



# Manual de Boa Governação de Código Aberto

Autores: Participantes da OSPO Alliance & da GGI

Version: v1.1

Date: 2023-06-26

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Contexto . . . . .	3
1.2	Sobre a Good Governance Initiative . . . . .	4
1.3	Sobre a OSPO Alliance . . . . .	4
1.4	Traduções . . . . .	5
1.5	Colaboradores . . . . .	5
1.6	Licença . . . . .	5
<b>2</b>	<b>Organização</b>	<b>6</b>
2.1	Terminologia . . . . .	6
2.2	Objetivos . . . . .	6
2.3	Atividades canônicas . . . . .	7
2.4	Quadros de pontuação de atividades personalizados . . . . .	7
<b>3</b>	<b>Metodologia</b>	<b>8</b>
3.1	Preparar o palco . . . . .	8
3.2	Fluxo de trabalho . . . . .	8
3.3	Configuração manual: usar Quadros de pontuação de atividades personalizados . . . . .	9
3.4	Configuração automática: usar o recurso de implantação do GGI . . . . .	10
3.5	Desfrutar . . . . .	11
<b>4</b>	<b>Atividades de objetivos de utilização</b>	<b>12</b>
4.1	Inventário de competências e recursos de código aberto . . . . .	12
4.2	Crescimento de competências de código aberto . . . . .	13
4.3	Supervisão de código aberto . . . . .	15
4.4	Software de empresa de código aberto . . . . .	16
4.5	Gerir habilidades e recursos de código aberto . . . . .	17
<b>5</b>	<b>Atividades de objetivos de confiança</b>	<b>19</b>
5.1	Gerir a conformidade legal . . . . .	19
5.2	Gerir vulnerabilidades de software . . . . .	20
5.3	Gerir as dependências de software . . . . .	22
5.4	Gerir indicadores chave . . . . .	24
5.5	Executar revisões de código . . . . .	25
<b>6</b>	<b>Atividades do objetivo cultura</b>	<b>27</b>
6.1	Promover práticas recomendadas de desenvolvimento de código aberto . . . . .	27
6.2	Contribuir a projetos de código aberto . . . . .	28
6.3	Pertencer à comunidade de código aberto . . . . .	29
6.4	Perspetiva do RH . . . . .	30
6.5	Primeiro a montante . . . . .	32
<b>7</b>	<b>Atividades de objetivos de envolvimento</b>	<b>34</b>
7.1	Envolver-se em projetos de código aberto . . . . .	34
7.2	Apoiar comunidades de código aberto . . . . .	35
7.3	Afirmar publicamente a utilização de código aberto . . . . .	36
7.4	Envolver-se com vendedores de código aberto . . . . .	36
7.5	Política de aprovisionamento de código aberto . . . . .	37
<b>8</b>	<b>Atividades de objetivos de estratégia</b>	<b>40</b>
8.1	Estabelecer uma estratégia para a governação empresarial de código aberto . . . . .	40
8.2	Consciência da administração superior . . . . .	41
8.3	Código aberto e soberania digital . . . . .	42
8.4	Código aberto permitindo a inovação . . . . .	43
8.5	Código aberto permitindo a transformação digital . . . . .	45
<b>9</b>	<b>Conclusão</b>	<b>47</b>
9.1	Contato . . . . .	47
9.2	Apêndice: Modelo personalizado do Quadro de pontuação de atividades personalizados . . . . .	47

# 1 Introdução

Este documento apresenta uma metodologia para implementar uma gestão profissional de software de código aberto numa organização. Aborda a necessidade de utilizar software de código aberto de forma adequada e justa, salvaguardar a empresa das ameaças técnicas, legais e de PI e maximizar as vantagens do software de código aberto. Onde quer que uma organização se encontre nestes tópicos, este documento propõe orientações e ideias para avançar e fazer da sua jornada um sucesso.

## 1.1 Contexto

A maioria dos grandes utilizadores e integradores de sistemas já utilizam Software Livre e de Código Aberto (FOSS) quer nos seus sistemas de informação, quer nas suas divisões de produtos e serviços. O cumprimento do código aberto tornou-se uma preocupação crescente, e muitas empresas grandes estabeleceram agentes de cumprimento. Contudo, enquanto a limpeza da cadeia de produção de uma empresa de código aberto - a conformidade - é fundamental, os utilizadores *devem* dar de volta às comunidades e contribuir para a sustentabilidade do ecossistema de código aberto. Vemos uma governação de código aberto que abrange todo o ecossistema, envolvendo as comunidades locais, alimentando uma relação saudável com fornecedores de software de código aberto e especialistas em serviços. Isto leva a conformidade ao próximo nível e é do que se trata numa "boa" governação de código aberto.

Esta iniciativa abrange mais do que o cumprimento e a responsabilidade. Trata-se de sensibilizar as comunidades de utilizadores finais (muitas vezes sendo os próprios criadores de software) e integradores de sistemas, e desenvolver relações mutuamente benéficas no ecossistema europeu de FOSS.

A OSS Good Governance permite organizações de todos os tipos -- empresas, pequenas e grandes, câmaras municipais, universidades, associações, etc. -- de maximizar os benefícios derivados ao código aberto por ajudá-las a alinhar pessoas, processos, tecnologia e estratégia. E nesta área, a de maximizar as vantagens do open source, especialmente na Europa, todos continuam a aprender e a inovar, sem que ninguém saiba em que ponto se encontram no que diz respeito ao estado da arte neste domínio.

