI.

* Juiz Federal, Vice-Diretor do Foro da Seção Judiciária de São Paulo, mestre em Ética e Filosofia Política, bacharel em Direito e Filosofia, estudante de Ciência da Computação. Email: <cmlima@trf3.jus.br>.

Palavras-chave: Projeto; Inovação; Tecnologia; Justiça; Poder Judiciário.

ABSTRACT

This chapter addresses some of the challenges involved in the spreading of a culture of technological innovation in the Judiciary and shows how iJuspLab has helped the administrative body of the Federal Justice of São Paulo to face such challenges through its Technological Solutions Incubator. The subject is approached from the perspective of the Shadow IT phenomenon and of how it can be used in favor of the institution, by changing the organizational culture and creating the proper means to gather talent and stimulate the prototyping of technological solutions without compromising IT services security. The chapter also addresses some important aspects of the incubator's operation, especially those concerning its structure, work methods and relationship with the IT department.

Keyword: Project; Innovation; Technology; Justice; Judiciary.r.

INTRODUÇÃO

Os departamentos de TI tendem a ser refratários à inovação. Esse não é um juízo depreciativo, mas a constatação de um fenômeno real, objetivo, que se explica por duas características intrínsecas a esses departamentos: (i) a sua especialização gera distanciamento das áreas de negócios e dificulta a compreensão do dia a dia dessas áreas; e (ii) a missão que lhes é conferida, de zelar pela integridade, disponibilidade e segurança dos recursos de TI, torna-os naturalmente avessos à mudança e à experimentação¹.

¹ Segundo TAURION (2015), o departamento de TI, especialmente em seu viés operacional, é “uma máquina muito eficiente em garantir consistência, aderência a regras de compliance e disponibilidade”, mas “arredia às experimentações e disrupções. Seu pensamento fundamental é que os sistemas têm que dar certo e funcionar exatamente como planejado. Falhar não está em sua cartilha, e, portanto, conceitos como ‘fail fast’ lhe são estranhos”. Essa dificuldade também se explica por um conflito interno dos próprios departamento de TI, chamado por KIM et al. de “conflito crônico básico”: “Em quase toda organização de TI há um conflito inerente entre Desenvolvimento e Operações de TI que cria uma espiral

A dificuldade dos departamentos de TI de trazer inovação para suas organizações é um dos principais fatores que explicam o fenômeno da *shadow IT*, definido como “todo hardware, software ou qualquer outra solução utilizada por empregados de uma organização, no interior do ecossistema organizacional, sem aprovação *formal do departamento de TI*”². A depender do rigor da política de segurança de cada organização, podem ser casos de *Shadow IT* desde os sistemas “caseiros”, instalados e utilizados pelos próprios usuários, até as mídias aparentemente inofensivas utilizadas para armazenamento de arquivos pessoais. Alguns recursos são em geral permitidos, como o uso de *pen drives*, por exemplo, e outros são quase sempre proibidos, como o uso de sistemas “caseiros”. Acessos a serviços *online*, como Google Drive ou Dropbox, costumam ficar numa zona cinzenta; são recursos praticamente inofensivos para a segurança interna, porque executados no ambiente isolado do browser (*sandbox*), mas podem ser proibidos se prejudicarem o funcionamento de serviços críticos em razão do consumo da banda de conexão.

Diante dos riscos gerados por algumas iniciativas de *shadow IT*, é natural que elas sejam tratadas com hostilidade pelos departamentos de TI. O problema principal está na falta de compromisso dos usuários com as políticas internas da

descendente, resultando em um tempo cada vez mais longo para comercializar novos produtos e recursos, qualidade reduzida, interrupções maiores e, o pior de tudo, uma dívida técnica cada vez maior. O termo ‘dívida técnica’ foi cunhado pela primeira vez por Ward Cunningham. Análoga à dívida financeira, a dívida técnica descreve como as decisões que tomamos levam a problemas cada vez mais difíceis de corrigir com o passar do tempo, reduzindo continuamente as opções disponíveis no futuro – mesmo quando enfrentados prudentemente, ainda incorremos em juros. Um fator que contribui para isso são os objetivos frequentemente concorrentes de Desenvolvimento e Operações de TI. As organizações de TI são responsáveis por muitas coisas. Entre elas estão dois objetivos a seguir, que devem ser buscados simultaneamente: – Responder ao cenário competitivo rapidamente mutante; – Fornecer serviço estável, confiável e seguro para o cliente. Frequentemente o Desenvolvimento assumirá a responsabilidade por responder às mudanças no mercado, implementando recursos e alterações na produção o mais rapidamente possível. Operações de TI assumirá a responsabilidade por fornecer aos clientes um serviço de TI estável, confiável e seguro, dificultando ou mesmo impossibilitando a introdução de mudanças que possam comprometer a produção. Configurados dessa maneira, Desenvolvimento e Operações de TI têm objetivos e incentivos diametralmente opostos”. (p. xxiii)

² “Shadow IT is a currently misunderstood and relatively unexplored phenomena. It represents all hardware, software, or any other solutions used by employees inside of the organisational ecosystem which have not received any formal IT department approval”. Behrens, 2009; Györy, 2012. SILIC & BACK, 2014.

organização e na dificuldade de identificar os riscos envolvidos nessas iniciativas, que elas, por definição, escapam aos meios normais de controle³.

É importante notar, no entanto, que tais iniciativas raramente brotam de má-fé do usuário. Em geral, seu propósito é bastante sintonizado com os objetivos da organização, pois o que se pretende é apenas melhorar a produtividade e a qualidade do trabalho realizado pelas áreas de negócio⁴. O que as torna tão comuns hoje em dia, além da mencionada dificuldade dos departamentos de TI de inovar, é o seguinte: (i) a tecnologia hoje faz parte do cotidiano das pessoas, dentro e fora do trabalho, o que torna os usuários muito mais exigentes no que diz respeito aos recursos e à qualidade dos serviços que lhes são oferecidos; e (ii) pela facilidade que se tem hoje de aprender linguagens de programação e desenvolvimento de sistemas, muitos usuários conseguem resolver por conta própria problemas que o departamento de TI, por questões de prioridade ou de estratégia, não esteja disposto ou demore muito a solucionar.

Assim, independentemente dos riscos gerados pelas iniciativas de *shadow IT*, não se pode negar que elas trazem também benefícios para a organização. Em especial, ajudam a revelar deficiências nos serviços de TI por indicarem em que pontos ou em que áreas esses serviços não atendem plenamente os anseios dos usuários, e tendem a conferir maior qualidade e rapidez aos serviços prestados pelas áreas de negócios, visto serem desenvolvidas e implementadas por pessoas que conhecem a fundo as necessidades e dificuldades dessas áreas⁵.

³ “The main threat posed by unsanctioned software and applications hides in its unaccountability – you can’t effectively manage something that you don’t even know exists. As a result, both security and performance of the entire network are put at risk”. YATSENKO, 2018.

⁴ “Similar to past studies (e.g. Györy et al., 2012) and based on the ISsec threats taxonomy (Warkentin, 1995), we classify Shadow IT as an insider threat which is caused by the human factor of an organisation. We consider this human factor to be an insider (i.e. employee) who installs non approved software without having any malicious intentions. From that perspective, Shadow IT can be situated on the fringes of organisations, where it fills the existing gap between users and the solutions provided by an IT department (Behrens, 2009). (...) When asked about the details of the software that was identified as illegal and unapproved, the majority of the organisations listed 1) productivity software (e.g. Google apps), 2) communication software (e.g. Skype), 3) utility tools (e.g. CCleaner or 7-Zip), 4) internet browsers (e.g. Firefox), and 5) PDF tools (e.g. PDFCreator). All respondents agreed that employees, generally, do not have any malicious intentions when they install the non-approved software. Instead, it was argued that employees believe that they are not doing anything illegal especially when installing open source software”. SILIC & BACK, 2014.

⁵ “What are the benefits of shadow IT? First and foremost, the mere fact that unapproved software is running on a company’s systems shows that approved solutions don’t meet the

Por isso, diversas organizações começam a ver com bons olhos as iniciativas de *shadow IT*, reconhecendo-as como fontes potenciais de inovação. Isso não significa que tais organizações estejam simplesmente abrindo mão de suas políticas internas de segurança e de controle; elas apenas estão buscando uma situação de equilíbrio, em que as referidas políticas possam ser aplicadas sem sufocar a criatividade dos usuários. Há, portanto, uma clara tendência de aceitar que os usuários também participem e até tomem a frente nos esforços de inovação tecnológica nas organizações, o que se confirma pelo fato de que pouco a pouco começa a ser reconhecida e até valorizada a figura do “desenvolvedor cidadão” (*citizen developer*) ou usuário-desenvolvedor⁶.

Os departamentos de TI não devem, contudo, ser excluídos desse processo. Ao contrário, sua atuação permanece decisiva para o sucesso das iniciativas de inovação tecnológica, mesmo aquelas lideradas pelas áreas de negócios. A única diferença em relação ao modelo usual é que os departamentos de TI precisam agora abrir mão do monopólio que antes detinham sobre o desenvolvimento de soluções tecnológicas e passar a agir como “embaixador[es] tecnológico[s], apoiando seus usuários nas pequenas rupturas, como um parceiro de implementação, que cobre justamente os gaps técnicos das demais equipes, permitindo ainda que os demais usuários sejam ‘desenvolvedores’ de negócio de fato, participando ativamente da construção e implementação dessas rupturas desejadas ao negócio”⁷. É nesse con-

requirements of employees: they’re either inefficient or uncomfortable or both. Secondly, there’s always a chance of shadow IT turning out to be more productive and cost-effective than already deployed solutions. The main task here is to recognize the solutions that can be more beneficial to the company and find a way to implement them effectively into the current infrastructure”. Ibidem.

⁶ “A citizen developer is a user who creates new business applications for consumption by others using development and runtime environments sanctioned by corporate IT. In the past, end-user application development has typically been limited to single-user or workgroup solutions built with tools like Microsoft Excel and Access. However, today, end users can build departmental, enterprise and even public applications using shared services, fourth-generation language (4GL)-style development platforms and cloud computing services”. GARTNER, verbatim consultado em 2019.

⁷ SOMMERFELD, 2015. Nesse mesmo sentido, cf. SHIF, 2018: “Shadow IT is inevitable, and as an organization, you can choose to understand your users and find ways for them to use the cloud services they need to fulfill their job duties, or, you can attempt to crush Shadow IT like a bug and eliminate it completely. No matter what route you choose to go with, one thing is for sure: Shadow IT exists at extremely high levels and there must be some type of plan put into place before a massive security breach of sensitive data occurs”. Essa nova forma de pensar

texto que grandes empresas de consultoria têm chamado atenção para a necessidade de institucionalizar as iniciativas de *shadow IT*⁸.

A história da Incubadora de Soluções Tecnológicas do iJusPLab pode ser compreendida como um caso de sucesso nesse movimento de conciliação entre iniciativas inovadoras no desenvolvimento de *software* promovidas pelas áreas de negócios e a necessidade de manter os controles de integridade, segurança e disponibilidade dos serviços pelo departamento de TI. Neste capítulo contamos de forma sucinta essa história e descrevemos, em seguida, a estrutura da incubadora, seus métodos de trabalho e como se dá seu relacionamento com o departamento de TI.

A ORIGEM DA INCUBADORA

A origem da Incubadora remonta a um convênio celebrado com a Fundação Getúlio Vargas (FGV) em 1998 e renovado em 2001⁹. Por meio dele, previu-se a

e agir está sintetizada de forma muito clara nas seguintes declarações dadas em 2016 pelos dirigentes das áreas de TI de duas grandes empresas com negócios fortemente apoiados na inovação tecnológica (*apud* TWENTYMAN, 2016): “*The idea that innovation is about the IT team giving people tools is a well-worn myth these days. Instead, it should be about creating a landscape of provision that allows users to get creative, because that’s what drives innovation down the channels that are more valuable to the organization*”. (Paul Clarke, diretor de tecnologia da Ocado) “I’m increasingly suggesting to the business that they ‘build it themselves’ but to do so within an agreed set of terms – the tests they need to do, the minimum documentation they need to put together. And in exchange for that, we provide the back-ups, the traceability, the audibility and the workflows around it. ‘We’ll help you,’ we tell them, ‘but the business is your business.’” Antoine de Kerviler, CIO da Eurostar.

⁸ A Fujitsu, por exemplo, em um *White Paper* sobre o assunto, publicado em 2016, disse o seguinte: “Here at Fujitsu, we not only understand the importance of managing shadow IT effectively but have the expertise and ability to help our customers spot and identify it in all its many possible manifestations. Our aim is to enable enterprises to understand the big picture of shadow IT in their particular organisations, help them assess their preferred levels of risk and innovation, and then guide them in the process of standardising and managing shadow IT use for optimal business results. The ultimate goal is to help companies prevent (as much as possible) ‘bad’ uses of shadow IT while allowing them to reap the benefits of ‘good’ shadow IT. We view this as the difference between ‘rogue IT’ and ‘digital enablement’. Fujitsu recognizes this reality: our IT experience and expertise has demonstrated to us that a ban on shadow IT is essentially a ban on innovation. Instead, we believe the best approach for any organization concerned about shadow IT is to place it in the correct framework so employees can use the devices and technologies they find most useful in a safe, secure and managed way”. (FUJITSU, 2016)

⁹ Convênios n. 01.001.10.98 e 01.002.10.2001, firmados em 12 de maio de 1998 e 4abr. 2001 pelo Tribunal Regional da 3ª Região, em nome da Justiça Federal de 1ª Instância em São Paulo

realização de projetos para a área de orçamento e finanças da Justiça Federal de São Paulo por alunos do Curso de Graduação em Administração Pública da FGV, em regime de estágio, sob a supervisão de professores e monitores da FGV e de servidores da Justiça Federal. Os professores e monitores eram remunerados por hora de trabalho e os estagiários por meio de uma bolsa-estágio.

No bojo dessa parceria, que vigorou até novembro de 2003, foram desenvolvidos diversos sistemas em MS Access voltados ao controle de arrecadação, de execução orçamentária e de pagamentos. Posteriormente, em 2008, com o objetivo de manter os sistemas desenvolvidos pelos alunos da FGV e desenvolver novos sistemas do mesmo tipo, foi criada a Seção de Modernização de Projetos Orçamentários (SUDO), sucedida em 2009 pela Seção de Apoio ao Desenvolvimento de Projetos (SUDT) e em 2011 pela Seção de Modernização de Projetos Orçamentários e Financeiros (SUMF). Como o MS Access era parte do pacote MS Office, disponibilizado a todos os usuários da Justiça Federal de São Paulo, as atividades da SUMF não eram vistas como serviços de TI. Além disso, os sistemas criados e mantidos pela SUMF eram utilizados por um grupo muito restrito de servidores da área administrativa.

A situação mudou a partir de 2016. A manutenção dos sistemas em MS Access tornou-se progressivamente mais difícil em razão de problemas de ordem técnica e jurídico-administrativa. No aspecto técnico, o uso de computadores em rede tornou-se muito comum e a tecnologia utilizada não era adequada para o uso em rede; no aspecto jurídico-administrativo, tomou-se a decisão de não renovar a licença do MS Access a partir de 2016. Paralelamente a isso, alguns sistemas da SUMF passaram a ser utilizados por um conjunto bem maior de usuários. Um desses sistemas, por exemplo, o de diárias e passagens, tem como usuários potenciais todos os servidores e magistrados da Justiça Federal de São Paulo, ou seja, mais de 4.500 pessoas. Isso tornou necessário que todos os sistemas então mantidos pela SUMF fossem rapidamente migrados

(JFPI/SP), e a Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas (EAESP/FGV), em prol do “desenvolvimento do Programa de Implantação da Qualidade Total da JFPI/SP e, simultaneamente, do aperfeiçoamento dos alunos do curso de Administração Pública da EASP/FGV, com a coordenação no Núcleo Financeiro e Orçamentário da JFPI/SP”. O gestor do projeto pela FGV era o Professor David Hastings. A atividade dos alunos era supervisionada pela Professora Ana Leda Moraes. Essas e as demais informações sobre a história desse projeto eu devo ao servidor Sérgio Tinoco Cordeiro Filgueiras, a quem agradeço por resgatá-las.

para tecnologias que não apresentassem as mesmas limitações. Pela gratuidade e relativa facilidade de aprendizado e implementação, optou-se por reescrever os sistemas para a Web (HTML-CSS-Javascript), utilizando, no *back-end*, o *stack* Apache-MySQL-PHP. Posteriormente, no lugar do MySQL, passou-se a utilizar o SQL Server, que era o sistema de gerenciamento de banco de dados preferencialmente utilizado na Justiça Federal de São Paulo para aplicações administrativas.

Como os sistemas em questão eram críticos e o departamento de TI não tinha condições de assumi-los no curto ou médio prazo, a SUMF recebeu acesso a equipamentos de infraestrutura para que ela própria pudesse continuar cuidando da implantação e da disponibilidade de seus sistemas. Ao mesmo tempo, no entanto, seus integrantes permaneciam sem ter privilégios de administradores e não podiam instalar ferramentas profissionais de desenvolvimento. Trabalhavam de modo artesanal, recorrendo a seus próprios equipamentos pessoais para conseguir atingir nível mínimo de produtividade; seus sistemas seguiam padrões próprios de desenvolvimento, em grande parte desvinculados dos padrões adotados pelo departamento de TI, e ficavam disponíveis somente na Intranet, diferentemente da maioria dos outros sistemas da Justiça Federal de São Paulo, que podiam ser acessados também externamente, mediante autenticação.

Nesse cenário, pode-se indagar se não teria sido mais fácil incorporar a SUMF ao departamento de TI. Esse talvez tivesse sido o desfecho natural, não fosse por uma peculiaridade da estrutura organizacional da Justiça Federal de São Paulo: desde 2007 todos os serviços de TI foram concentrados em uma única secretaria vinculada ao Tribunal Regional Federal da 3ª Região, a Secretaria de Tecnologia da Informação – SETI, ao passo que os integrantes da SUMF permaneciam vinculados à Justiça Federal de 1ª Instância. Em vista disso, a SUMF tornou-se uma espécie de *shadow IT*, pois suas atividades, mesmo sendo conhecidas e toleradas, não eram plenamente *reconhecidas* pela instituição.

Era essa a situação com a qual nos deparamos no início de 2018, quando começamos a expandir as atividades do iJuspLab para a área tecnológica. Apesar de se tratar de situação provisória, até então indesejada e potencialmente causadora de atritos, foi ela, no final das contas, que nos permitiu pensar no laboratório como um espaço que permitiria trazer à luz as iniciativas de *shadow IT* existentes na Justiça Federal de São Paulo, especialmente as da SUMF, sem deixar de lado a

necessidade de seguir os padrões de segurança e controle dos serviços de TI. Ou seja, graças à situação peculiar da SUMF, abriu-se na Justiça Federal de São Paulo a oportunidade de criar uma espécie de *sandbox* (espaço delimitado e seguro) para prototipar e testar soluções tecnológicas. Daí surgiu a ideia de criar uma incubadora de soluções tecnológicas, que seria justamente esse espaço onde a atividade de prototipagem poderia ser exercida e estimulada, mas sem conflitar com as políticas internas de segurança ou com os interesses da instituição.

A INCUBADORA E OS SERVIÇOS DE TI

Como explicado no site da Justiça Federal de São Paulo, chama-se de incubadora um ambiente que “proporciona condições ideais para o desenvolvimento e crescimento de um organismo”¹⁰. No âmbito da inovação, fala-se principalmente em incubadoras de empresas, mas esse conceito também se mostrou adequado para definir o ambiente que pretendíamos criar no laboratório para estimular a inovação tecnológica pelas áreas de negócios. Queríamos que o laboratório pudesse proporcionar os meios adequados para dar vida a iniciativas desse tipo, testar sua viabilidade e torná-las suficientemente maduras para futura implantação. Esse espaço seria, assim, um meio de transformar potenciais iniciativas de *shadow IT* em projetos de inovação tecnológica conduzidos com responsabilidade e com apoio institucional.

A Incubadora de Soluções Tecnológicas do iJuspLab foi criada pela Portaria n. 36, de 13 de agosto de 2018, da Diretoria do Foro, posteriormente alterada pela Portaria n. 49, de 20 de dezembro de 2018. Suas atribuições estão assim definidas nos referidos normativos:

Art. 55-C. Para fomentar a modernização das ferramentas de trabalho na Administração, fica instituída a Incubadora de Soluções Tecnológicas do iNovaJusp, com as seguintes atribuições:

I – Prospectar e desenvolver novas soluções tecnológicas para melhorar os serviços da Justiça Federal, propondo-as aos órgãos competentes e auxiliando-os em sua implantação e manutenção; e

II – Auxiliar os demais órgãos da Administração a adotar e utilizar técnicas e soluções tecnológicas de modo eficiente, segundo os valores da inovação, dando-lhes assessoria e treinamento nessas áreas.

¹⁰ <<http://www.jfsp.jus.br/inovajusp/incubadora-de-solucoes-tecnologicas/>>. Acesso em: 6 abr. 2019.

•• Inovação no Judiciário

Art. 55-D. Para bem desempenhar suas atribuições, a Incubadora de Soluções Tecnológicas poderá:

I – Acolher colaboradores internos e externos, públicos ou privados, que se disponham a melhorar as soluções por ela desenvolvidas.

II – Realizar atividades de prototipagem com os recursos disponíveis, desde que não haja violação das regras de Tecnologia de Informação em vigor na Justiça Federal.

III – Manter relação de interlocução e colaboração direta com a Secretaria de Tecnologia da Informação – SETI do Tribunal Regional Federal da 3ª Região, podendo solicitar ao referido órgão, a qualquer tempo, para si ou para outros órgãos da Diretoria do Foro, os recursos de infraestrutura que entender necessários ou convenientes para seus próprios fins ou para os fins dos outros órgãos solicitantes.

IV – Propor à Diretoria do Foro a celebração de convênios e acordos de cooperação em sua área de atuação.

V – Colaborar em projetos inovadores da área tecnológica desenvolvidos e mantidos por terceiros, desde que sejam de interesse e estejam no escopo dos serviços da Justiça Federal ou da própria Incubadora; e

VI – Manter repositórios públicos ou privados relacionados a seus projetos de inovação tecnológica.

Parágrafo único – Para os fins previstos no inciso VI do ‘caput’, a Diretoria do Foro poderá abrir contas institucionais em portais de repositórios reconhecidos como padrões de indústria.

Art. 55-E. A Incubadora de Soluções Tecnológicas deverá:

I – Cuidar para que os colaboradores internos e externos atuem sempre segundo o interesse público e os princípios da inovação.

II – Utilizar preferencialmente recursos tecnológicos gratuitos, que sigam a filosofia do código aberto (open source).

III – Atribuir a seus projetos licenças de uso compatíveis com a filosofia do código aberto (open source), dando preferência às licenças padronizadas mais comumente adotadas na comunidade de software livre, podendo vedar, quando entender adequado ao projeto, o uso para fins comerciais.

IV – Observar e zelar para que sejam observadas todas as regras editadas pelos órgãos superiores, especialmente as do Conselho Nacional de Justiça, do Conselho da Justiça Federal e do Tribunal Regional Federal da 3ª Região, para o desenvolvimento e uso seguro de soluções tecnológicas.

V – Utilizar padrões profissionais de desenvolvimento e empregar preferencialmente meios, técnicas, plataformas, metodologias e frameworks que permitam e estimulem o trabalho em equipe e em regime de colaboração, tais com as técnicas e metodologias de desenvolvimento ágil.

VI – Manter repositórios públicos ou privados relacionados a seus projetos de inovação tecnológica e aos projetos da Equipe de Gestão de Dados.

Trata-se, portanto, de um espaço de prospecção, prototipagem e desenvolvimento colaborativo. Suas atividades se distinguem dos serviços de TI, porque, mesmo quando seus projetos envolvem o desenvolvimento de *software*, a Incubadora não pode incorporar novas soluções ao parque tecnológico da Justiça Federal de São Paulo sem prévia autorização.

Essa aparente limitação da Incubadora é, na verdade, o que a torna um lugar adequado para a inovação. Se tivesse de se preocupar com a manutenção do parque tecnológico da Justiça Federal de São Paulo, ela acabaria por incorporar os mesmos hábitos e características que tornam os departamentos de TI refratários à inovação. Aos poucos passaria a evitar experimentos e mudanças; progressivamente, diminuiriam seu tempo e motivação para acolher novos projetos das áreas de negócios.

Assim, é muito bom e desejável que os membros da Incubadora não tenham privilégios de administrador, que não tenham liberdade para lidar com infraestrutura ou serviços próprios de TI e que haja uma regra clara que restrinja o seu âmbito de atuação ao campo da prospecção, da prototipagem e do desenvolvimento colaborativo¹¹.

Outro aspecto importante da Incubadora é o de servir como “ponte” entre as áreas de negócios e o departamento de TI. Por conhecer ambos os mundos, ela é capaz de “filtrar” as demandas dos usuários, de modo a fazer chegar ao departamento de TI somente aquelas que realmente precisam de atenção. Além disso, ela pode também ajudar os usuários a formular tais demandas de um modo que seja mais facilmente compreensível pelos técnicos da área de TI. Portanto, a Incubadora não é “concorrente” do departamento de TI, mas, uma importante aliada sua no que diz respeito à administração das demandas internas por inovação tecnológica e por melhorias nos sistemas informatizados.

As atividades da Incubadora foram inicialmente conduzidas pela SUMF. Posteriormente, com a criação da Subsecretaria de Comunicação, Conhecimento e

¹¹ Esse é, no fundo, o modelo de inovação da “*startup enxuta*” (*lean startup*), o qual, apesar do nome, pode ser replicado em quaisquer tipos de organização. O modelo é assim explicado por RIES (2011, p. 5096-5099): “O desafio aqui é criar um mecanismo para delegar poder às equipes de inovação às claras, sem nada ocultar. Esse é o caminho rumo a uma cultura sustentável de inovação ao longo do tempo, pois as empresas enfrentam ameaças existenciais frequentes. Minha proposta é criar uma área restrita (*sandbox*) para a inovação, que conterà o impacto da nova inovação, mas sem restringir os métodos da equipe da *startup*”.

Inovação da Justiça Federal de São Paulo, a SUMF foi extinta e sucedida pelo Núcleo de Inovação Tecnológica (NUIT).

O ato de criação dessa nova Subsecretaria, a Resolução n. 37, de 28 de janeiro de 2019, do Conselho da Justiça Federal da 3ª Região, é um marco importante na breve história da Incubadora, porque por meio dele o Tribunal Regional Federal da 3ª Região referendou a criação do iJuspLab e adotou oficialmente o nosso entendimento de que as atividades do NUIT, órgão de inovação tecnológica responsável pela Incubadora, não se confundem com os serviços de TI prestados pela SETI.

A composição da Incubadora não se restringe, contudo, aos membros do NUIT. Conforme as normas do Programa de Gestão e Inovação da Justiça Federal de São Paulo – iNovaJusp, podem também integrar a Incubadora outros servidores públicos nomeados por meio de portaria específica, pertencentes ou não aos quadros da administração. Atualmente há três outros integrantes da Incubadora que não pertencem ao NUIT: um técnico de TI e dois contadores.

METODOLOGIAS E FERRAMENTAS DE TRABALHO

Como mencionado anteriormente, as principais atividades da Incubadora são: prospecção, prototipagem e desenvolvimento colaborativo. No desempenho dessas funções, ela cuida também do encaminhamento de demandas ao departamento de TI. Além disso, os membros da Incubadora precisam ser capacitados para o desempenho de cada uma dessas atividades. Passamos a tratar de cada um desses temas em separado.

Prospecção

A prospecção é uma atividade contínua, que requer o acompanhamento de publicações especializadas em meio físico ou eletrônico. O objetivo é que os membros da Incubadora estejam sempre cientes das últimas tendências e avanços na área da tecnologia e possam utilizar esse conhecimento para decidir melhor sobre quais tecnologias e ferramentas utilizar em cada novo projeto.

Todavia, seria um grande erro achar que a Incubadora deve adotar e testar toda e qualquer nova tecnologia prospectada. Ao contrário, convém que a Incubadora trabalhe com um conjunto pequeno de ferramentas e não varie tanto o seu repertório, optando sempre pelas mais conhecidas e disseminadas. A inovação não

requer alteração constante; é perfeitamente possível e aceitável inovar usando o mesmo conjunto de ferramentas por anos a fio, desde que tais ferramentas sejam adequadas ao trabalho. Ademais, o uso de um conjunto mais restrito de ferramentas facilita o aprendizado e a colaboração, simplifica o trabalho de prototipagem e evita que os protótipos sejam depois rejeitados pelo departamento de TI por falta de técnicos habilitados a trabalhar com as tecnologias escolhidas. Há, ainda, riscos de ordem pessoal, relacionados à qualidade de vida: querer conhecer e testar tudo pode causar ansiedade e paralisia, especialmente diante da quantidade de novidades que aparecem todos os dias nessa área.

Por essas razões, os membros da Incubadora são orientados a cuidar da prospecção com serenidade e distanciamento, como quem acompanha o noticiário pelos jornais. Quando indagados sobre alguma novidade tecnológica, devem ser capazes de dizer “ouvi falar disso em algum lugar, sei onde buscar”, mas não precisam necessariamente conhecer a nova tecnologia nem tampouco saber utilizá-la.

É importante também distinguir a atividade de prospecção, que todos os membros da Incubadora devem realizar, da atividade de reunir as notícias relacionadas a inovações tecnológicas. Esse trabalho de compilação é atualmente realizado pela Biblioteca Central da Justiça Federal de São Paulo, não pela Incubadora. É claro que convém aos membros da Incubadora examinar esse material sempre que disponibilizado, mesmo que o façam rapidamente, numa leitura superficial, mas a prospecção principal deve se dar pela atividade diária de leitura de *blogs*, *podcasts*, *tweets*, *repository watching* etc. Nem todos precisam ler tudo. É normal que cada um tenha suas preferências. O essencial é que a atividade faça parte da rotina, que se crie o hábito de prospectar.

Prototipagem

A prototipagem é a “alma” da Incubadora. No site *usability.gov*, uma excelente base de conhecimento sobre melhores práticas e orientações em matéria de UX (*user experience*), o termo “protótipo” é definido como a “versão preliminar de um produto que permite explorar as ideias e intenções por trás de um recurso ou mostrar aos usuários o conceito geral do design antes de investir tempo e dinheiro no desenvolvimento”¹².

¹² Disponível em: <<https://www.usability.gov/how-to-and-tools/methods/prototyping.html>>. Acesso em: 6 abr. 2019.

Mesmo na Incubadora, os protótipos nem sempre precisam ser programas de computador. É possível criar tanto protótipos de baixa fidelidade, como, por exemplo, desenhos de papel ou planilhas de cálculo que apenas indicam algumas das características principais da solução pretendida, quanto protótipos de alta fidelidade, bem próximos da solução tecnológica que se pretende implementar¹³.

O importante é conseguir atingir o máximo de clareza sobre o que se pretende com um mínimo de recursos. Às vezes também o grau de fidelidade do protótipo dependerá de considerações práticas, como, por exemplo, sobre a conveniência de avançar o máximo possível para poupar o departamento de TI de atividades de desenvolvimento e viabilizar uma implantação mais rápida da solução tecnológica. Nesses casos, o protótipo pode chegar muito perto de um sistema completo e acabado.

A breve história da Incubadora é rica em exemplos de protótipos de baixa e alta fidelidade.

O primeiro projeto desenvolvido pela Incubadora em conjunto com a SETI envolveu a implementação de um algoritmo de emparelhamento estável (*stable matching*) no sistema de recursos humanos da Justiça Federal de São Paulo, para possibilitar a movimentação de servidores públicos que desejam mudar de cidade, mediante triangulação.

Depois de alguma pesquisa, foi identificado um algoritmo criado nos Estados Unidos pelo *National Resident Matching Program* (NRMP)¹⁴ com base no trabalho

¹³ “There is an ongoing debate about using low versus high fidelity prototyping and how much a prototype should resemble the final version of your design. Both have been found to be basically equivalent in finding usability issues (Walker et al 2002). With that said, there are things to consider when trying to decide which option is best for your project: – Low-fidelity prototypes are often paper-based and do not allow user interactions. They range from a series of hand-drawn mockups to printouts. In theory, low-fidelity sketches are quicker to create. Low-fidelity prototypes are helpful in enabling early visualization of alternative design solutions, which helps provoke innovation and improvement. An additional advantage to this approach is that when using rough sketches, users may feel more comfortable suggesting changes. High-fidelity prototypes are computer-based, and usually allow realistic (mouse-keyboard) user interactions. High-fidelity prototypes take you as close as possible to a true representation of the user interface. High-fidelity prototypes are assumed to be much more effective in collecting true human performance data (e.g., time to complete a task), and in demonstrating actual products to clients, management, and others”. Ibidem.