Esta iniciativa visa ajudar organizações a alcançar esses objetivos com:

- Um catálogo estruturado de **atividades**, um roteiro para a implementação da gestão profissional de software de código aberto.
- A **ferramenta de gestão** para definir, monitorizar, relatar e comunicar sobre o progresso.
- Um **caminho claro e prático para a melhoria**, com passos pequenos e acessíveis para mitigar riscos, educar as pessoas, adaptar processos, comunicar para dentro e para fora do domínio da organização.
- **Orientações** e uma série de **referências selecionadas** sobre o licenciamento de código aberto, práticas recomendadas, formação, e envolvimento do ecossistema para alavancar a consciencialização e a cultura de código aberto, consolidar o conhecimento interno e ampliar a liderança.

Este guia foi desenvolvido com os seguintes requisitos em mente:

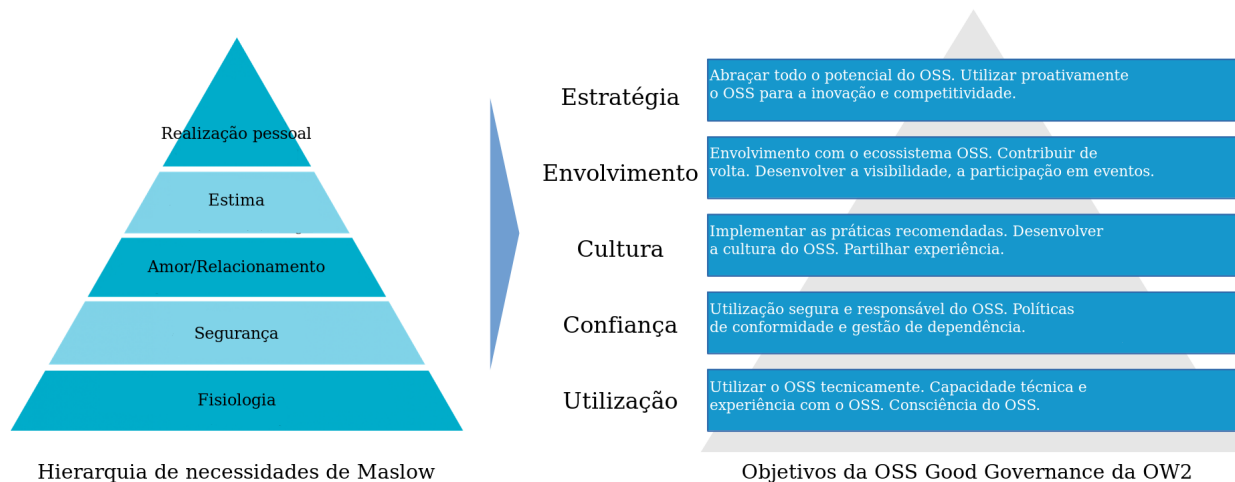
- Qualquer tipo de organização é abrangido: desde PMEs a grandes empresas e organizações sem fins lucrativos, desde autoridades locais (por exemplo, câmaras municipais) a grandes instituições (por exemplo, instituições europeias ou governamentais). O quadro fornece blocos de construção para uma estratégia e dicas para a sua realização, mas *como* as atividades são executadas depende inteiramente do contexto do programa e depende do gestor do programa. Pode ser útil procurar serviços de consultoria e trocar ideias com pares.
- Nenhuma suposição é feita sobre o nível de conhecimentos técnicos na organização ou do domínio de atividade. Por exemplo, algumas organizações terão de criar um currículo de formação completo, enquanto outras poderão simplesmente propor material improvisado às equipas.

Algumas atividades não serão relevantes para todas as situações, mas o quadro completo ainda assim fornece um roteiro abrangente e prepara o caminho para estratégias à medida.

## 1.2 Sobre a Good Governance Initiative

Na OW2, uma iniciativa é um esforço conjunto para responder a uma necessidade do mercado. A **Good Governance Initiative** propõe um quadro metodológico para implementar uma gestão profissional de software de código aberto em organizações.

A Good Governance Initiative baseia-se num modelo abrangente inspirado na hierarquia popular de Abraham Maslow das necessidades e motivações humanas, como ilustrado pela imagem abaixo.



Através de ideias, diretrizes e atividades, a Good Governance Initiative fornece um plano para a implementação de entidades organizacionais encarregadas da gestão profissional de software de código aberto, o que também é chamado OSPO (= Open Source Program Offices/Departamentos de Programas de Código Aberto). A metodologia é também um sistema de gestão para definir prioridades, e acompanhar e partilhar o progresso.

Ao implementarem a metodologia da OSS Good Governance, as organizações irão melhorar as suas competências em várias direções, incluindo:

- **utilizar** software de código aberto de forma adequada e segura numa empresa para melhorar a reutilização e manutenção do software e a velocidade de desenvolvimento do software;
- **mitigar** os riscos legais e técnicos associados a código externo e à colaboração;
- **identificar** a formação necessária para as equipas, dos programadores aos chefes de equipa e gestores, pelo que todos partilham a mesma visão;
- **prioritizar** objetivos e atividades, para desenvolver uma estratégia eficiente de código aberto;
- **comunicar** eficazmente numa empresa e com o exterior para desfrutar ao máximo da estratégia de código aberto;
- **melhorar** a competitividade e atratividade da organização para os melhores talentos de código aberto.

## 1.3 Sobre a OSPO Alliance

A **OSPO Alliance** foi lançada por uma coligação das organizações europeias sem fins lucrativos de código aberto principais, incluindo OW2, Eclipse Foundation, OpenForum Europe e a Foundation for Public Code, com a missão de aumentar a sensibilização para o código aberto na Europa e no mundo e de promover a gestão estruturada e profissional de código aberto por empresas e administrações.

Enquanto a iniciativa da Good Governance se foca no desenvolvimento de uma metodologia de gestão, a OSPO Alliance tem o objetivo mais amplo de ajudar empresas, particularmente nos setores não tecnológicos e instituições públicas, a descobrir e compreender o código aberto, a começar a beneficiar dele em todas as suas atividades e a crescer para hospedar as próprias OSPOs.

A OSPO Alliance estabeleceu o site da **OSPO Alliance** hospedado em <https://ospo-alliance.org>. A OSPO Alliance serve a comunidade com um lugar seguro para discutir e trocar informação sobre os tópicos das OSPOs e fornece um repositório para um conjunto abrangente de recursos

para empresas, instituições públicas e organizações acadêmicas e de investigação. A OSPO.Zone liga-se à OSPOs em toda a Europa e no mundo, bem como a organizações comunitárias de apoio. Encoraja práticas recomendadas e fomenta a contribuição para a sustentabilidade do ecossistema de código aberto. Consulte o sítio web [OSPO Alliance](#) para uma rápida visão geral das estruturas complementares das práticas recomendadas de gestão de TI.

O sítio web [OSPO Alliance](#) é também o local onde recolhemos comentários sobre a iniciativa e o seu conteúdo (por exemplo, atividades, corpo de conhecimentos) da comunidade em geral.

## 1.4 Traduções

Há um trabalho comunitário em andamento para traduzir o Manual da GGI em vários idiomas. À medida que o progresso evolui rapidamente, recomendamos que confira o nosso site oficial para obter uma lista completa de traduções disponíveis.

Veja <https://hosted.weblate.org/projects/ospo-zone-ggi/#languages>

O manual do GGI é traduzido usando [Weblate](#), um projeto e plataforma de código aberto que oferece hospedagem gratuita para projetos de código aberto. Queremos agradecer-lhes profundamente, assim como todos os nossos colaboradores de tradução. Vocês são incríveis.

## 1.5 Colaboradores

As seguintes grandes pessoas contribuíram à Good Governance Initiative:

- Frédéric Aatz (Microsoft França)
- Boris Baldassari (Castalia Solutions, Eclipse Foundation)
- Philippe Bareille (cidade de Paris)
- Gaël Blondelle (Eclipse Foundation)
- Vicky Brasseur (Wipro)
- Philippe Carré (Nokia)
- Pierre-Yves Gibello (OW2)
- Michael Jaeger (Siemens)
- Sébastien Lejeune (Thales)
- Max Mehl (Free Software Foundation Europe)
- Hervé Pacault (Orange)
- Stefano Pampaloni (RIOS)
- Christian Paterson (OpenUp)
- Simon Phipps (Meshed Insights)
- Silvério Santos (Orange Business Services)
- Cédric Thomas (OW2)
- Nicolas Toussaint (Orange Business Services)

## 1.6 Licença

Esta obra é licenciada sob a licença [Creative Commons Attribution 4.0 International](#) (CC-BY 4.0). Do sítio web da Creative Commons:

Você tem o direito de:

- Partilhar — copiar e redistribuir o material em qualquer suporte ou formato
- Adaptar — remixar, transformar, e criar a partir do material

para qualquer fim, mesmo que comercial.

Você deve atribuir o devido crédito, fornecer um link para a licença, e indicar se foram feitas alterações. Você pode fazê-lo de qualquer forma razoável, mas não de uma forma que sugira que o licenciante o apoia ou aprova o seu uso. [Fonte: <https://creativecommons.org/licenses/by/4.0/deed.pt>]

Todos os conteúdos são direitos autorais: 2020-2022 participantes da OW2 & The Good Governance Initiative.

## 2 Organização

### 2.1 Terminologia

O projeto da metodologia de Boa Governança do OSS está estruturado em torno de quatro conceitos-chave: objetivos, atividades canônicas, quadros de pontuação de atividades personalizadas e Iteração.

- **Objetivos:** um objetivo é um conjunto de atividades associadas a uma área de preocupação comum, existem cinco objetivos: objetivo de utilização, objetivo de confiança, objetivo de cultura, objetivo de envolvimento e objetivo de estratégia. Os objetivos podem ser alcançados de forma independente, em paralelo, e iterativamente refinados por atividades.
- **Atividades canônicas:** num objetivo, uma atividade aborda uma única preocupação ou tópico de desenvolvimento - tal como a gestão da conformidade legal - que pode ser utilizada como um passo incremental em direção aos objetivos do programa. O conjunto completo de atividades, tal como definido pela GGI, chamam-se atividades canônicas.
- **Quadro de pontuação de atividades personalizado (CAS: Customised Activity Scorecard):** para implementar a GGI numa determinada organização, as atividades canônicas devem ser adaptadas às especificidades do contexto, construindo assim um conjunto de quadros personalizados de pontuação de atividades. O quadro personalizado de pontuação de atividades descreve como a atividade será implementada no contexto da organização e como o progresso será monitorado.
- **Iteração:** o OSS Good Governance é um sistema de gestão e assim requer avaliações e revisões periódicas. Pense no sistema de contabilidade de uma organização, é um processo contínuo com pelo menos um ponto de controlo anual, o balanço; da mesma forma, o processo de OSS Good Governance requer pelo menos uma revisão anual, mas as revisões podem ser parciais ou mais frequentes, dependendo das atividades.