¹⁴ <<http://www.nrmp.org>>. Acesso em: 6 abr. 2019.

de dois economistas premiados com o Nobel em 2012, Alvin E. Roth e Lloyd S. Shapley¹⁵. O algoritmo é utilizado para automatizar e otimizar a alocação de médicos recém-formados para postos de residência abertos em hospitais. Para tanto, as opções dos médicos são emparelhadas com a ordem de preferência informada pelos hospitais participantes até que se atinja a melhor acomodação possível entre os interesses envolvidos¹⁶.

A ideia que tivemos na Incubadora foi adotar a mesma estratégia para a movimentação de servidores, considerando, no lugar do médico recém-formado, o servidor, e, no lugar dos hospitais, as sedes das subseções judiciárias.

Para mostrar a viabilidade dessa ideia, foi criado um protótipo em Google Sheets que permitia a inclusão de candidatos fictícios, com as respectivas lotações e tempo de carreira e a execução do algoritmo adaptado em Google Apps Script (uma versão simplificada do Javascript que pode interagir com os serviços do G Suite) que, além de realizar a triangulação, gerava um texto em linguagem natural explicando para o humano leigo cada uma das etapas percorridas, a fim de dar transparência ao processo e permitir eventual questionamento.

Essa abordagem tornou possível testar, de forma bastante simples e rápida, diversos cenários distintos antes de iniciar o desenvolvimento propriamente dito da solução, o que nos deu confiança de que o projeto seria realmente viável. Além disso, a facilidade de fazer alterações e de experimentar nos fez perceber muito cedo quais seriam as adaptações necessárias para atender as regras de negócio específicas da Justiça Federal de São Paulo, o que certamente abreviou o tempo de desenvolvimento.

Todo o processo de prototipagem dessa solução não levou mais que duas ou três semanas no total, mas foi determinante para que pudéssemos seguir à fase de desenvolvimento com a parte mais complexa do problema resolvida.

As figuras seguintes mostram as partes principais do protótipo, dando uma ideia geral de seu funcionamento.

¹⁵ <<https://www.nobelprize.org/prizes/economic-sciences/2012/summary/>>. Acesso em: 6 abr. 2019.

¹⁶ <<http://www.nrmp.org/matching-algorithm/>>. Acesso em: 6 abr. 2019.

•• Inovação no Judiciário

Figura 1. Projeto Movimentação de Servidores. Detalhe da planilha que simula as inscrições.

	A	B	C	D	E	F	G	H
1	Código	Nome	Classificação	Especialidade	Lotação	Descrição	Preferências	
2	A	<input checked="" type="checkbox"/> Alfredo	9	1 - Analista	1	São Paulo	3,4,2,5,1	
3	B	<input checked="" type="checkbox"/> Bruno	8	1 - Analista	3	Santos	2,4,1,5,3	
4	C	<input checked="" type="checkbox"/> Carlos	7	2 - Técnico	2	Ribeirão Preto	1,3,5,4,2	
5	D	<input checked="" type="checkbox"/> Daniela	10	1 - Analista	4	Campinas	3,1,5,2,4	
6	E	<input checked="" type="checkbox"/> Eloísa	6	2 - Técnico	5	P. Prudente	1,4,3,2,5	
7	F	<input checked="" type="checkbox"/> Fabiana	5	1 - Analista	1	São Paulo	3,4,2,5,1	
8	G	<input checked="" type="checkbox"/> Gabriel	4	3 - Agente	3	Santos	2,4,1,5,3	
9	H	<input checked="" type="checkbox"/> Hélio	3	1 - Analista	2	Ribeirão Preto	1,3,5,4,2	
10	I	<input checked="" type="checkbox"/> Igor	1	3 - Agente	5	P. Prudente	3,1,4,2,5	
11	J	<input checked="" type="checkbox"/> Juliana	2	2 - Técnico	4	Campinas	1,5,3,2,4	
<div>Adicionar mais</div> <div>1000</div> <div>linhas ao fim.</div>								

Fonte: Gestão de Dados da Justiça Federal de Primeiro Grau em São Paulo.

Figura 2. Projeto Movimentação de Servidores. Detalhe da planilha que simula as preferências.

	A	B	C	D	E	F	G	H	I
1	Código	Incluir	Descrição	Total Vagas	Vagas Analistas	Vagas Técnicos	Vagas Agentes	Servidores Lotados	Prioridades
2	1	<input checked="" type="checkbox"/>	São Paulo	-	2,00	(2,00)	-	A,F	A,F,I,J,H,G,E,C,B,D
3	2	<input checked="" type="checkbox"/>	Ribeirão Preto	-	(2,00)	2,00	-	C,H	C,H,I,J,G,F,E,B,A,D
4	3	<input checked="" type="checkbox"/>	Santos	-	-	-	-	B,G	B,G,I,J,H,F,E,C,A,D
5	4	<input checked="" type="checkbox"/>	Campinas	-	4,00	(4,00)	-	D,J	D,J,I,H,G,F,E,C,B,A
6	5	<input checked="" type="checkbox"/>	P. Prudente	-	(1,00)	1,00	-	E,I	E,I,J,H,G,F,C,B,A,D
	Adicionar mais		1000	linhas ao fim.					

Fonte: Gestão de Dados da Justiça Federal de Primeiro Grau em São Paulo.

Figura 3. Projeto Movimentação de Servidores. Resultado da movimentação.

	A	B	H	I	J	K	L	M	N	O	P	Q	R	S
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40														
41														
42														
43														
44														
45														
46														
47														
48														
49														
50														
51														
52														
53														
54														
55														
56														
57														
58														
59														
60														
61														
62														
63														
64														
65														
66														
67														
68														
69														
70														
71														
72														
73														
74														
75														
76														
77														
78														
79														
80														
81														
82														
83														
84														
85														
86														
87														
88														
89														
90														
91														
92														
93														
94														
95														
96														
97														
98														
99														
100														
101														
102														
103														
104														
105														
106														
107														
108														
109														
110														
111														
112														
113														
114														
115														
116														
117														
118														
119														
120														
121														
122														
123														
124														
125														
126														
127														
128														
129														
130														
131														
132														
133														
134														
135														
136														
137														
138														
139														
140														
141														
142														
143														
144														
145														
146														
147														
148														
149														
150														
151														
152														
153														
154														
155														
156														
157														
158														
159														
160														
161														
162														
163														
164														
165														
166														
167														
168														
169														
170														
171														
172														
173														
174														
175														
176														
177														
178														
179														
180														
181														
182														
183														

Fonte: Gestão de Dados da Justiça Federal de Primeiro Grau em São Paulo.

Figura 4. Projeto Movimentação de Servidores. Operação do algoritmo em linguagem natural.

=STABLE_MATCH(CodigoServidor;Especialidade;Lotacao;Classificacao;Preferencias;IncluirServidor;CodigoUnidade;P					
A					
5 unidades: 1, 2, 3, 4, 5					
10 servidores: I, J, H, G, F, E, C, B, A, D					
=====					
Início das Tentativas de Permuta					
=====					
... permanecem pendentes o(s) servidor(es) I, J, H, G, F, E, C, B, A, D					
### servidor I (agente), opções: 3, 1, 4, 2, 5 ###					
>>> buscando permuta para o servidor I (agente)					
>>> tentativa de permutar o servidor com excedentes da unidade 3 (0 vaga(s))					
>>> não há espaço na unidade para agente					
>>> tentativa de permutar o servidor com excedentes da unidade 1 (0 vaga(s))					
>>> não há espaço na unidade para agente					
>>> tentativa de permutar o servidor com excedentes da unidade 4 (0 vaga(s))					
>>> não há espaço na unidade para agente					
>>> tentativa de permutar o servidor com excedentes da unidade 2 (0 vaga(s))					
>>> não há espaço na unidade para agente					
<<< opções esgotadas >>>					
<<< servidor I mantido em sua unidade de origem (5) >>>					
### servidor J (técnico), opções: 1, 5, 3, 2, 4 ###					
>>> buscando permuta para o servidor J (técnico)					
>>> tentativa de permutar o servidor com excedentes da unidade 1 (2 vaga(s))					
>>> tentativa de permutar com o servidor F (analista)					
>>> não há vagas disponíveis na unidade do candidato para o servidor F					
>>> tentativa de permutar com o servidor A (analista)					
>>> não há vagas disponíveis na unidade do candidato para o servidor A					
>>> tentativa de permutar o servidor com excedentes da unidade 5 (-1 vaga(s))					
>>> não há espaço na unidade para técnico					
>>> tentativa de permutar o servidor com excedentes da unidade 3 (0 vaga(s))					
>>> não há espaço na unidade para técnico					
+	☰	Premissas ▾	Servidores ▾	Unidades ▾	Resultado ▾
					Descrição ▾

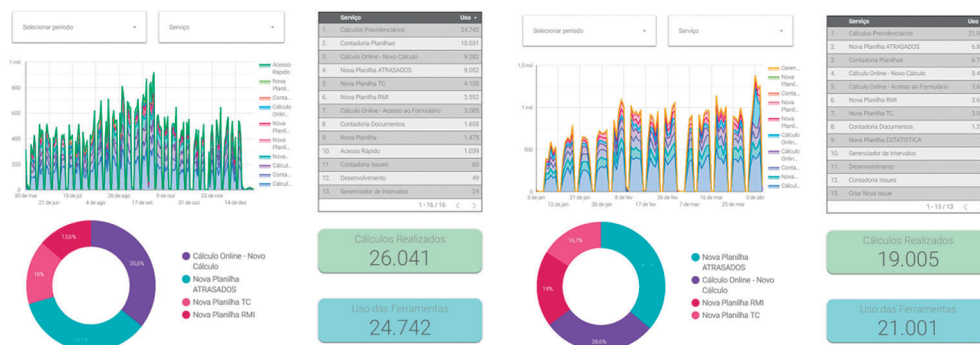
Fonte: Gestão de Dados da Justiça Federal de Primeiro Grau em São Paulo.

Outro caso interessante de prototipagem, também em Google Sheets, é o projeto denominado “Contadoria”, iniciado em 2016 pelo Gabinete da Coordenação dos Juizados Especiais Federais de São Paulo e trazido para a Incubadora em

2018. Esse projeto envolveu a criação de planilhas em nuvem para o cálculo de benefícios previdenciários juntamente com um “ecossistema” de ferramentas de desenvolvimento, distribuição e preenchimento automático, envolvendo a criação de complementos (*add-ons*) e aplicativos (*web apps*) escritos em Google Apps Script.

Os protótipos vêm sendo testados há mais de dois anos por usuários em diversas localidades do Brasil, incluindo varas, juizados e advogados e foram homologados pelo Tribunal Regional Federal da 3ª Região no início de 2018. Para verificar o desempenho das ferramentas, além do *feedback* pessoal dos usuários, foram colhidas métricas de uso diário a partir de maio de 2018. Até 6abr. 2019 tinham sido realizados 45.046 cálculos e as ferramentas auxiliares haviam sido utilizadas 45.743 vezes.

Figura 5. Métricas de uso colhidas em 2018 (maio-dez.) e 2019 (jan.-ab.).



Fonte: Gestão de Dados da Justiça Federal de Primeiro Grau em São Paulo.

Recentemente foi iniciado o processo de integração desses protótipos ao parque tecnológico da Justiça Federal de São Paulo, em colaboração com a SETI, mediante a migração das planilhas e de todo o seu “ecossistema” para tecnologias compatíveis com os recursos internos de TI.

Desenvolvimento colaborativo

No clássico ensaio *The Cathedral and the Bazaar*, Eric Raymond relata de forma envolvente como nasceu o *software* de código aberto mais bem-sucedido da história – o Linux – e como esse fato marcou para sempre a forma como a indústria de *software* passou a ver o desenvolvimento colaborativo.

Antes do surgimento do Linux, havia comunidades de desenvolvedores trabalhando em projetos comuns de código aberto. Imaginava-se, todavia, que o modo aparentemente caótico como se dava a interação entre os desenvolvedores nessas

comunidades, semelhante à confusão de sons e vozes em uma feira ou *bazaar*, não era apropriado para a criação de *softwares* complexos como os sistemas operacionais. Acreditava-se que tais tipos de *software* somente poderiam ser adequadamente desenvolvidos, em todas as suas partes, se o trabalho fosse conduzido de forma sistemática por um grupo previamente definido de pessoas e se houvesse planejamento detalhado, à semelhança de como são construídas as catedrais.

A história do Linux provou o contrário. Não só foi possível criar um sistema operacional pelo método do *bazaar*, mas um sistema operacional dos mais robustos e confiáveis do mercado, inclusive no que diz respeito ao uso corporativo¹⁷. Esse método vem sendo usado com sucesso pelo criador do *kernel* (núcleo) do Linux (Linus Torvalds) e por seus colaboradores desde 1991, ininterruptamente¹⁸.

Os traços principais desse método de desenvolvimento são (i) código-fonte aberto (*open source*); (ii) entregas rápidas e contínuas (*continuous delivery*); (iii) *feedback* contínuo; (iv) comunidade aberta, em que usuários podem tornar-se co-desenvolvedores a qualquer tempo¹⁹.

¹⁷ “The most important feature of Linux, however, was not technical but sociological. Until the Linux development, everyone believed that any software as complex as an operating system had to be developed in a carefully coordinated way by a relatively small, tightly-knit group of people. This model was and still is typical of both commercial software and the great freeware cathedrals built by the Free Software Foundation in the 1980s; also of the freeBSD/netBSD/OpenBSD projects that spun off from the Jolitzes’ original 386BSD port. Linux evolved in a completely different way. From nearly the beginning, it was rather casually hacked on by huge numbers of volunteers coordinating only through the Internet. Quality was maintained not by rigid standards or autocracy but by the naively simple strategy of releasing every week and getting feedback from hundreds of users within days, creating a sort of rapid Darwinian selection on the mutations introduced by developers. To the amazement of almost everyone, this worked quite well. By late 1993 Linux could compete on stability and reliability with many commercial Unixes, and hosted vastly more software”. RAYMOND, 2001.

¹⁸ Detalhes dessa história fascinante também podem ser lidos no capítulo “Richard Stallman, Linus Torvalds e os Movimentos de Software Livre e Código Aberto” do livro *os Inovadores*, de Walter Isaacson, citado na bibliografia.

¹⁹ “I believed that the most important software (operating systems and really large tools like the Emacs programming editor) needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time. Linus Torvalds’ style of development – release early and often, delegate everything you can, be open to the point of promiscuity – came as a surprise. No quiet, reverent cathedral-building here – rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles. Linus’s open development policy was the very opposite of cathedral-building. Linux’s Internet archives were burgeoning, multiple distributions

Apesar do sucesso comprovado, o desenvolvimento colaborativo desperta ainda alguns receios. Alguns supõem, equivocadamente, que esse método de desenvolvimento facilita a quebra de segredos de negócio e torna os sistemas mais vulneráveis a ataques. Ambos os receios são infundados. A preservação de segredos de negócio não depende tanto do modelo de desenvolvimento adotado, mas da arquitetura do sistema. Definida uma arquitetura adequada, é possível conduzir a parte não sigilosa do projeto pelo método do desenvolvimento colaborativo e a parte restrita pelo método tradicional, com a designação de uma equipe interna, de modo a separar o que pode e o que não pode ser revelado publicamente²⁰. É o que hoje fazem diversos gigantes da indústria de software, como a Microsoft, o Google e o Facebook, que mantêm diversos projetos de código aberto ao lado de seus produtos proprietários²¹.

were being floated. And all of this was driven by an unheard-of frequency of core system releases. Linus was treating his users as co-developers in the most effective possible way: Release early. Release often. And listen to your customers. I think the future of open-source software will increasingly belong to people who know how to play Linus' game, people who leave behind the cathedral and embrace the bazaar. This is not to say that individual vision and brilliance will no longer matter; rather, I think that the cutting edge of open-source software will belong to people who start from individual vision and brilliance, then amplify it through the effective construction of voluntary communities of interest". Ibidem.

²⁰ "Another reason often cited is the fear that disclosing source of a particular special accounting function might be tantamount to revealing confidential aspects of your business plan. This is really an argument not for closed source but against bad design; in a properly-written accounting package, business knowledge should not be expressed in code at all but rather in a schema or specification language implemented by the accounting engine (for a closely parallel case, consider the way that database schemas separate business knowledge from the mechanics of the database engine). The separation of function would enable you to guard the crown jewels (the schema) while getting maximum benefit from open-sourcing the engine". Ibidem.