### 2.2 Objetivos

As atividades canônicas definidas pela IGP estão organizadas em objetivos. Cada objetivo aborda uma área específica de progresso no âmbito do processo. Da utilização à estratégia, os objetivos cobrem questões relacionadas com todos os intervenientes, desde equipas de desenvolvimento até à administração superior.

- **Objetivo Utilização:** este objetivo cobre os passos básicos na utilização de software de código aberto. As atividades relacionadas ao objetivo de utilização cobrem os primeiros passos através de um programa de código aberto, a identificar a eficiência da utilização do código aberto e o que dá à organização. Inclui formação e gestão do saber, a produzir inventários de software de código aberto existente já utilizado internamente e apresenta conceitos de código aberto que podem ser utilizados ao longo de todo o processo.
- **Objetivo Confiança:** este objetivo trata-se da utilização segura de código aberto. O objetivo da confiança trata da conformidade legal, dependência e gestão de vulnerabilidades e visa geralmente criar confiança na forma como a organização utiliza e gere código aberto.
- **Objetivo Cultura:** o objetivo de cultura inclui atividades que visam fazer as equipas sentirem-se à vontade com código aberto, participar individualmente em atividades de colaboração, compreender e implementar as práticas recomendadas de código aberto. Este objetivo fomenta um sentimento de pertença à comunidade de código aberto entre os indivíduos.
- **Objetivo Envolvimento:** este objetivo é de se envolver com o ecossistema de código aberto a nível corporativo. Os recursos humanos e financeiros estão orçamentados para contribuir de volta para projetos de código aberto. Aqui, a organização afirma que é um cidadão responsável de código aberto e reconhece a sua responsabilidade de assegurar a sustentabilidade do ecossistema de código aberto.
- **Objetivo Estratégia:** este objetivo é o de fazer o código aberto visível e aceitável nos níveis mais altos da gestão empresarial. Trata-se de reconhecer que o código aberto é um facilitador estratégico da soberania digital, inovação de processos e, em geral, uma fonte de atratividade e boa vontade.

## 2.3 Atividades canônicas

As atividades canônicas estão no centro do projeto GGI. Na sua versão inicial, a Metodologia GGI oferece cinco atividades canônicas por objetivo, 25 no total. As atividades canônicas são descritas a usar as seguintes seções predefinidas:

- *Descrição*: um resumo do tópico que a atividade aborda e os passos para a conclusão.
- *Avaliação de oportunidades*: descreve porque e quando é relevante realizar esta atividade.
- *Avaliação do progresso*: descreve como medir o progresso da atividade e avaliar o seu sucesso.
- *Ferramentas*: uma lista de tecnologias ou ferramentas que podem ajudar a realizar esta atividade.
- *Recomendações*: dicas e práticas recomendadas recolhidas dos participantes do GGI.
- *Recursos*: ligações e referências para ler mais sobre o tema abrangido pela atividade.

### Descrição

Esta secção fornece uma descrição de alto nível da atividade, um resumo do tópico para definir o objetivo da atividade no contexto da abordagem de código aberto num objetivo.

### Avaliação de oportunidades

Para ajudar a estruturar uma abordagem iterativa, cada atividade tem uma secção de "avaliação de oportunidades", com uma ou mais questões anexas. A avaliação de oportunidades concentra-se na razão pela qual é relevante realizar esta atividade, que necessidades visa colmatar. A avaliação da oportunidade ajudará a definir quais são os esforços esperados, os recursos necessários e ajudará a avaliar os custos e o RSI esperado.

### Avaliação do progresso

Este passo concentra-se na definição de objetivos, KPIs e no fornecer *pontos de verificação* que ajudam a avaliar o progresso na atividade. Os pontos de verificação são sugeridos, podem ajudar a definir um roteiro para o processo de Good Governance, as prioridades e como o progresso será medido.

### Ferramentas

Aqui estão as ferramentas listadas que podem ajudar na entrega da atividade ou instrumento de uma etapa específica das atividades. As ferramentas não são uma recomendação obrigatória, nem fingem ser exaustivas, mas são sugestões ou categorias a serem elaboradas com base no contexto existente.

### Recomendações

Esta secção é regularmente atualizada com opiniões dos utilizadores e todo o tipo de recomendações que podem ajudar a gerir a atividade.

### Recursos

Os recursos são propostos para alimentar a abordagem com estudos de fundo, documentos de referência, eventos ou conteúdo on-line para enriquecer e desenvolver a abordagem relacionada sobre a atividade. Os recursos não são exaustivos, são pontos de partida ou sugestões para expandir a semântica da atividade conforme o próprio contexto.

## 2.4 Quadros de pontuação de atividades personalizados

Os quadros de pontuação de atividades personalizados (CAS: Customised Activity Scorecards) são ligeiramente mais detalhados do que as atividades canônicas. Um CAS inclui detalhes específicos para a organização que implementa a GGI. A utilização do CAS é descrita na secção metodologia.

## 3 Metodologia

A implementação da metodologia da OSS Good Governance é, ultimamente, uma iniciativa consequente e com impacto. Envolve várias categorias de pessoas, serviços e processos de empresas, desde as práticas quotidianas à gestão de RH e dos promotores de nível executivo. Não existe realmente um mecanismo mágico para implementar uma boa governação de código aberto. Tipos de organização e culturas e situações de empresas diferentes exigirão abordagens diferentes à boa governação de código aberto. Para cada organização, haverá constrangimentos e expectativas diferentes, levando a diferentes caminhos e formas de gestão do programa.

Com isto em mente, a Good Governance Initiative fornece um plano genérico de atividades que pode ser adaptado ao próprio domínio, cultura e requisitos de uma organização. Embora o projeto alegue ser abrangente, a metodologia pode ser implementada progressivamente. É possível arrancar com o programa simplesmente selecionando os objetivos e atividades mais relevantes no contexto específico de cada um. A ideia é construir um primeiro esboço do roteiro para ajudar a criar a iniciativa local.

Para além deste quadro, também recomendamos vivamente o contacto com colegas através de uma rede estabelecida como a iniciativa europeia [OSPO Alliance](#), ou outras iniciativas semelhantes do grupo TODO, ou OSPO++. O que é importante é poder trocar ideias com pessoas que dirigem uma iniciativa semelhante e partilhar problemas encontradas e soluções existentes.

### 3.1 Preparar o palco

Dada a ambição da metodologia da boa governação e o seu impacto potencialmente amplo, é importante comunicar com uma variedade de pessoas numa organização. Seria apropriado envolvê-las para estabelecer um conjunto inicial de expectativas e requisitos realistas para um bom início, atrair interesse e apoio. Uma boa ideia é a publicação de Quadros de pontuação de atividades personalizados na plataforma de colaboração da organização para poderem ser utilizados para comunicar com as partes interessadas. Algumas dicas:

- Identifique as principais partes interessadas, faça-as concordar sobre um conjunto de objetivos primários. Envolve-os no sucesso da iniciativa, como parte da própria agenda.
- Obtenha a adesão inicial, concorde com as etapas e o ritmo e configure verificações regulares para os informar sobre o progresso.
- Certifique-se que eles entendam os benefícios do que pode ser alcançado e do que envolve: a melhoria esperada deve ser clara e o resultado visível.
- Estabeleça um primeiro diagnóstico ou estado da arte de código aberto na organização candidata. Resultado: um documento que descreva o que este programa irá alcançar, onde está a organização e para onde pretende ir.

### 3.2 Fluxo de trabalho

Como profissionais de software modernos, gostamos de métodos ágeis que definem incrementos pequenos e seguros, uma vez que é boa prática reavaliar a situação regularmente e fornecer resultados intermédios significativos mínimos.

No contexto de um programa do OSPO em funcionamento, isto é altamente relevante, uma vez que muitos aspetos secundários irão mudar ao longo do tempo, desde a estratégia e resposta da organização de código aberto até à disponibilidade e envolvimento das pessoas. Reavaliação e iteração periódica também permitem a adaptação à aceitação do programa em curso, um melhor acompanhamento das tendências e oportunidades atuais, e benefícios incrementais pequenos para as partes interessadas e para a organização na totalidade.

Idealmente, a metodologia poderia ser implementada em cinco fases, como segue:

1. **Descoberta** Entender os conceitos-chave, apropriar-se na metodologia, alinhar as expectativas das metas.
2. **Personalização** Adaptação da descrição da atividade e avaliação de oportunidades às especificidades da organização.
3. **Priorização** Identificação de objetivos e resultados-chave, tarefas e ferramentas, cronograma de agendamento e elaboração de cronograma.



4. **Ativação** Finalização do cartão de pontuação, orçamento, atribuições, tarefas de documentação sobre o gestor de problemas.
5. **Iteração** Avaliar e pontuar resultados, destacar questões, melhorar, ajustar. Itere todos os trimestres ou semestres.

Preparação para a primeira iteração do programa:

- Identifique um primeiro conjunto de tarefas a trabalhar, e atribua prioridade conforme as necessidades (lacunas até ao estado desejado) e a linha do tempo. Resultado: uma lista de tarefas a trabalhar durante a iteração.
- Defina um conjunto de requisitos e áreas de melhoria, comunique às partes interessadas e utilizadores finais, e obtenha a sua aprovação ou compromisso.
- Preencha os cartões de pontuação para acompanhar o progresso. Um modelo de cartão de pontuação pode ser descarregado do [repositório GGI](#).

No final de cada iteração, faça uma retrospectiva e prepare-se para a próxima iteração:

- Comunique sobre as últimas melhorias.
- Avalie onde se encontra, se as tarefas visadas tiverem sido concluídas, refine o roteiro em conformidade.
- Verifique os pontos ou questões de dor restantes, peça apoio a outros atores ou serviços, se necessário.
- Redefine as tarefas conforme o contexto atualizado.
- Define um novo subconjunto de tarefas a executar.

### 3.3 Configuração manual: usar Quadros de pontuação de atividades personalizados

Um quadro de pontuação de atividades personalizado é uma forma que descreve uma atividade canónica personalizada para as especificidades de uma organização. Em conjunto, o conjunto de quadros personalizados de pontuação de atividades fornece o roteiro para a gestão de software de código aberto.

Observe que, a partir da experiência inicial com a metodologia, leva até uma hora para adaptar uma atividade canónica a um quadro personalizado de pontuação específico a uma organização.