²¹ Em junho de 2018, a Microsoft adquiriu o GitHub, o maior portal de repositórios de projetos de código aberto na atualidade. Esse portal é uma das várias implementações visuais do Git, uma ferramenta de versionamento de código criada pelo próprio Linus Torvalds para facilitar o desenvolvimento colaborativo do Linux. No GitHub, a Microsoft mantém diversos projetos de código aberto em torno dos quais se formaram enormes comunidades de colaboradores. Entre eles, merecem destaque o projeto "microsoft/vscode", a IDE gratuita mais popular na atualidade, com quase 10.000 colaboradores, e o "microsoft/typescript", uma ferramenta criada para aprimorar o desenvolvimento em Javascript, com mais de 6.000 colaboradores. Também está no GitHub o projeto "officedev/office-js", com o código-fonte das APIs (*application programming interfaces*) em Javascript do MS Office 2013 e seguintes, que permitem criar suplementos para Excel e Word, por exemplo. Além disso, a própria Microsoft tornou-se colaboradora do Linux e distribui versões do referido sistema operacional juntamente com o Windows 10 (*Windows Subsystem for Linux*) e com seus produtos de IoT (*Azure Sphere OS*). Google e Facebook mantêm portais dedicados a seus projetos *open source* (<<https://opensource.google.com/>> e <<https://opensource.facebook.com/>>. Acesso em: 6 abr. 2019).

Quanto ao receio de que a transparência do código-fonte aumente a vulnerabilidade do sistema, o que ocorre, na realidade, é justamente o inverso. Os sistemas de código aberto, desenvolvidos de forma colaborativa, tendem a ser mais seguros, porque testados publicamente por um número grande de pessoas e corrigidos com maior rapidez. Os sistemas de código fechado, por não terem o benefício do escrutínio público, tornam-se mais vulneráveis²².

Outra preocupação comum é a de que outros se aproveitem do trabalho árduo do desenvolvedor abnegado, utilizando-o para fins comerciais. Quanto a isso, é importante notar que as licenças de código aberto não precisam necessariamente autorizar o uso comercial do *software* e, ainda que o façam, podem resguardar expressamente, como, aliás é normal ocorrer, o direito do desenvolvedor de ser reconhecido e citado como autor do código. Além disso, essa preocupação parte da premissa equivocada de que o uso comercial do *software* criado colaborativamente é algo mau. Não é. Além de ser prática comum e aceita há muitos anos entre os desenvolvedores colaborativos, o uso comercial potencializa a descoberta de defeitos e contribui para elevar a qualidade do projeto como um todo. Em suma, o uso do código é o que importa, seja ou não para fins comerciais. Quanto mais utilizado e testado, melhor se torna o *software* e todos acabam sendo beneficiados.

O desenvolvimento colaborativo não é, evidentemente, a resposta para todos os problemas da área de tecnologia. Há sistemas críticos que não podem ser desenvolvidos por esse método. Mas não há dúvida de que esse é o melhor método de trabalho para o desenvolvimento de protótipos num laboratório de inovação, pois ele incorpora, em sua própria essência, os valores da co-criação, da experimentação e do desenvolvimento centrado no usuário.

Além disso, os métodos de desenvolvimento colaborativo, quando implementados de forma adequada, podem servir de incentivo aos usuários mais qualificados na área de tecnologia para que, no lugar de adotarem iniciativas arriscadas de *shadow IT*, possam colaborar em projetos conduzidos pela organização ou iniciar projetos próprios que atraiam colaboradores internos e externos.

²² “There are other reasons for closing source that are outright irrational. You might, for example, be laboring under the delusion that closing the sources will make your business systems more secure against crackers and intruders. If so, I recommend therapeutic conversation with a cryptographer immediately. The really professional paranoids know better than to trust the security of closed-source programs, because they’ve learned through hard experience not to. Security is an aspect of reliability; only algorithms and implementations that have been thoroughly peer-reviewed can possibly be trusted as secure”. RAYMOND, 2001.

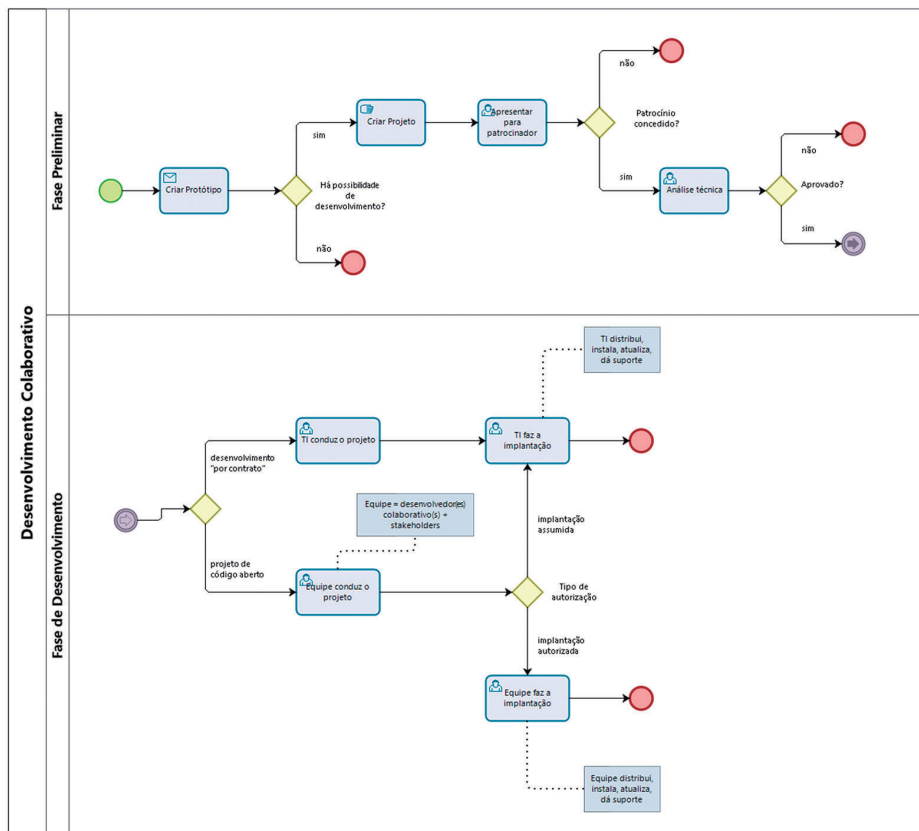
Para estimular o desenvolvimento colaborativo na Justiça Federal, foi apresentado em 2018, ao Tribunal Regional Federal da 3ª Região, um modelo de regimento cujos elementos principais são os seguintes:

- a) Criação de um cadastro de desenvolvedores colaborativos, para reunir todos os potenciais desenvolvedores-cidadãos da Justiça Federal da 3ª Região (Seções Judiciárias de São Paulo e Mato Grosso do Sul), a fim de que possam receber meios adequados e seguros de desenvolver protótipos, prospectar e estudar novas tecnologias e apresentar projetos inovadores.
- b) Criação de um procedimento para a aprovação de projetos inovadores, contemplando, inclusive, as etapas de desenvolvimento e implantação, de modo a assegurar que os projetos estão alinhados com os objetivos estratégicos da instituição e que sua implantação é viável. Nessa fase, o protótipo é apresentado a um possível patrocinador interno, um órgão administrativo com competência para analisar se a solução proposta é de interesse da Justiça Federal, se está alinhada com os objetivos estratégicos da instituição e se não há projeto de solução semelhante em andamento. Sendo concedido o patrocínio, o projeto é submetido à análise técnica do departamento de TI, para verificar a viabilidade de implantação. Superadas essas fases, inicia-se o processo de desenvolvimento.
- c) Estruturação de dois modelos de desenvolvimento para os projetos aprovados: (i) modelo de “código aberto”, em que o desenvolvimento é conduzido pelos próprios proponentes do projeto, com a participação intensiva de um grupo de potenciais usuários da solução (*stakeholders*); e (ii) modelo de “contrato”, em que os proponentes do projeto oferecem ao departamento de TI auxílio na criação ou aperfeiçoamento de soluções e o departamento de TI conduz todo o processo.
- d) Estruturação de dois modelos de implantação para o desenvolvimento em “código aberto”: (i) implantação autorizada, em que, dentro das condições estabelecidas pelo departamento de TI quanto ao universo de usuários, recursos disponíveis, modo de instalação e uso etc., a implantação e manutenção da solução (salvo apenas pela parte da infraestrutura de TI) ficam a cargo da própria equipe de desenvolvimento, com adesão voluntária pelos usuários; (ii) implantação assumida, em que o departamento de TI se responsabiliza pela implantação.

Esse modelo de desenvolvimento colaborativo vai além das atividades da Incubadora, pois contempla, inclusive, a criação de protótipos fora do iJusPLab. Pareceu-nos importante, todavia, preservar essa liberdade, evitando dar ao iJusPLab o monopólio sobre as iniciativas de inovação, pois isso poderia acabar sufocando tais iniciativas no lugar de incentivá-las. Além disso, o iJusPLab continua a ter um papel relevante nesse modelo, porque (i) o ambiente do laboratório continuaria a atrair os usuários-desenvolvedores que estivessem em busca de colaboradores e de maior apoio institucional para a criação de seus protótipos; e (ii) a Incubadora continuaria envolvida em todos os projetos inovadores originários da Justiça Federal de São Paulo, que participaria do processo de aprovação desses projetos.

A versão do normativo proposto apresentada ao Tribunal, ainda não aprovada, encontra-se no Anexo I. A figura abaixo ilustra as características principais do modelo de desenvolvimento colaborativo ali previsto.

Figura 6. Modelo de desenvolvimento colaborativo proposto ao TRF.



Fonte: Justiça Federal de Primeiro Grau em São Paulo.

A adoção de métodos de desenvolvimento colaborativo pela Incubadora traz algumas consequências importantes, dentre as quais vale destacar as seguintes:

Crítérios de escolha das tecnologias

As ferramentas e as tecnologias empregadas devem ser apropriadas para o desenvolvimento colaborativo. Assim, devem ser preferidas as tecnologias mais populares, *open source* e de uso gratuito²³, que tenham em seu entorno uma comunidade grande e ativa, ainda que tais tecnologias não sejam as melhores do ponto de vista técnico. PHP e Javascript, por exemplo, são duas linguagens bastante utilizadas em projetos de desenvolvimento colaborativo, mas que durante muito tempo foram consideradas tecnicamente inadequadas para o desenvolvimento de aplicações mais complexas e robustas²⁴.

²³ Essas duas características (código aberto e licença gratuita) constituem os elementos essenciais do *software* livre. Em 2003, no Acórdão n. 1.521/2003, ao analisar a contratação de produtos da Microsoft sem licitação pela administração pública federal, o Tribunal de Contas da União alertava sobre a conveniência de adotar preferencialmente as soluções de *software* livre. O acórdão pode ser obtido no endereço <https://pesquisa.apps.tcu.gov.br/#/documento/acordao-completo/*/NUMACORDAO%253A1521%2520ANOACORDAO%253A2003/DTRELEVANCIA%20desc,%20NUMACORDAOINT%20desc/0/%20?uuid=7f77e370-5ad8-11e9-ae50-7b8f165af2f8>. Acesso em: 9 abr. 2019. Nesse acórdão, diz o TCU, citando o parecer do Ministério Público, que “os softwares livres se diferenciam dos *softwares* proprietários – como os da Microsoft – por não serem protegidos pelos direitos autorais clássicos – o chamado *copyright* –, que impedem a execução, a reprodução, a divulgação ou a alteração dos programas sem a autorização dos titulares dos direitos. Os softwares livres, utilizados mediante ‘licença’ diferenciada, que mais se assemelha a uma simples adesão, podem ser executados e reproduzidos gratuitamente, bem como livremente aperfeiçoados e adequados às necessidades do usuário (...)”, e cita as seguintes vantagens desse modelo: (i) o *software* livre poder ser constantemente evoluído e adaptado às necessidades do cliente a um custo, em princípio, baixo; (ii) o *software* livre apresenta tempo de obsolescência menor que o do *software* proprietário; (iii) os requisitos de funcionamento do *software* livre, no que se refere ao *hardware*, costumam ser menores em comparação com os do *software* proprietário; (iv) o *software* livre costuma apresentar boa compatibilidade com diversos ambientes e aplicações; (v) há, no mercado, uma crescente oferta de serviços relacionados ao suporte, consultoria, desenvolvimento e capacitação em *software* livre; (vi) o custo total da implantação do *software* livre é significativamente mais baixo que o do *software* proprietário. Em razão disso, concluiu o relator que “a alternativa do ‘software livre’ poderá significar economia substancial aos cofres públicos, além de segurança e independência para o Estado brasileiro”.

²⁴ O Javascript, por exemplo, foi uma linguagem desenvolvida em apenas 10 dias, no ano de 1995, em plena “guerra dos browsers”, quando Netscape e Internet Explorer disputavam a hegemonia na Internet. Seu criador, Brendan Eich, que se tornou depois um dos fundadores do Projeto Mozilla, recebeu a missão de apresentar uma linguagem de *script* que fosse de algum modo semelhante ao Java (daí a semelhança também no nome) e que pudesse ser executada pelo browser.

É importante também levar em conta que esses critérios raramente coincidem com os adotados pelo departamento de TI. Por terem um perfil mais conservador, os departamentos de TI tendem a dar maior ênfase à segurança e a tecnologias proprietárias, que oferecem garantias contratuais. Assim, para resguardar a futura implantação interna da solução, não se pode deixar de levar em conta também as preferências do departamento de TI, a fim de que a solução não termine rejeitada em razão do uso de uma tecnologia de difícil integração com os serviços existentes. Levando em conta todos esses critérios, a Incubadora optou por adotar as mesmas tecnologias que vinham sendo utilizadas pela SUMF (tecnologias Web, com o *stack* Apache-SQL Server-PHP no *back-end*). A razão é que a equipe estava capacitada para o uso dessas tecnologias, as quais, salvo no que diz respeito ao SQL Server, são bastante conhecidas e utilizadas em projetos colaborativos *open source*. O SQL Server entrou na história para atender ao critério da compatibilidade com os serviços de TI existentes, a fim de viabilizar a implantação das soluções internamente.

Em razão do tempo exíguo para a conclusão do trabalho e do uso mais restrito que teria a nova linguagem (foi pensada inicialmente apenas para melhorar a interação dos usuários com os conteúdos dos sites, não para criar aplicações de grande porte – cf. entrevistas dadas por Brendan Eich ao site www.computerworld.com.au e ao editor do livro *Coders at Work*, ambas citadas na bibliografia), o Javascript apresentava diversas deficiências técnicas (sobre esse assunto, vale a pena ler o clássico *Javascript: The Good Parts*, de Douglas Crockford, conhecido, entre outras coisas, pela criação do formato JSON). Apesar disso e da vitória do Internet Explorer, o Javascript acabou se tornando a *lingua franca* da Internet, especialmente após dois marcos importantes na evolução da Web: (i) a criação, pela Microsoft, da programação AJAX (*Asynchronous Javascript and XML*) que, por meio de uma nova API (o objeto XMLHttpRequest), tornou possível a atualização parcial dos sites e facilitou, desse modo, a criação de aplicações *online*; e (ii) a criação, pelo Google, do interpretador V8, que tornou muito mais rápida a execução dos programas em Javascript pelo browser. O Javascript, que oficialmente agora se chama “ECMAScript” (por ser uma linguagem padronizada pela ECMA International), recebeu diversos aperfeiçoamentos importantes a partir do lançamento do ECMAScript 3 em 1999, tanto no que diz respeito a sua arquitetura interna quanto em termos de técnicas e boas práticas de programação introduzidas principalmente pelos inúmeros *frameworks* que foram surgindo ao longo dos anos (jQuery, Backbone, Knockout, YUI, AngularJS, Ember, React, Angular, Vue etc.), até se tornar o que é hoje: uma linguagem robusta, a mais popular de todas entre os desenvolvedores, segundo a última pesquisa do Stack Overflow (<<https://insights.stackoverflow.com/survey/2019#technology>>). Parte do Processo Judicial Eletrônico – PJe, por exemplo, é desenvolvida em Javascript (por meio do *framework* Angular, mantido pelo Google). Na Incubadora, temos utilizado um *framework* de Javascript chamado Vue, bastante popular na comunidade de desenvolvedores *open source* autônomos. A história do Javascript ilustra, portanto, como pode dar certo uma tecnologia que tinha tudo para dar errado, e como as comunidades em torno de determinada tecnologia podem fazer a diferença muito mais do que a qualidade da tecnologia em si.

A mescla de tecnologias *open source* com tecnologias proprietárias não se torna um problema para o desenvolvimento colaborativo, desde que se adote uma arquitetura adequada, com uso de técnicas de abstração e de acoplamento fraco (*loose coupling*). Em termos leigos, isso significa que a solução deve ser construída em módulos que se encaixem como as peças de um Lego, de modo a possibilitar que as diversas partes sejam desencaixadas e substituídas quando necessário, sem alteração da estrutura principal do projeto. Assim, desde que observada essa arquitetura, o uso do SQL Server, por exemplo, não deve ser impeditivo ou desestímulo à colaboração de terceiros em projetos da Incubadora, pois estes podem “encaixar” o seu próprio sistema de gerenciamento de banco de dados ao conjunto sem qualquer perda de funcionalidade.