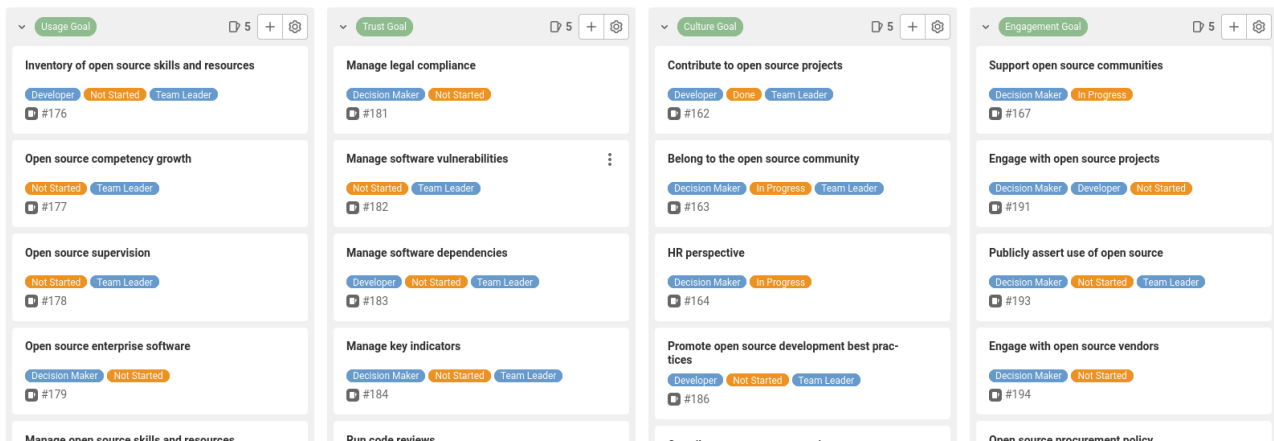
O Quadro de pontuação de atividades personalizado contém as secções seguintes:

- **Desambiguação do título** Antes de mais, leve alguns minutos a desenvolver uma compreensão do que a atividade pode ser e da relevância dela, como se pode enquadrar na sua jornada global de gestão do OSS.
- **Descrição personalizada** Adapte a atividade às especificidades da organização, a delimitar o âmbito. Defina o âmbito da atividade, o caso de uso particular que irá abordar.
- **Avaliação de oportunidades** Explique porque é relevante empreender esta atividade, que necessidades aborda. Quais são os nossos pontos de dor? Quais são as oportunidades para progredir? O que se pode ganhar?
- **Objetivos** Defina alguns objetivos cruciais para a atividade. Pontos de dor a serem corrigidos, oportunidades de progresso, desejos. Identifique tarefas-chave. O que pretendemos alcançar nesta iteração.
- **Ferramentas** Tecnologias, ferramentas e produtos utilizados na atividade.
- **Notas operacionais** Indicações sobre a abordagem, o método, a estratégia para progredir nesta atividade.
- **Resultados-chave** Defina resultados a esperar mensuráveis e verificáveis. Escolha resultados que indiquem o progresso em relação aos objetivos. Indique aqui os KPIs.
- **Progresso e pontuação** Progresso é, em %, a taxa de conclusão do resultado; pontuação é a classificação de sucesso pessoal.
- **Avaliação pessoal** Para cada resultado pode acrescentar uma breve explicação e explicar a sua taxa de satisfação pessoal expressa na pontuação.
- **Linha do tempo** Indicar datas de início e fim, tarefas de faseamento, passos críticos, marcos.
- **Esforços** Avalie o tempo e os recursos materiais solicitados, internos e de terceiros. Quais são os esforços esperados? Quanto é que vai custar? Quais são os recursos necessários?

- **Consignatários** Digam quem participa. Atribua tarefas ou liderança e responsabilidades da atividade.
- **Problemas** Identifique os problemas principais, dificuldades previstas, riscos, obstáculos, incertezas, pontos de atenção, dependências críticas.
- **Estado** Escreva aqui uma avaliação sintética de como a atividade decorre: saudável? Atrás? Etc.
- **Avaliação global do progresso** A sua própria avaliação do progresso da atividade de alto nível sintética, orientada à gestão.

### 3.4 Configuração automática: usar o recurso de implantação do GGI

A partir da versão 1.1 do Manual, o GGI propõe [My GGI Board](#), uma ferramenta automatizada para implantar a sua própria instância do GGI como um projeto do GitLab. O processo de instalação leva menos de 10 minutos para ser configurado, é totalmente documentado e fornece uma maneira simples e confiável de personalizar as atividades, acompanhar a sua execução à medida que você progride e comunicar os resultados às partes interessadas. Um exemplo ao vivo da implantação pode ser visto no [GitLab da iniciativa](#), com o site gerado automaticamente disponível nas [suas páginas do GitLab](#).



Aqui está um fluxo de trabalho padrão para usar o recurso de implantação:

1. Bifurque a My GGI Board na sua própria instância ou projeto do GitLab e configure-a seguindo as instruções no README do projeto: <https://gitlab.ow2.org/ggi/my-ggi-board>. Isso irá:
  - Criar todas as atividades como problemas no projeto.
  - Criar um bom quadro para ajudá-lo a visualizar e gerir as atividades.
  - Criar um site estático, servido nas páginas da sua instância de GitLab, com as informações extraídas das atividades.
  - Atualizar a descrição do projeto com as ligações apropriadas ao quadro de atividades e o seu site estático.
1. Desde aí, pode começar a rever as atividades e preencher a secção do cartão de pontuação.
  - A secção do cartão de pontuação é o equivalente eletrónico (e simplificado) dos cartões de pontuação em ODT mencionados acima. São usados para adaptar a atividade ao seu contexto, listando os recursos locais, riscos e oportunidades e definindo objetivos personalizados necessários para concluir a atividade.
  - Se alguma atividade não se aplicar ao seu contexto, basta marcá-la como "Não Seleccionada" ou fechá-la.
  - Este é um processo bastante demorado, mas altamente necessário, pois irá ajudá-lo, passo a passo, a definir o seu próprio roteiro e plano.
1. Quando as atividades forem definidas, poderá começar a implementar o seu próprio OSPO. Selecione algumas atividades que considera relevantes para começar e altere o rótulo de progresso de 'Não iniciado' para 'Em andamento'. Pode usar os recursos do GitLab para ajudá-lo a organizar o trabalho (comentários, responsáveis, etc.) ou qualquer outra ferramenta. É fácil de combinar as atividades e há muitas integrações excelentes disponíveis.

2. Regularmente (semanalmente, mensalmente, dependendo do seu horário), avalie e reveja as atividades atuais e, quando forem concluídas, altere o rótulo de 'Em andamento' para 'Concluído'. Selecione alguns outros e comece novamente na etapa 3 até que todos estejam concluídos.

O site oferece uma visão geral rápida das atividades atuais e passadas e extrai a secção do cartão de pontuação de problemas para exibir apenas as informações locais relevantes. Quando ocorrem alterações nos problemas (atividades) estas são atualizadas automaticamente no site gerado. Observe que os pipelines CI para a geração automática do site são executados automaticamente todas as noites, mas você pode iniciá-los facilmente na secção CI/CD do projeto GitLab. A imagem a seguir mostra a interface do site gerada automaticamente.

The screenshot shows a web dashboard for 'My Good Governance Initiative'. At the top, there are navigation links for 'Dashboard' and 'My Board'. The main heading is 'Welcome', followed by a sub-heading 'Current activities' and a link for '[ details ]'. Below this, there is a list of activities with progress bars:

- 17 activities `not_started`
- 4 activities `in_progress`
- 4 activities `done`

The 'Current activities' section is defined as having the label `in_progress`. It lists two activities:

- Open source enabling digital transformation (GGI-A-37)**: Tasks: 1 done / 2 total. Progress bar at 50%.
- Support open source communities (GGI-A-30)**: Tasks: 2 done / 3 total. Progress bar at 66%.

Pode fazer perguntas ou obter suporte para o recurso de implantação na nossa página inicial do GitLab e agradecemos comentários.

Página inicial da implantação da GGI: <https://gitlab.ow2.org/ggi/my-ggi-board>

### 3.5 Desfrutar

Comunique sobre o seu sucesso e desfrute da paz de espírito de uma estratégia de código aberto de última geração!

O OSS Good Governance é um método para implementar um programa de melhoria contínua e assim nunca termina. No entanto, é importante destacar os passos intermédios e apreciar as mudanças que produz, para tornar o progresso visível e partilhar os resultados.

- Comunique com interessados e utilizadores finais para lhes dar a conhecer as vantagens e benefícios que o esforço da iniciativa traz.
- Fomente a sustentabilidade do programa. Assegure que as práticas recomendadas e lições aprendidas com o programa são sempre aplicadas e atualizadas.
- Partilhe a sua experiência com os seus pares: dê opinião ao grupo de trabalho GGI e no seio da sua comunidade OSPO de adoção, e partilhe a sua abordagem.

## 4 Atividades de objetivos de utilização

### 4.1 Inventário de competências e recursos de código aberto

ID da atividade: [GGI-A-17](#).

#### Descrição

Em qualquer fase, de uma perspectiva de gestão, é útil ter um mapeamento, um inventário de recursos de código aberto, bens, utilização e o seu estado, bem como as necessidades potenciais e soluções disponíveis. Inclui também a avaliação do esforço e das competências necessárias para preencher a lacuna.

Esta atividade visa capturar a situação de código aberto numa organização e no mercado e avaliar a ponte entre eles.

- Inventário da utilização de OSS na cadeia de desenvolvimento de software, bem como nos produtos e componentes de software utilizados na produção.
- Identificar tecnologias de código aberto (soluções, estruturas, características inovadoras) que possam servir as suas necessidades e ajudar a melhorar o seu processo.

Não incluído

- Identificar e qualificar ecossistemas e comunidades de OSS relacionados. (Objetivo da cultura)
- Identificar as dependências das bibliotecas e componentes do OSS. (Objetivo da confiança)
- Identificar as competências técnicas (por exemplo, línguas, estruturas...) e habilidades interpessoais (por exemplo, colaboração, comunicação) necessárias. (pertence às próximas atividades: crescimento das competências OSS e competências de desenvolvimento de software de código aberto)

#### Avaliação de oportunidades

Um inventário dos recursos de código aberto disponíveis que ajudará a otimizar o investimento e a dar prioridade ao desenvolvimento de competências.

Esta atividade cria as condições para melhorar a produtividade do desenvolvimento, dada a eficiência e popularidade dos componentes OSS, princípios e ferramentas de desenvolvimento, particularmente no desenvolvimento de aplicações e infraestruturas modernas.

- Isto pode exigir a simplificação do portfólio de recursos do OSS.
- Isso pode exigir o retreinamento do pessoal.
- Isto permite a identificação de necessidades e preenche o seu roteiro de TI.

#### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe uma lista funcional de recursos OSS "Usamos", "Integramos", "Produzimos", "Alojamos", e as competências relacionadas
- Estamos no caminho de melhorar a eficiência através da utilização de métodos e ferramentas de última geração.
- Identificámos recursos de OSS não catalogados até agora (podem ter sido incorporados: temos elementos para definir uma política neste domínio?)
- Solicitamos que novos projetos suportem ou reutilizem recursos de OSS existentes. (Objetivo de Cultura?)
- Temos uma perceção e compreensão razoavelmente segura do alcance da utilização do OSS na nossa organização.