Finalmente, convém ressaltar que os critérios de escolha abordados acima devem ser aplicados não somente ao código desenvolvido pela equipe, como a todo o conjunto de elementos do projeto, incluindo suas dependências (componentes, bibliotecas, *frameworks* etc.). Além disso, deve-se evitar a proliferação desnecessária de dependências, pois isso pode tornar o projeto insustentável e reduzir a performance. Também deve-se ter o cuidado de conhecer minimamente as dependências utilizadas, mesmo as de desenvolvimento, pois de outro modo perdem-se dois dos benefícios mais importantes do desenvolvimento colaborativo, que são a transparência e segurança. Há casos em que dependências aparentemente inofensivas trazem verdadeiros “cavalos de Tróia”²⁵.

Propriedade intelectual

Também é importante escolher uma licença adequada para o projeto, que seja compatível com as outras partes criadas por terceiros (componentes, bibliotecas, *frameworks* etc.) e que não afastem potenciais colaboradores desnecessariamente. Por exemplo, se for proibida a utilização do software para fins comerciais, provavelmente será bem mais difícil atrair desenvolvedores profissionais, mesmo *freelancers*, que poderiam imprimir qualidade maior ao projeto. Se o objetivo é atrair também esses desenvolvedores, o melhor é permitir o uso comercial.

Além disso, não se deve complicar demais a licença. Os desenvolvedores colaborativos geralmente preferem trabalhar com licenças padronizadas, conhecidas.

²⁵ Por exemplo, houve casos de distribuição de imagens do Docker com *scripts* maliciosos, que se aproveitavam dos sistemas em que essas imagens eram instaladas para mineração de criptomoedas. <<https://www.bleepingcomputer.com/news/security/17-backdoored-docker-images-removed-from-docker-hub/>>. Acesso em: 7 abr. 2019.

Por essas razões, temos preferido adotar, na Incubadora, sempre que possível, a licença MIT, a mais concisa de todas as licenças padrão de software livre²⁶.

Uso de metodologias ágeis

O desenvolvimento colaborativo de *software* exige, ainda, o uso de metodologias ágeis e de ferramentas compatíveis com essas metodologias. Dificilmente se pode pensar em desenvolver colaborativamente utilizando a metodologia tradicional, em cascata (*waterfall*), salvo se a colaboração for concebida num sentido muito estreito, em que os colaboradores não trabalham para um projeto comunitário, mas para o projeto de alguém. Nesse sentido, pode-se falar em colaboração num projeto do departamento de TI a ser desenvolvido pelo método tradicional, em cascata. Todavia, a colaboração num projeto comunitário, que é o modo preferível de atuação da Incubadora, por ser o mais apto a gerar inovação, não admite outras metodologias senão as ágeis, especialmente no que diz respeito às formas de documentação, que a especificação de requisitos se dá ao longo do processo de desenvolvimento, à medida que os próprios *stakeholders* vão adquirindo consciência de suas necessidades e das potencialidades do projeto.

Além disso, as metodologias ágeis são as únicas compatíveis com a técnica do *minimum viable product* – MVP e dos ciclos rápidos e contínuos de entrega (*continuous delivery*) e de *feedback* que permitem a evolução rápida da solução e evitam o desperdício²⁷.

²⁶ O texto da licença está disponível em: <<https://mit-license.org/>>. Acesso em: 7 deabr. 2019. Seu teor é o seguinte: “Copyright © 2019 <copyright holders> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ‘Software’), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. The software is provided ‘as is’, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software”.

²⁷ Sobre as metodologias ágeis e as vantagens da criação de MVPs, ver o capítulo que trata da Equipe de Gestão de Dados do iJuspLab. As considerações feitas ali sobre a documentação dos projetos também se aplicam integralmente à Incubadora.

É importante também adotar um sistema de versionamento distribuído, como o Git, seguindo-se, sempre que possível, as boas práticas no uso desse sistema²⁸, manter o repositório do projeto em nuvem, mediante a abertura de conta em algum dos portais reconhecidos como padrão de indústria, como, por exemplo, GitHub, GitLab ou Bitbucket²⁹, e adotar ferramentas que permitam a distribuição visual das tarefas em interfaces compartilháveis. Na Incubadora, temos utilizado ferramentas de Kanban em nuvem, especialmente Trello e MeisterTask.

Comunicação eficaz e contínua

Outro elemento importante para o desenvolvimento colaborativo é assegurar canais de comunicação eficazes entre os membros da comunidade de desenvolvedores. Se a comunicação se torna difícil, aos poucos a comunidade se esvazia. Parte desses canais pode ser provido pelas plataformas em nuvem citadas, como o GitLab e o MeisterTask, por exemplo. Nada substitui, no entanto, os encontros presenciais periódicos, pois é nesses encontros que os colaboradores podem se conhecer melhor e trocar experiências de forma direta.

Na Incubadora, temos proporcionado esses encontros mediante reuniões ocasionais de trabalho, para as quais trazemos também os colaboradores residentes em outras cidades. Recentemente começamos a realizar também conferências curtas, que chamamos de MiniConfs, nas quais um dos integrantes da Incubadora ou um expositor externo se encarrega de expor de forma sucinta um assunto de interesse de todos, abrindo-se em seguida uma rodada de debates.

Outro modo de assegurar a comunicação entre os desenvolvedores é promover workshops sobre ferramentas, metodologias e quaisquer outros assuntos de interesse para os projetos. Embora ainda não tenhamos feito workshops na Incubadora, há planos de fazer em breve um workshop sobre ferramentas de versionamento.

²⁸ Atualmente, tem-se adotado como padrão o uso de versionamento semântico (<<https://semver.org/lang/pt-BR/>>. Acesso em: 7 abr. 2019) e a organização do desenvolvimento pelo método denominado *Gitflow* (cf. DRIESSEN, 2010).

²⁹ O versionamento adequado do código-fonte é o que assegura o fluxo de entrega e feedback contínuos, essenciais para a adoção da estratégia do MVP e para a coordenação das atividades desenvolvidas pelos colaboradores: “No centro deste processo está o versionamento de código fonte e um serviço de integração contínua, responsável por executar todos os testes unitários e de integração e gerar um pacote de software pronto para instalação a cada modificação introduzida (commit) pelos desenvolvedores”. RIES, 2011, p. 3715-3718.

Por fim, a comunicação pode também ser estimulada pela criação de grupos e subgrupos de pessoas interessadas em assuntos específicos ou que trabalhem nos mesmos projetos. Esses grupos podem contar com a participação de pessoas de fora, incluindo, por exemplo, usuários comuns e até técnicos do departamento de TI.

Nesse campo, a Incubadora tem adotado como exemplo o que vem sendo feito na Equipe de Gestão de Dados do iJuspLab. Além de ser constituída por um grupo de pessoas provenientes de vários departamentos da administração, reunidas em torno do interesse comum de trabalhar com planilhas, bancos de dados e ferramentas de BI, a equipe é também subdividida em times de projetos, subgrupos formados por membros e *stakeholders* que trabalham juntos em torno de um problema de interesse comum. Os integrantes desses times se reúnem eventualmente com a coordenação para receber orientações e traçar estratégias de desenvolvimento. Essa dinâmica de grupos e subgrupos ajuda a enriquecer a comunicação e fortalecer o entrosamento³⁰. Na Incubadora é mais difícil implementar algo semelhante, tendo em vista o número bem menor de pessoas. Todavia, isso pode ser compensado pela participação em atividades de outros grupos, como os da própria Equipe de Gestão de Dados ou os que se formam no laboratório para atividades de *design thinking*, por exemplo.

Encaminhamento de demandas

Uma vez que a função principal da Incubadora é a prototipagem de *soluções inovadoras*, ela deve ter muito cuidado para não assumir tarefas típicas do departamento de TI, como, por exemplo, a criação e manutenção de aplicações de uso comum, que não agregam valor aos fluxos de trabalho, mas apenas preservam o que existe.

³⁰ Um caso interessante que ilustra essa ideia é o modelo organizacional do Spotify. Como se sabe, o Spotify é uma empresa de tecnologia responsável pelo famoso aplicativo de *streaming* de músicas. Para facilitar a aplicação de metodologias ágeis de trabalho, os funcionários da empresa dividem-se em *squads*, *tribes*, *chapters* e *guilds*. Os *squads* são pequenas equipes de desenvolvimento com certo grau de autonomia para escolher métodos e ferramentas de trabalho. Os *squads* que atuam no desenvolvimento de soluções relacionadas são organizados em *tribes*. Indivíduos da mesma “tribo”, mas de diferentes *squads*, podem formar *chapters*, que são grupos de indivíduos que trabalham com as mesmas ferramentas ou que têm as mesmas competências, e que se reúnem para fins de mentoria e troca de informações e conhecimento. Por fim, *guilds* são grupos de indivíduos de “tribos” diferentes que compartilham interesses comuns em determinados assuntos. SPOTIFY, 2014.

Para evitar que isso aconteça, a Incubadora deve exercer de modo adequado o seu papel de intermediária entre os usuários e o departamento de TI, encaminhando àquele departamento tudo o que não tiver relação clara com a inovação, evitando assumir funções que não lhe são próprias e que, por tomarem tempo, poderiam impedi-la de cumprir sua tarefa principal.

É preciso cuidar, no entanto, para que essa diretriz geral não seja aplicada de modo muito rígido. Pode ocorrer por vezes que a solução solicitada seja realmente bem simples e que a perspectiva de atendimento pelo departamento de TI seja tão remota que a atuação da Incubadora se justifique ao menos como forma de demonstrar boa vontade perante os usuários finais.

E também pode ocorrer o contrário, ou seja, que uma determinada demanda, apesar de estar relacionada à atividade de inovação, não possa ser assumida pela Incubadora, seja por razões de estratégia, seja por razões de viabilidade ou de competência. Por exemplo, na Justiça Federal as demandas relacionadas ao Sistema Processual Eletrônico (PJe) são administradas por um comitê gestor local e levadas posteriormente ao Conselho Nacional de Justiça, que coordena de modo centralizado o desenvolvimento colaborativo do sistema. Assim, as demandas desse tipo não poderiam ser assumidas pela Incubadora sem que antes fosse ouvido o comitê gestor local. Por outro lado, mesmo no que tange a essas demandas, a Incubadora poderia eventualmente ajudar o usuário a explicar melhor e fundamentar tecnicamente a sua ideia. Poderia, por exemplo, ajudá-lo a construir um protótipo de baixa fidelidade para ilustrar o que deseja propor antes de encaminhar a demanda ao comitê gestor local do PJe.

Em suma, a Incubadora deve exercer suas atribuições com razoabilidade e bom senso. Deve triar cuidadosamente as demandas que lhe chegam às mãos, separando as verdadeiramente inovadoras (ou seja, as que agregam valor aos fluxos de trabalho) das não-inovadoras (as que apenas mantêm os serviços atuais em funcionamento) e procurar assumir para si somente as do primeiro tipo, sem, contudo, fechar completamente as portas para os casos excepcionais.

Capacitação

De modo semelhante ao que foi dito no capítulo anterior em relação à Equipe de Gestão de Dados do iJuspLab, a estratégia de capacitação deve levar em conta os objetivos que se pretende atingir, os quais, no caso da Incubadora, são os seguintes: (i) prospecção de novas tecnologias e ferramentas; (ii) formação de repertório

conceitual básico; e (iii) treinamento em tecnologias específicas, efetivamente utilizadas em seus projetos.

Os cursos de prospecção podem envolver qualquer tipo de assunto da área de tecnologia, mas devem ser breves, de modo a nunca concorrer com a tarefa mais importante da Incubadora, que é conduzir seus projetos de inovação. Pela mesma razão, convém que também sejam curtos os cursos de capacitação da terceira espécie, voltados à formação em tecnologias específicas. Nesse caso, deve-se procurar limitar o objeto dos cursos a tecnologias utilizadas ou que se pretenda utilizar em breve nos projetos da Incubadora. Convém, ainda, que esses cursos tenham caráter eminentemente prático e que, de preferência, envolvam projetos reais da Incubadora.

Tivemos, nessa seara, uma experiência malsucedida em que contratamos um curso muito longo (260 horas) para treinamento em tecnologias muito específicas, que pretendíamos utilizar, mas acabamos não utilizando completamente. Apesar de não ter sido de todo inútil, esse treinamento produziu alguns inconvenientes graves, que convém evitar no futuro: (i) as aulas tomaram muito tempo e acabaram suprimindo horas de trabalho que poderiam ter sido empregadas nos projetos da Incubadora; (ii) os projetos realizados no curso são muito diferentes daqueles atualmente conduzidos pela Incubadora e, pior ainda, não estavam completamente alinhados com as estratégias de inovação da Justiça Federal de São Paulo; e (iii) as tecnologias estudadas estão sujeitas a obsolescência, o que torna bastante desproporcional o número de horas dedicadas ao curso em relação aos benefícios obtidos.

Por fim, no que se refere aos cursos destinados à formação conceitual básica, tal como mencionado no capítulo anterior, estes sim podem e devem ser mais longos. Convém, no entanto, que as aulas sejam espaçadas, de modo a equilibrar o tempo empregado em capacitação com o tempo empregado no desenvolvimento de projetos. Além disso, a distribuição das aulas num período maior contribui para a assimilação dos conceitos e seu emprego em projetos reais. Aliás, seria até mesmo desejável que cursos desse tipo seguissem um programa de formação contínuo, permanente, em que fossem previstos ciclos sucessivos de capacitação no formato de “trilhas de formação”.

PROJETOS EM ANDAMENTO, VISÃO PARA O FUTURO, CONCLUSÃO

Quando começamos a pensar numa Incubadora de Soluções Tecnológicas, tínhamos na Justiça Federal de São Paulo um grupo de desenvolvedores motivado

e disposto a ingressar conosco nesse projeto. O grupo era experiente. Tinha passado pelo desafio de migrar diversos sistemas críticos de uma tecnologia antiga para uma outra, completamente nova para eles, que teriam de aprender do “zero”. Alguns decidiram ingressar na universidade para melhor enfrentar o desafio. Tinha também um portfólio respeitável de projetos bem-sucedidos, de altíssimo valor para a administração, como o Sistema de Acompanhamento e Controle Orçamentário (SIACOR) e o Sistema de Diárias e Passagens. Além disso, o grupo não integrava a área de TI, o que nos permitiria realocá-lo mais facilmente para o laboratório de inovação.

Esse conjunto de características permitiu que avançássemos bastante desde que a Incubadora foi formalmente criada em agosto de 2018. Nesse breve período de aproximadamente oito meses, lançamos a versão 2.0 do SIACOR, estendendo o sistema para uso pela Justiça Federal do Mato Grosso do Sul, iniciamos a modernização do Sistema de Diárias e Passagens e lançamos a versão alfa de um Sistema de Administração Predial e Plano de Obras, que trará grandes benefícios também para a gestão de dados, reunindo informações hoje dispersas em diversas bases de dados e planilhas.

Além desses sistemas, que são protótipos de altíssima fidelidade que o nosso departamento de TI nos autorizou a colocar diretamente em produção, a Incubadora também desenvolveu outras soluções inovadoras importantes nesse curto período, dentre as quais:

- a) A extensão “Minha Intranet” para Typo3, que permite aos usuários internos da Justiça Federal de São Paulo customizar sua visualização da Intranet, selecionando apenas os serviços que lhes interessam.
- b) O protótipo em Google Sheets e Google Apps Script demonstrando a viabilidade da implementação do algoritmo de emparelhamento estável para a movimentação de servidores mediante triangulação.

A criação da Incubadora permitiu, ainda, que os protótipos desenvolvidos em Google Sheets e Google Apps Script para a melhoria das ferramentas de cálculo de benefícios previdenciários fossem trazidos para o laboratório de inovação e que se pudesse, em razão disso, iniciar um projeto conjunto com o departamento de TI para integrar essas ferramentas ao parque tecnológico da Justiça Federal da 3ª Região.

Para o futuro, há três grandes linhas de ação que se abrem para a Incubadora:

- 1) Esperamos que os sistemas criados pela Incubadora possam ser aos poucos compartilhados com outras Seções Judiciárias que se interessarem em desenvolvê-los conosco, colaborativamente.
- 2) Pretendemos tornar cada vez mais clara a separação entre as atividades de prototipagem da Incubadora e as atividades de nosso departamento de TI. Para tanto, estamos nos preparando para implantar, juntamente com a SETI, uma esteira de DevOps, por meio da qual não precisemos mais lidar diretamente com infraestrutura³¹. Também gostaríamos de utilizar mais serviços de infraestrutura em nuvem. Essa é uma estratégia que foi seguida pela Incubadora nos protótipos desenvolvidos no Google Drive (em Google Sheets e Google Apps Script) e que pode ser expandida pela contratação de serviços mais sofisticados, como o provisionamento de ambientes inteiros em nuvem, por exemplo³².
- 3) Por fim, em conjunto com a Equipe de Gestão de Dados, pretendemos aos poucos reunir os meios necessários (ferramentas, infraestrutura, capacitação) para começar a testar possíveis aplicações da inteligência artificial aos serviços administrativos.

³¹ DevOps é um termo que expressa mais uma cultura que um conjunto de ferramentas. Em resumo, trata-se da confluência de diversas técnicas e metodologias extraídas do movimento *lean*, do manifesto ágil, do movimento *velocity* (aplicação de princípios ágeis na infraestrutura), do movimento entrega contínua (*continuous delivery*) e do Toyota Kata (movimento de melhoria contínua também inspirado no Sistema Toyota de Produção) para resolver o “conflito crônico básico” entre as áreas de desenvolvimento e de operações das organizações de TI (cf. KIM et al., 2018, xxiii e 4-6). A organização e o funcionamento da Incubadora incorpora diversos elementos dessa cultura, mas sem as ferramentas adequadas alguns aspectos importantes dessa cultura ficam comprometidos, como a ideia de IAC (*infrastructure as code*), por exemplo. Agora, portanto, é necessário avançar para implantar ferramentas que permitam trazer para os fluxos de trabalho da Incubadora outros elementos de DevOps incompatíveis com as ferramentas atuais. Esse plano está em vias de se concretizar. Recentemente a SETI implantou uma infraestrutura de *containers* (Docker-Kubernetes-Openshift), que a Incubadora pretende começar a utilizar em breve, juntamente com ferramentas complementares de provisionamento de ambientes, gerenciamento de configurações e automação (provavelmente Ansible, Puppet e Jenkins).

³² As vantagens e os riscos da contratação de serviços em nuvem pelo poder público foram analisados em profundidade pelo Tribunal de Contas da União no Acórdão n. 1.739/2015, citado no capítulo anterior.

Equipes pequenas não podem prometer muito. Hoje a Incubadora conta com nove integrantes, dos quais dois lidam somente com gestão de dados, dois são contadores, um está licenciado e outro, diretor e líder do grupo, está constantemente ocupado em reuniões e outras atividades de gestão. Assim, dos nove membros atuais, apenas três estão diretamente envolvidos com o desenvolvimento de soluções tecnológicas. Difícil saber, portanto, o quanto poderemos avançar em cada uma das frentes mencionadas. Todavia, pelo tanto que fizemos e pelo valor do nosso grupo, há razões para esperar que ainda faremos muito.

Parte da nossa capacidade de avançar em tão pouco tempo se deve certamente ao fato de termos começado com um grupo formado e capacitado. Nem todos os laboratórios de inovação terão a mesma sorte. Muitos provavelmente precisarão formar sua incubadora do “zero”, assim como fizemos com a nossa Equipe de Gestão de Dados. Mas o fato de termos sido bem-sucedidos na formação dessa equipe demonstra que esse caminho também é possível e pode produzir bons frutos. Isso nos leva à conclusão de que o importante não é o elemento acidental – a sorte de termos podido contar com um grupo formado – mas sim todo o resto: focar em prototipagem, adotar metodologias ágeis e enxutas, promover o desenvolvimento colaborativo.

BIBLIOGRAFIA

ALLSPAW, John; HAMMOND, Paul. 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr (vídeo da conferência ministrada no evento Velocity – Web Performance and Operations Conference, publicado por O’Reilly Media em 25 jun. 2009). Disponível em: <<https://www.youtube.com/watch?v=LdOe18KhtT4>>. Acesso em: 5 mar. 2019.

APRENDA sobre desenvolvimento ágil de software. Disponível em: <<http://www.desenvolvimentoagil.com.br/>>. Acesso em: 5 mar. 2019.

CIMPANU, Catalin. *17 Backdoored docker images removed from docker hub*. 13 jun. 2018. Disponível em: <<https://www.bleepingcomputer.com/news/security/17-backdoored-docker-images-removed-from-docker-hub/>>. Acesso em: 7 abr. 2019.

CROCKFORD, Douglas. *JavaScript: the good parts: the good parts*. Sebastopol: O’Reilly, 2008 (e-book Kindle).

DRIESSEN, Vincent. A Successful Git Branching Model. 5 jan. 2010. Disponível em: <<https://nvie.com/posts/a-successful-git-branching-model/>>. Acesso em: 7 abr. 2019.

EICH, Brendan. *The A-Z of programming languages: javascript*. Entrevista disponível em: <https://www.computerworld.com.au/article/print/255293/a-z_programming_languages_javascript/>. Acesso em: 7 abr. 2019.

FUJITSU. *Spotting shadow IT (white paper)*. 2016. Disponível em: <https://www.fujitsu.com/us/Images/Hybrid%20IT%20WP7_Shadow%20IT.pdf>. Acesso em: 5 mar. 2019.

GARTNER. *Citizen developer (verbete)*. Disponível em: <<https://www.gartner.com/it-glossary/citizen-developer/>>. Acesso em: 5 mar. 2019.

HUMBLE, Jez; FARLEY, David. *Entrega contínua*. Como entregar software de forma rápida e confiável. Trad. Marco Aurélio Valtas Cunha, Ronaldo Melo Ferraz. Porto Alegre: Bookman, 2014 (e-book Kindle).

ISAACSON, Walter. *Os inovadores*. Uma biografia da revolução digital. São Paulo: Cia. das Letras, 2014.

KIM, Gene; HUMBLE, Jez; DEBOIS, Patrick; WILLIS, John. *Manual de devops*. Como obter agilidade, confiabilidade e segurança em organizações tecnológicas. Trad. João Tortello. São Paulo: Alta Books, 2018.

MANIFESTO *for agile software development*. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 5 mar. 2019.

RAYMOND, Eric. S. *The cathedral and the bazaar*. Musings on Linux and Open Source by an Accidental Revolutionary. Sebastopol: O'Reilly, 2001 (revised & expanded) (e-book Kindle).

RIES, Eric. *A startup enxuta*. Como os empreendedores atuais utilizam a inovação contínua para criar empresas extremamente bem-sucedidas. Trad. Texto Editores. São Paulo: Leya, 2011.

ROGERS, David L. *Transformação digital*. Repensando o seu negócio para a era digital. São Paulo: Autêntica Business, 2017 (e-book Kindle).

SEIBEL, Peter (ed.). *Coders at work*. Reflections on the craft of programming. Interviews with some of the top programmers of our times. New York: Apress, 2009 (e-book Kindle).

SILIC, M.; Back, A. (2014). *Shadow IT*. A view from behind the curtain. Computers & Security, 45, 274-283. Disponível em: <https://www.researchgate.net/publication/263284725_Shadow_IT_-_A_view_from_behind_the_curtain>. Acesso em: 5 mar. 2019.

SOMMERFELD, Rafael. *Como sobreviver ao paradoxo da shadow IT x TI convencional*. 15 maio 2015. Disponível em: <<https://computerworld.com.br/2015/05/15/como-sobreviver-ao-paradoxo-da-shadow-it-x-ti-convencional/>>. Acesso em: 5 mar. 2019.

SPOTIFY *Engineering Culture, parts 1 and 2*. 30 jan. 2014 e 12 maio 2014. Disponível em: <<https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>> e <<https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/>>. Acesso em: 7 abr. 2019.

TAURION, Cezar. *Por que a TI resiste à inovação?* 5 out. 2015. Disponível em: <<https://cio.com.br/por-que-a-ti-resiste-a-inovacao/>>. Acesso em: 31 mar. 2019.

TWENTYMAN, Jessyca. *CIOs start to view “Shadow IT” as a catalyst for innovation*. Disponível em: <<https://www.i-cio.com/innovation/consumerization/item/how-cios-start-to-view-shadow-it-as-a-catalyst-for-innovation>>. Acesso em: 5 mar. 2019.

YATSENKO, Maria. *Risks of shadow IT and how to mitigate them*. 23 jun. 2018. Disponível em: <<https://www.apriorit.com/dev-blog/542-shadow-it-risks>>. Acesso em: 5 mar. 2019.

ANEXO I

PROPOSTA DE REGRAMENTO PARA O DESENVOLVIMENTO COLABORATIVO FORMULADA PELA JUSTIÇA FEDERAL DE SÃO PAULO

MÓDULO 1: GENERALIDADES

I – REFERÊNCIAS:

1. Resolução CNJ n. 90, de 29 de setembro de 2009, alterada pela Resolução CNJ n. 136, de 13 de julho de 2011.
2. Resolução CJF n. 148, de 26 de maio de 2011.
3. Portaria CJF n. 23, de 19 de janeiro de 2016.
4. Resolução PRES n. 293, de 22 de maio de 2012, e alterações.
5. Resolução PRES n. 424, de 9 de junho de 2015.
6. Resolução PRES n. 167, de 22 de janeiro de 2018.

II – FINALIDADE:

Estabelecer regras e procedimentos para atividades de prototipagem e desenvolvimento colaborativo de projetos de software e de gestão de dados no âmbito da Justiça Federal da 3ª Região, mediante coordenação pela Comissão de Informática do TRF3, pela DFOR/MS, pela DFOR/SP, pela UDEM e pela SETI.

III – ABREVIATURAS:

1. CJF: Conselho da Justiça Federal.
2. CNJ: Conselho Nacional de Justiça.
3. Comissão de Informática: Comissão Permanente de Informática do TRF3.
4. Comitê Gestor do PJe: Comitê Gestor Regional do PJe do TRF3.
5. DFOR/SP: Diretoria do Foro da Seção Judiciária de São Paulo.
6. DFOR/MS: Diretoria do Foro da Seção Judiciária de Mato Grosso do Sul.
7. IN: Instrução Normativa.
8. PRES: Presidência do Tribunal Regional Federal da 3ª Região.
9. SEI: Sistema Eletrônico de Informações.
10. SETI: Secretaria de Tecnologia da Informação do TRF3.
11. SJMS: Seção Judiciária do Mato Grosso do Sul.

12. SJSP: Seção Judiciária de São Paulo.
13. TCU: Tribunal de Contas da União.
14. TI: Tecnologia da Informação.
15. TRF3: Tribunal Regional Federal da 3ª Região.
16. UDEM: Subsecretaria de Desenvolvimento e Manutenção de Sistemas do TRF3.

IV – CONCEITOS:

1. Armazém de dados ou *data warehouse*: banco de dados criado no contexto das atividades de gestão de dados com a finalidade de reunir, integrar e disponibilizar informações dispersas, estruturadas ou não, oriundas de fontes distintas e frequentemente heterogêneas, a fim de tornar possível a análise de dados e a produção de relatórios gerenciais.
2. Artefato: qualquer documento criado no processo de desenvolvimento de software, incluindo itens de documentação, formulários, programas etc.
3. Centros Locais de Inteligência: os Centros Locais de Inteligência criados no âmbito da Justiça Federal da 3ª Região por determinação da Resolução n. 369/2017, do Conselho da Justiça Federal.
4. Código aberto: mesmo sentido de “*open source*” tal como definido pela Open Source Initiative no site <https://opensource.org/docs/definition.php>, consultado nesta data.
5. Codificação: fase do desenvolvimento que tem como objetivo a programação dos componentes do software na linguagem definida e aprovada no projeto.
6. Colaborador externo: qualquer pessoa física, servidora pública ou não, pertencente ou não aos quadros da Justiça Federal, não qualificada como desenvolvedor cadastrado, mas admitida a colaborar num determinado projeto.
7. Comunidade do projeto: o conjunto dos colaboradores externos admitidos pelo coordenador do projeto para auxiliar nas atividades de desenvolvimento.
8. Coordenador do projeto: desenvolvedor cadastrado ou grupo de desenvolvedores cadastrados responsável por coordenar um projeto de software e por representar a equipe de desenvolvimento e a comunidade do projeto perante os órgãos internos da Justiça Federal da 3ª Região.
9. Dependências: trechos de código, bibliotecas, módulos e *frameworks* de terceiros agregados ao código-fonte criado pelo desenvolvedor colaborativo com vistas a aproveitar funcionalidades existentes e disponíveis.
10. Desenho do projeto: fase do desenvolvimento em que é definida a arquitetura do software, ou seja, seus componentes e a infraestrutura necessária ao seu funcionamento.
11. Desenvolvedor cadastrado: qualquer magistrado ou servidor da Justiça Federal da 3ª Região, incluindo servidores dos quadros da SETI, que esteja inscrito no cadastro de desenvolvedores colaborativos.

•• Inovação no Judiciário

12. Desenvolvimento: processo de criação e manutenção de software, implantado ou não, considerado em todas as suas fases, incluindo o desenho do projeto, a especificação de requisitos, a codificação, a aplicação de testes, a homologação, a implantação e o treinamento.

13. Desenvolvimento ágil: conjunto de valores e princípios para o desenvolvimento de software expressos no Manifesto para Desenvolvimento Ágil de Software de 2001, reproduzido em <http://agilemanifesto.org/iso/ptbr/manifesto.html>, consultado nesta data.

14. Desenvolvimento colaborativo: desenvolvimento de software realizado por desenvolvedor cadastrado ou grupo de desenvolvedores cadastrados em conjunto com um ou mais *stakeholders* e colaboradores externos, em regime de colaboração com a SETI, mediante prévia autorização da Administração.

15. Equipe de desenvolvimento: conjunto de um ou mais desenvolvedores cadastrados e um ou mais *stakeholders* responsáveis pelo desenvolvimento de um projeto de software.

16. Especificação de requisitos: fase do desenvolvimento que tem como objetivo definir as regras de negócio junto aos *stakeholders*, integrantes ou não da equipe de desenvolvimento.

17. Gestão de dados: conjunto de atividades realizadas com o auxílio dos instrumentos teóricos e técnicos da ciência de dados (*data science*) para a coleta, integração e análise de dados, estruturados ou não, provenientes ou não da mesma fonte, com o objetivo de (i) subsidiar a tomada de decisões gerenciais, mediante análise situacional e identificação de causas e tendências; (ii) automatizar o cálculo de métricas de desempenho e outros indicadores numéricos importantes para a tomada de decisões gerenciais; e (iii) automatizar e aprimorar fluxos de trabalho.

18. Homologação: fase do desenvolvimento que tem como objetivo assegurar que a solução de software criada atende aos requisitos especificados pelos *stakeholders* durante o desenvolvimento.

19. Implantação ou liberação da aplicação: fase do desenvolvimento que tem como objetivo colocar a solução de software (ou parte dela) em operação, ou seja, torná-la disponível aos usuários finais em ambiente de produção.

20. Implantação assumida: modelo de implantação de solução de software em que a SETI assume a responsabilidade pela distribuição, instalação, atualização e suporte da solução no âmbito da Justiça Federal da 3ª Região, podendo, a partir de então, a seu critério, criar um novo ramo (*branch*) do código-fonte para assumir também o controle do desenvolvimento, sem prejuízo da continuidade do desenvolvimento do ramo principal (*master branch*) pela equipe de desenvolvimento e pela comunidade do projeto.

21. Implantação autorizada: modelo de implantação de software em que a equipe de desenvolvimento assume a responsabilidade pela distribuição, instalação, atualização

e suporte da solução de software diretamente aos usuários que manifestem interesse em utilizá-la, segundo termos de uso e de privacidade previamente aprovados pela SETI caso a caso, com teor semelhante ao do modelo do Anexo IV desta IN.

22. Infraestrutura da Justiça Federal: *data centers* públicos, privados e/ou híbridos, próprios ou terceirizados, mantidos ou contratados por órgãos do Poder Judiciário e disponibilizados oficialmente à Justiça Federal da 3ª Região, incluindo, por exemplo, os *data centers* próprios mantidos pela SETI e os *data centers* mantidos e disponibilizados a todas a Justiça Federal pelo Conselho Nacional de Justiça e pelo Conselho da Justiça Federal.

23. Melhoria simples: qualquer alteração no código-fonte que não caracterize melhoria substancial, como, por exemplo, (i) a mera correção de falhas; (ii) as melhorias de performance, organização do código ou interface do usuário; e (iii) os *upgrades* ou *downgrades* das dependências e/ou das tecnologias utilizadas, com o intuito de corrigir ou promover qualquer das alterações mencionadas nos itens anteriores.

24. Melhoria substancial: qualquer alteração no código-fonte que implique mudança em regra de negócio, inclusão ou supressão de dependências ou substituição das tecnologias aprovadas, no todo ou em parte.

25. Modelo de contrato: modelo de desenvolvimento colaborativo em que um desenvolvedor cadastrado ou *stakeholder* propõe e/ou aceita prestar auxílio à SETI em projeto de iniciativa daquele órgão, segundo condições e especificações pré-determinadas.

26. Modelo de software aberto: modelo de desenvolvimento colaborativo em que o desenvolvimento do projeto é coordenado por uma equipe de desenvolvimento, que mantém o código-fonte em portal de repositórios, público ou privado, acessível pela SETI.

27. Órgãos de inovação tecnológica: os órgãos já previstos ou que venham a ser previstos na estrutura administrativa do TRF3, da DFOR/MS e da DFOR/SP com a atribuição de incubar (pesquisar, prospectar, estudar e/ou desenvolver) soluções inovadoras na área de tecnologia, sem ingerência nos serviços de TI (salvo pelas atividades aqui previstas ou previamente autorizadas pela SETI), seguindo metodologias de desenvolvimento ágil, *design thinking*, *lean startup*, empatia, foco no usuário e outras, com o intuito de proporcionar o rápido desenvolvimento de propostas de soluções de software ou de gestão de dados, prontas ou não para implantação, a serem levadas às áreas técnicas.

28. Patrocínio: reconhecimento, por um órgão autorizado da Justiça Federal da 3ª Região, sem qualquer conotação financeira ou de apoio material, de que o projeto é de interesse dos serviços judiciais e/ou administrativos e está alinhado ao plano estratégico da instituição.

29. Patrocinador: órgão da Justiça Federal da 3ª Região com atribuição para receber e apreciar pedidos de patrocínio. São patrocinadores: (i) a Comissão de Informática

•• Inovação no Judiciário

no caso de projetos de interesse da área administrativa propostos por desenvolvedor cadastrado pertencente ao quadro do TRF3; (ii) a DFOR/MS no caso de projetos de interesse da área administrativa ou do Centro Local de Inteligência da SJMS, propostos por desenvolvedor cadastrado pertencente ao quadro da SJMS; (iii) a DFOR/SP no caso de projetos de interesse da área administrativa ou do Centro Local de Inteligência da SJSP, propostos por desenvolvedor cadastrado pertencente ao quadro da SJSP; e (iv) a Comissão de Informática e o Comitê Gestor do PJe no caso de projetos de interesse das áreas judiciais do TRF3, da SJMS ou da SJSP, cabendo ao Comitê Gestor do PJe tratar dos projetos envolvendo o PJe e à Comissão de Informática tratar dos demais projetos.

30. PJe: projeto de software denominado “Processo Judicial Eletrônico (PJe)”, desenvolvido pelo CNJ em parceria com os tribunais, com a participação da Ordem dos Advogados do Brasil (OAB), para a automação dos processos judiciais, conforme descrito em <http://www.pje.jus.br/>, consultado nesta data.

31. Portal de repositórios: qualquer implementação de um sistema de versionamento em nuvem pública ou privada, como, por exemplo, o GitHub (<https://github.com/>), o Portal do Software Público Brasileiro (<https://softwarepublico.gov.br/social/>) e o GitLab (<https://about.gitlab.com/>).

32. Projeto de software: iniciativa de desenvolvimento de software com o objetivo de oferecer uma nova solução de software ou de aprimorar soluções de software já existentes no âmbito da Justiça Federal da 3ª Região.

33. Protótipo: conjunto de rascunhos, diagramas, blocos de código, partes de sistemas de software, com ou sem dependências, conjuntos de documentos, bancos de dados e quaisquer outros artefatos, criados em caráter experimental, com o intuito de testar o emprego de diferentes tecnologias e abordagens para solução de problemas específicos, para a melhoria de fluxos de trabalho ou para o aperfeiçoamento de softwares já existentes, no intuito de eventualmente propor um novo projeto de software de interesse da Justiça Federal. O protótipo pode ser de alta, média ou baixa fidelidade, conforme seja maior ou menor a fidelidade com que reproduza o funcionamento real da solução a ser desenvolvida.

34. Serviços de terceiros: aplicativos e *application programming interfaces* (APIs) disponibilizados por terceiros, gratuita ou onerosamente, por acesso remoto, para obtenção, armazenamento e/ou processamento de dados, tais como *webservices*, bancos de dados em tempo real, conversores, *linters*, *parsers* etc.

35. Sistema de versionamento: qualquer sistema de controle de versões de código-fonte reconhecido como padrão de indústria, como Git e SVN. Os termos técnicos “branch”, “master branch”, “repositório”, “patch” etc., quando aqui utilizados, são entendidos segundo as acepções que eles têm no bojo de um sistema de versionamento.

36. Solução de software: código de programação criado para resolver problemas de gestão ou conferir maior eficiência a fluxos de trabalho.

37. *Stakeholders*: usuários efetivos ou potenciais da solução de software e/ou gestores direta ou indiretamente afetados pela solução de software.

MÓDULO 2: DESENVOLVIMENTO COLABORATIVO

I – CADASTRO DE DESENVOLVEDORES COLABORATIVOS

1. A SETI manterá o cadastro de desenvolvedores colaborativos da Justiça Federal da 3ª Região.

1.1. Qualquer servidor ou magistrado que atenda às condições previstas nesta IN poderá ser cadastrado como desenvolvedor colaborativo.

1.2. É automático o cadastro dos servidores e magistrados integrantes dos órgãos de inovação tecnológica da Justiça Federal da 3ª Região, conforme lista informada à SETI pela Comissão de Informática, pela DFOR/MS ou pela DFOR/SP, conforme o caso.

2. Para inclusão no cadastro, os interessados não sujeitos a inclusão automática preencherão formulário próprio, disponibilizado pela SETI na Intranet do TRF3, e deverão:

a) comprovar conhecimento técnico, em nível intermediário ou avançado, em programação de computadores e princípios de desenvolvimento de software, mediante apresentação de código-fonte parcial ou integral de projeto do qual já tenham participado.

b) prestar declaração escrita de que (i) se comprometem a utilizar somente as tecnologias e ferramentas de desenvolvimento autorizadas pela Justiça Federal da 3ª Região; e (ii) conhecem e aceitam observar as normas desta IN e as demais normas de segurança de TI e de desenvolvimento de software em vigor na Justiça Federal.

c) listar e justificar as tecnologias e ferramentas de desenvolvimento que pretendem utilizar e os recursos de infraestrutura de que irão precisar para atividades previstas nesta IN.

2.1. Na página de cadastro, a SETI publicará e manterá atualizados os links para o texto desta IN e de todas as normas de segurança de TI e de desenvolvimento de software em vigor na Justiça Federal aplicáveis às atividades aqui previstas.

3. A SETI terá 10 (dez) dias para aprovar ou rejeitar o pedido de cadastro.

3.1. A aprovação poderá ser parcial, se as ferramentas e a infraestrutura solicitadas forem apenas parcialmente deferidas ou se for decidido fornecer ferramentas e/ou infraestrutura equivalentes, mas não iguais às solicitadas.

3.2. A rejeição ou a aprovação parcial do pedido deverão ser fundamentadas.

3.3. A SETI poderá disponibilizar ferramentas e infraestrutura adicionais, não solicitadas pelo interessado, inclusive determinando seu uso obrigatório, com a

•• Inovação no Judiciário

finalidade de facilitar o controle das atividades exercidas e/ou facilitar a colaboração entre desenvolvedores cadastrados.

3.4. Havendo necessidade de uso de novas ferramentas ou de outros recursos de infraestrutura ainda não autorizados, cabe ao desenvolvedor cadastrado solicitar o aditamento de seu cadastro, a fim de submetê-lo a nova análise.

4. Aprovado o cadastro, total ou parcialmente, a SETI fornecerá ao desenvolvedor cadastrado a infraestrutura deferida, com a identificação da estação ou das estações de trabalho, e concederá privilégios de acesso adequados, em caráter temporário ou permanente, para viabilizar a instalação das tecnologias e ferramentas autorizadas.

4.1. Ao desenvolvedor cadastrado cabe providenciar, por sua própria conta, a instalação de todas as tecnologias e ferramentas autorizadas e cuidar da respectiva manutenção, incluindo posteriores atualizações.

5. São obrigações dos desenvolvedores cadastrados, nas atividades de prototipagem e desenvolvimento colaborativo:

a) adotar em seu código-fonte licenças compatíveis com os modelos aprovados por esta IN ou pela SETI.

b) somente adotar dependências de código aberto ou que, já estando licenciadas à Justiça Federal, tenham seu uso previamente autorizado pela SETI.

c) somente adotar dependências compatíveis com a licença do projeto em que forem utilizadas.

d) expressamente ceder à Justiça Federal o código-fonte de seus projetos, sem qualquer ônus para a Administração Pública, com todas as liberdades inerentes ao código aberto, caso tal condição já não esteja contemplada pela licença adotada.

e) não utilizar as ferramentas e a infraestrutura disponibilizadas ou autorizadas pela SETI senão para as atividades previstas nesta IN.

f) não permitir que quaisquer terceiros utilizem as ferramentas e a infraestrutura disponibilizadas ou autorizadas pela SETI.

g) não inserir no código-fonte compartilhado ou tornar públicas por qualquer outro modo informações sigilosas ou de conhecimento restrito, como, por exemplo, dados sujeitos a sigilo profissional, senhas, *tokens*, *api keys*, *client secrets* etc.

h) salvo se as atividades previstas fizerem parte de suas atribuições funcionais, realizar as atividades de prototipagem, de aprendizagem, de prospecção e de desenvolvimento colaborativo em espírito de total doação e liberalidade, sem qualquer ônus para Administração e sem prejuízo às atribuições inerentes a seu cargo, ficando sob sua exclusiva responsabilidade obter, perante o superior hierárquico, eventuais autorizações que se façam necessárias para participação em reuniões, encontros ou apresentações relacionados às atividades de que trata esta IN.

i) somente utilizar serviços de terceiros gratuitos ou já contratados pela Justiça Federal, nesse último caso mediante prévia e expressa autorização da SETI.

j) não utilizar serviços de terceiros que impliquem armazenamento de dados fora da infraestrutura da Justiça Federal, salvo nos seguintes casos: (i) quando se tratar de dados públicos; (ii) quando os dados pertencerem ao usuário e houver concordância deste com os termos de uso e a política de privacidade do serviço; ou (iii) quando houver prévia autorização da SETI; e

k) adotar as boas práticas que vierem a ser recomendadas pela SETI para segurança da infraestrutura da Justiça Federal e facilitar as atividades de colaboração e de manutenção das soluções de software.

6. O desenvolvedor cadastrado que descumprir qualquer das obrigações aqui previstas será advertido pela SETI por escrito. Se permanecer em situação irregular ou se for reincidente na mesma conduta, poderá ser excluído do cadastro de desenvolvedores colaborativos, após ouvido o órgão patrocinador a que estiver vinculado.

II – PESQUISA E PROTOTIPAGEM DE SOLUÇÕES DE SOFTWARE

1. Todo desenvolvedor cadastrado tem permissão para utilizar as tecnologias, as ferramentas e a infraestrutura disponibilizadas ou autorizadas pela SETI para:

a) pesquisar, prospectar, experimentar e estudar as tecnologias em uso na Justiça Federal ou que possam ser úteis para a Justiça Federal no futuro.

b) prototipar soluções de software que possam vir a ajudar na melhoria de suas próprias tarefas profissionais, das tarefas realizadas por outras áreas ou dos trabalhos da Justiça Federal como um todo.

c) obter treinamento adequado nas tecnologias utilizadas no âmbito da Justiça Federal, mediante a realização de exercícios e testes, desenvolvimento de projetos com fins didáticos e acompanhamento de aulas.

d) colaborar em projetos de iniciativa de outros desenvolvedores cadastrados, desde que admitido na equipe de desenvolvimento ou na comunidade do projeto pelo respectivo coordenador; e

e) colaborar com outros projetos, mantidos fora do âmbito da Justiça Federal da 3ª Região, mas que sejam de interesse da Administração Pública, desde que tal colaboração seja informada em seu cadastro de desenvolvedor colaborativo e aprovada pela SETI.

III – PROJETOS DE SOFTWARE

1. Para iniciar um novo projeto de software, o desenvolvedor cadastrado deverá encaminhar uma proposta ao patrocinador a que estiver vinculado, instaurando, para tanto, um novo expediente SEI, mediante o preenchimento de uma “Ficha Inicial do Projeto”, segundo o modelo em vigor.

•• Inovação no Judiciário

1.1. A proposta deverá contar, no mínimo, com as seguintes informações: (i) finalidade, escopo e justificativa do projeto; (ii) tecnologias, ferramentas e infraestrutura necessárias para o desenvolvimento, implantação e manutenção da solução de *software*; (iii) lista de integrantes da equipe de desenvolvimento, com especificação do coordenador ou dos coordenadores e dos *stakeholders*, todos os quais também assinarão o documento juntamente com seus respectivos superiores hierárquicos; (iv) universo de colaboradores externos, efetivos e potenciais, os critérios de admissão de novos colaboradores externos, os meios de divulgação do projeto e a intenção de manter clones do código-fonte em portais de repositórios mantidos por entes públicos fora da infraestrutura da Justiça Federal da 3ª Região, como, por exemplo, o Portal do Software Público Brasileiro, mantido pelo Ministério do Planejamento, Desenvolvimento e Gestão; (v) cronograma estimado para cada etapa de desenvolvimento; (vi) compromisso da equipe de desenvolvimento de concluir o projeto em todas as suas etapas; e (vii) forma de implantação sugerida e os meios necessários para efetivá-la.

1.2. Sendo a solução de software de interesse de outros órgãos da Justiça Federal da 3ª Região além daquele a que esteja vinculado o proponente, a proposta deverá ser elaborada de modo a contemplar também a futura implantação da solução de software para esses outros órgãos.

1.3. Visando à melhor compreensão do projeto, o proponente poderá (i) juntar ao expediente SEI documentos complementares e recursos audiovisuais; e (ii) agendar com o patrocinador a que estiver vinculado reuniões, exposições ou demonstrações presenciais, sem necessidade de registro no expediente.

1.4. O patrocínio será negado se o projeto (i) estiver em desacordo com norma legal ou infralegal; (ii) não atender a critérios de economicidade e eficiência; (iii) não atender a critérios de conveniência e oportunidade; (iv) não atender a critérios de utilidade; ou (v) não estiver alinhado às diretrizes de planejamento estratégico da Justiça Federal.

1.5. Negado o patrocínio, o expediente será arquivado.

1.6. O patrocínio poderá ser concedido em parte. Nesse caso, o patrocinador especificará as partes não acolhidas, as alterações necessárias ou as sugestões de aperfeiçoamento e o proponente terá a opção de desistir do projeto ou ajustá-lo às orientações do patrocinador, submetendo-o a uma nova análise.

1.7. Concedido o patrocínio, não havendo desistência, o proponente prestará no mesmo expediente SEI as informações listadas no Anexo I desta IN, e encaminhará o expediente à SETI para análise e aprovação técnica.

1.8. A SETI, ouvida a UDEM, terá 30 (trinta) dias para analisar a proposta e somente a rejeitará se o projeto (i) estiver em conflito ou já estiver abarcado por outro projeto já em andamento na Justiça Federal da 3ª Região; (ii) não atender

a critérios de segurança de TI; (iiii) for incompatível com a infraestrutura de TI ou com os aplicativos em funcionamento na Justiça Federal da 3ª Região; (iv) violar norma ou diretriz de TI em vigor na Justiça Federal; (v) estiver em contradição com as diretrizes de planejamento estratégico da área de TI; ou (vi) não estiver em conformidade com o item III-1.2 deste módulo.

1.9. No lugar de autorizar o prosseguimento do projeto, a SETI poderá assumir o seu desenvolvimento, concedendo ou não à equipe de desenvolvimento ou a parte dela a opção de continuar a participar do projeto na modalidade de desenvolvimento colaborativo por contrato.

1.10. Na hipótese do item anterior, caberá à SETI confeccionar o Documento de Oficialização de Demanda para Desenvolvimento de Software e seguir com todas as demais formalidades e etapas previstas na IN 37-04, implantada pelo Resolução PRES n. 424/2015.

1.11. A SETI poderá conceder autorização parcial ao projeto, especificando as partes não acolhidas, as alterações necessárias ou as sugestões de aperfeiçoamento. Nesse caso, o proponente terá a opção de desistir do projeto ou ajustá-lo às orientações da SETI, submetendo-o a uma nova análise.

2. Autorizado o prosseguimento do projeto, não havendo desistência, a SETI, no prazo máximo de 10 (dez) dias úteis, fornecerá à equipe de desenvolvimento todos os meios necessários para o uso das tecnologias, ferramentas e infraestrutura aprovadas.

IV – DESENVOLVIMENTO DO PROJETO DE SOFTWARE

1. Iniciado o desenvolvimento, cabe ao coordenador do projeto:

a) coordenar a equipe de desenvolvimento e a comunidade do projeto, cuidando especificamente de (i) selecionar novos membros da equipe de desenvolvimento e colaboradores externos, (ii) aplicar e fazer aplicar as regras de desenvolvimento colaborativo entre os membros da equipe de desenvolvimento e os colaboradores externos; (iii) administrar a incorporação ou rejeição dos *patches* apresentados; e (iv) zelar pela produção de todos os artefatos de documentação, os quais deverão ser preferencialmente produzidos *pari passu* às atividades de desenvolvimento; e b) informar à SETI a conclusão de cada etapa do desenvolvimento por meio do respectivo expediente SEI.

2. O coordenador do projeto pode formular ao patrocinador e à SETI, a qualquer tempo, pedidos de aditamento ao projeto, inclusive no tocante a tecnologias utilizadas, infraestrutura necessária e cronograma de entrega.

3. A SETI poderá assumir o desenvolvimento do projeto a qualquer tempo, mediante prévia autorização do patrocinador, concedendo ou não à equipe de desenvolvedores ou a parte dela a opção de continuar a participar do desenvolvimento, mas na modalidade por contrato.

•• Inovação no Judiciário

4. A conclusão do projeto será formalizada mediante a aceitação, por declaração escrita, da solução de software pelo patrocinador e pela SETI, sem necessidade de qualquer documento adicional.

4.1. Cabe à SETI cadastrar o projeto no catálogo de projetos da Justiça Federal da 3ª Região.

4.2. A equipe de desenvolvimento poderá dar continuidade à manutenção e ao aperfeiçoamento do código-fonte por tempo indeterminado, nos termos da autorização inicial, inclusive com a participação dos colaboradores externos.

5. No desenvolvimento de soluções de software pelo regime desta IN poderão ser adotadas técnicas e metodologias de desenvolvimento ágil, mescladas ou não com técnicas tradicionais ou outras que sejam consideradas padrão de indústria, desde que produzidos todos os artefatos de documentação aqui previstos, sem os quais a entrega da solução de software poderá ser recusada.

6. Além de outras licenças que venham a ser aprovadas pela SETI, os desenvolvedores cadastrados poderão utilizar licenças no padrão *MIT*, *Apache 2.0*, *3-Clause BSD*, *2-Clause BSD*, *GPL 3.0* e *Creative Commons CC-BY-SA 3.0 BR*.

V – IMPLANTAÇÃO DE SOLUÇÕES DE SOFTWARE

1. A SETI tem a prerrogativa de implantar ou não os projetos desenvolvidos na modalidade por contrato, segundo critério de conveniência e oportunidade.

2. No caso de projeto de código aberto, cabe à SETI definir, após a entrega da solução de software, a modalidade de implantação, independentemente da modalidade que tiver sido inicialmente sugerida pelo proponente.

2.1. Os projetos de implantação autorizada poderão ser descontinuados a qualquer tempo pelo coordenador do projeto, com autorização do patrocinador, mediante prévio aviso a todos os usuários, com antecedência mínima de 30 (trinta) dias.

2.2. A alteração da modalidade de implantação para implantação assumida ou a descontinuação do projeto em caso de implantação autorizada não impedem que o ramo originário do código-fonte continue a ser mantido pela equipe de desenvolvimento, inclusive com a participação dos colaboradores externos, nos termos do item IV-4.2 deste módulo

3. No caso de implantação assumida, caberá à SETI produzir a documentação de infraestrutura e de mudança previstas na Resolução PRES n. 424/2015 (IN 37-04) e na Resolução PRES n. 167/2018.

4. No caso de implantação autorizada, a documentação prevista na Resolução PRES n. 424/2015 (IN 37-04) e na Resolução PRES n. 167/2018 será, em regra, dispensada, cabendo à SETI apontar a sua necessidade, caso a caso, fundamentadamente.

VI – MANUTENÇÃO E MELHORIAS

1. As soluções de software desenvolvidas pelo modelo de código-aberto continuarão a ser mantidas pela equipe de desenvolvimento sem necessidade de qualquer formalidade adicional.

1.1. As alterações posteriores na composição da equipe de desenvolvimento deverão ser aprovadas pelo órgão patrocinador e registradas no expediente SEI do projeto.

1.2. Aplica-se às atividades de manutenção do código-fonte o disposto no item IV-5 deste módulo.

2. As melhorias simples ao código-fonte poderão ser efetuadas sem necessidade de prévia abertura de ordem de serviço, de documento de oficialização de demanda ou de qualquer outro tipo de formulário, mas deverão ser documentadas de forma clara e completa e comunicadas à SETI antes de sua publicação em ambiente de produção.

3. As melhorias substanciais ao código-fonte deverão ser previamente submetidas à aprovação do patrocinador e da SETI, mediante apresentação de relatório sucinto das alterações pretendidas e de sua justificativa.

3.1. Uma vez aprovadas e concluídas, as melhorias a que se refere este item serão submetidas a nova etapa de homologação e deverão ser aprovadas também pelos *stakeholders* integrantes da equipe de desenvolvimento.

4. A SETI poderá, por razões técnicas, vedar a publicação de melhorias simples ou substanciais em ambiente de produção ou especificar as condições nas quais a publicação deve ser feita.

4.1. Caberá à SETI cuidar dos procedimentos relacionados ao gerenciamento de mudanças de TI necessários à publicação das melhorias simples e substanciais em ambiente de produção, inclusive no que se refere à confecção dos artefatos previstos na Resolução PRES n. 167/2018.

VII – DOCUMENTAÇÃO

1. Os projetos de código aberto poderão ser documentados de forma simplificada, sem necessidade de preenchimento de formulários, mediante a juntada de todas as informações e artefatos pertinentes no próprio expediente SEI do projeto.

1.1. Nos projetos de novas soluções de software, além das informações listadas no Anexo I, o coordenador deverá juntar ao expediente, até a conclusão do processo de desenvolvimento, um relatório contendo as informações mínimas listadas no Anexo II desta IN e, quando for caso, o modelo e dicionário de dados.

1.2. As melhorias nos projetos de código aberto poderão ser documentadas mediante utilização de versionamento semântico, comentários e ferramentas de automação.

2. Projetos de desenvolvimento colaborativo que envolvam o aperfeiçoamento de soluções de software já implantadas na Justiça Federal da 3ª Região seguirão as mesmas regras de documentação do projeto original.

MÓDULO 3: COOPERAÇÃO EM GESTÃO DE DADOS

1. A SETI franqueará aos órgãos de inovação e aos Centros Locais de Inteligência da Justiça Federal da 3ª Região amplo acesso ao armazém de dados por ela mantido, a fim de assegurar o bom desempenho de suas atividades de gestão de dados.

1.1. Os órgãos de inovação poderão manter armazéns de dados próprios na estrutura a que se refere o Módulo 4, mediante prévia autorização e sob a supervisão da SETI, para fins de colaborar no desenvolvimento do armazém de dados da Justiça Federal da 3ª Região mantido pela SETI.

1.2. Para o fim previsto no item anterior, sempre que as inovações introduzidas no armazém de dados de um órgão de inovação atingirem nível satisfatório de maturidade, referido órgão proverá a SETI com as instruções necessárias para reproduzir tais inovações em seu armazém de dados.

1.3. A SETI disponibilizará aos órgãos de inovação os meios necessários para a publicação de seus relatórios e análises de dados em servidor institucional, a fim de torná-los disponíveis a todos os órgãos destinatários das informações, efetivos ou potenciais.

MÓDULO 4: LABORATÓRIOS DE INOVAÇÃO TECNOLÓGICA

1. A SETI e a UDEM manterão relacionamento de permanente intercâmbio, parceria e colaboração com os órgãos de inovação tecnológica da Justiça Federal da 3ª Região, a fim de promover a pesquisa, a prospecção e o estudo de novas soluções tecnológicas voltadas ao aprimoramento do serviço público.

1.1. A cooperação será feita preferencialmente por meio das incubadoras de tecnologia associadas a laboratórios de inovação.

1.2. Para o fim previsto neste módulo, a SETI poderá fornecer às incubadoras de tecnologia infraestrutura própria para prototipagem, prospecção, aprendizagem, desenvolvimento, homologação e produção, segundo as especificações previamente autorizadas, em conformidade com os padrões de segurança e gestão de TI em vigor, mediante solicitação justificada da Comissão de Informática, da DFOR/MS ou da DFOR/SP, conforme o caso.

2. As incubadoras de tecnologia poderão abrir e manter contas institucionais em portais de repositórios com a finalidade de armazenar e gerenciar seus projetos de código aberto.

MÓDULO 5: ATIVIDADES QUE INDEPENDEM DE AUTORIZAÇÃO

1. Não estão sujeitas a autorização ou qualquer outra formalidade, podendo ser praticadas por qualquer servidor ou magistrado na Justiça Federal da 3ª Região, desde que não impliquem violação das regras de segurança e gestão de TI em vigor, as seguintes atividades de desenvolvimento:

- a) criação e distribuição de fórmulas, *scripts* ou códigos no contexto de atividades de gestão de dados, mediante o uso de ferramentas já adquiridas e distribuídas aos usuários ou cujo uso já esteja homologado pela SETI, como, por exemplo, ferramentas de *business intelligence* e planilhas de cálculo.
 - b) criação e distribuição de fórmulas, *scripts* ou códigos no bojo de aplicativos que integram pacotes *office* de uso autorizado na Justiça Federal da 3ª Região, como, por exemplo, macros para automação de planilhas de cálculo ou documentos de texto.
2. A SETI poderá bloquear ou limitar a execução das fórmulas, *scripts* e códigos mencionados no item anterior sempre que haja risco para a segurança dos usuários ou da infraestrutura de TI ou para o bom funcionamento dos demais aplicativos e sistemas da Justiça Federal da 3ª Região.

ANEXO I – INFORMAÇÕES MÍNIMAS PARA O INÍCIO DO PROJETO

O coordenador deve prestar, no início do projeto, as seguintes informações, as quais poderão ser ou não organizadas em formulário próprio pelo órgão patrocinador:

- a) finalidade do projeto: breve descrição do problema que o projeto pretende solucionar e dos fluxos de trabalho envolvidos.
- b) escopo do projeto: breve descrição da solução proposta, com menção a eventuais alterações nos fluxos de trabalho, e dos resultados esperados (p. ex. tipos de relatório a serem gerados, lista de tarefas a serem automatizadas, descrição das funcionalidades previstas etc.).
- c) beneficiários.
- d) equipe do projeto, incluindo ao menos um representante dos beneficiários.
- d) alinhamento estratégico.
- e) prazo estimado para a conclusão de cada etapa de desenvolvimento.

ANEXO II – INFORMAÇÕES MÍNIMAS DO RELATÓRIO DE ESPECIFICAÇÃO DE REQUISITOS

O Relatório de Especificação de Requisitos conterá, no mínimo, as seguintes informações, que poderão ou não ser organizadas em formulário próprio pelo órgão patrocinador:

- a) relação dos artefatos produzidos com suas especificações.
- b) memórias de reunião.
- c) relação das necessidades técnicas verificadas no desenvolvimento.
- d) código-fonte comentado.
- d) relação dos testes realizados, dos resultados obtidos e das alterações promovidas nessa fase.

•• Inovação no Judiciário

e) relação dos requisitos para implantação, das ocorrências havidas nessa fase e das alterações eventualmente promovidas no projeto ou no ambiente de produção.

ANEXO III – MODELO DE TERMO DE USO E PRIVACIDADE

ABRANGÊNCIA

Este documento se aplica a todas as rotinas, bibliotecas, *frameworks* e demais softwares que integram o projeto denominado [****nome****], objeto do expediente SEI n. [****número****], sendo todos os referidos softwares doravante designados, em conjunto, “**SOFTWARE**”.

TERMOS DE USO

A utilização de qualquer parte do **SOFTWARE** implica ciência e aceitação dos seguintes termos e condições:

1. O **SOFTWARE**, de uso facultativo, é fornecido a título gratuito, de boa-fé, no interesse dos próprios usuários, com a anuência destes, para [****finalidade****].
2. Os usuários ficam cientes de que o **SOFTWARE**, embora oferecido como ferramenta auxiliar ao trabalho no âmbito da Justiça Federal da 3ª Região e homologado nos termos do procedimento SEI anteriormente citado, teve seu desenvolvimento conduzido exclusivamente por [****desenvolvedor****], doravante designado **DESENVOLVEDOR**, sem qualquer participação direta dos órgãos da Justiça Federal da 3ª Região, inclusive da Secretaria de Tecnologia da Informação – SETI do Tribunal Regional Federal da 3ª Região, os quais são doravante designados, em conjunto, “**ADMINISTRAÇÃO**”.
3. Assim, é de inteira responsabilidade do **DESENVOLVEDOR** distribuir, instalar e atualizar o **SOFTWARE** e fornecer o respectivo suporte, não cabendo à **ADMINISTRAÇÃO** tomar qualquer dessas providências.
4. Sem prejuízo do disposto no item anterior, os usuários do **SOFTWARE** deverão reportar à **ADMINISTRAÇÃO**, na primeira oportunidade, eventual risco ou dano de qualquer espécie produzido pelo **SOFTWARE** aos respectivos usuários ou aos serviços da Justiça Federal da 3ª Região.
5. Os usuários deverão utilizar o **SOFTWARE** exclusivamente para a finalidade mencionada no item 1. No caso de constatarem qualquer incompatibilidade entre o uso do **SOFTWARE** e seus deveres funcionais, especialmente os de sigilo profissional, devem abster-se imediatamente de utilizar o **SOFTWARE** e reportar o problema à **ADMINISTRAÇÃO**.
6. O **DESENVOLVEDOR** fornece o **SOFTWARE** no estado em que se encontra (“*as is*”), sem assumir qualquer responsabilidade por eventuais falhas de concepção ou de execução dos códigos ou por erros decorrentes de seu mau uso, cabendo aos

próprios usuários, nesses casos, verificar a incorreção dos resultados e reportar o fato ao **DESENVOLVEDOR** para futuros aperfeiçoamentos.

7. No momento em que instalarem e/ou iniciarem a utilização do **SOFTWARE**, os usuários estarão no mesmo ato assumindo o compromisso de (i) não empregar qualquer parte do **SOFTWARE** ou de suas funcionalidades para fins ilícitos ou imorais; e (ii) agir não como “consumidores” passivos do **SOFTWARE**, mas como efetivos parceiros e colaboradores do **DESENVOLVEDOR**, prestando-lhe, na medida de sua disponibilidade e de seus conhecimentos, todo o auxílio possível na identificação e na resolução de defeitos (“*bugs*”) ou de falhas de concepção dos códigos.

POLÍTICA DE PRIVACIDADE

O **DESENVOLVEDOR** assegura aos usuários do **SOFTWARE** a observância das seguintes regras de confidencialidade:

1. O **DESENVOLVEDOR** não obterá, armazenará ou cederá de qualquer forma, a quem quer que seja, informações de natureza pessoal ou profissional dos usuários, sem o seu prévio e expresso consentimento, salvo na extensão necessária ao bom funcionamento do **SOFTWARE**. Não se incluem nessa restrição, todavia, as informações de domínio público.
2. Todas as informações dos usuários porventura necessárias ao funcionamento do **SOFTWARE** serão mantidas em absoluto sigilo e descartadas quando não mais necessárias.
3. A regra do item anterior não se aplica aos relatórios automáticos de falhas (*bugs*), aos relatórios de performance e às estatísticas de uso, os quais poderão ser mantidos em bancos de dados próprios, desde que utilizados exclusivamente para tais finalidades, com as devidas restrições de acesso que assegurem a preservação do sigilo.
4. O **DESENVOLVEDOR** cuidará para que as informações a que se referem os itens 2 e 3 não cheguem ao conhecimento de terceiros, salvo na extensão necessária para o correto funcionamento do **SOFTWARE**.
5. Todo o procedimento de autenticação será gerenciado integralmente pelo órgão técnico competente da **ADMINISTRAÇÃO**, sem que o **DESENVOLVEDOR** ou terceiros tenham acesso a qualquer informação sobre os dados de login dos usuários senão as que forem disponibilizadas pelo referido procedimento de autenticação.