#### Ferramentas

Há muitas formas diferentes de estabelecer tal inventário. Uma forma seria classificar os recursos do SAA em quatro categorias:

- OSS que usamos: software que usamos na produção ou no desenvolvimento

- OSS que integramos: por exemplo, bibliotecas de OSS que integramos numa aplicação feita à medida
- OSS que produzimos: por exemplo, uma biblioteca que publicámos no GitHub ou um projeto de OSS que desenvolvemos ou para o qual contribuímos regularmente.
- OSS que hospedamos: OSS que executamos para oferecer um serviço interno, como um CRM, GitLab, Nexus, etc. Uma tabela de exemplo seria parecida com a seguinte:

Utilizamos	Integramos	Produzimos	Hospedamos	Habilidades
Firefox, OpenOffice, Postgresql	Biblioteca slf4j	Biblioteca YY em GH	GitLab, Nexus	Java, Python

A mesma identificação deve ser aplicada às habilidades

- Habilidades e experiências disponíveis através das equipas existentes
- Habilidades e experiências que poderiam ser desenvolvidas ou adquiridas internamente (treino, coaching, experimentação)
- Habilidades e experiências que precisam ser buscadas no mercado ou por parceria/contratação

### Recomendações

- Manter tudo simples.
- É um exercício de relativamente alto nível, não um inventário detalhado para o departamento de contabilidade.
- Embora esta atividade seja um bom ponto de partida, não é necessário tê-la 100% concluída antes de lançar outras atividades.
- Trate de questões, recursos e habilidades relacionadas ao **desenvolvimento de software** na atividade #42.
- O inventário deve abranger todas as categorias de TI: sistemas operativos, middlewares, SGBD, administração de sistemas, ferramentas de desenvolvimento e teste, etc.
- Comece a identificar comunidades relacionadas: é mais fácil obter apoio e comentário do projeto quando elas já o conhecem.

### Recursos

- Um curso excelente sobre [Free \(/Libre\), e Open Source Software \(FOSS\)](#), pelo Professor Dirk Riehle.

## 4.2 Crescimento de competências de código aberto

ID da atividade: [GGI-A-18](#).

### Descrição

Esta atividade tem a ver com o planeamento e a iniciação de capacidades técnicas e a experiência inicial com o OSS, após a realização de um inventário (#17). É também a oportunidade de começar a estabelecer um roteiro de desenvolvimento de habilidades básicas e leves.

- Identifique quais são as habilidades e o treino necessários.
- Estabeleça um projeto-piloto para iniciar a abordagem, aprenda fazendo, estabeleça um primeiro marco de realização.
- Aproveite das lições aprendidas e construa um corpo de conhecimento.
- Comece a identificar e documentar os próximos passos para uma adoção mais ampla.
- Elabore uma estratégia durante os próximos meses ou um ano para envolver a gestão e o apoio financeiro.

O âmbito da atividade:

- Linux, Apache, Debian, habilidades de administração.
- Bases de dados de código aberto MariaDB, MySQL, PostgreSQL, etc.
- Virtualização de código aberto e tecnologias de nuvem.
- Stack LAMP e as suas alternativas.

### Avaliação de oportunidades

Como qualquer tecnologia da informática, e provavelmente ainda mais, o código aberto traz inovação. O código aberto cresce rapidamente e muda rapidamente. Exige que as organizações se mantenham atualizadas.

Esta atividade ajuda a identificar áreas onde a formação pode ajudar as pessoas a tornarem-se mais eficientes e a sentirem-se mais seguras ao utilizar código aberto. Ajuda os funcionários a tomar decisões de desenvolvimento. Semear competências básicas de código aberto permite avaliar a oportunidade de:

- Ampliar as soluções informáticas com as tecnologias de mercado existentes, desenvolvidas pelo ecossistema.
- Desenvolver novas formas de colaboração dentro e fora da organização.
- Adquirir competências em tecnologias novas e inovadoras.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- Uma matriz de habilidades é desenvolvida.
- O âmbito das tecnologias OSS utilizadas é definido proativamente, isto é, evitando o uso descontrolado das tecnologias OSS.
- Um nível satisfatório de perícia é adquirido para essas tecnologias.
- As equipas receberam uma formação "básicos de código aberto" para começar.

### Ferramentas

Uma ferramenta chave aqui é chamada matriz (ou mapeamento) de atividade (ou competência).

Esta atividade pode ser realizada por:

- utilizar tutoriais online (há muitos gratuitos na Internet),
- participar em conferências de desenvolvedores,
- receber formação de vendedores, etc.

### Recomendações

- A utilização e desenvolvimento de componentes de código aberto de forma segura e eficiente requer uma mentalidade aberta e colaborativa que precisa de ser reconhecida e propagada tanto a partir do topo (gestão) como da base (desenvolvedores).
- Assegure que a abordagem é ativamente apoiada e promovida pela administração. Nada acontecerá se não houver nenhum compromisso por parte da hierarquia.
- Envolver as pessoas (desenvolvedores, intervenientes) no processo: organize mesas redondas e ouça ideias.
- Dê tempo e recursos para que as pessoas descubram, tentem e brinquem com estes novos conceitos. Se possível, faça-o divertido - a gamificação e as recompensas são incentivos bons.

Um projeto-piloto com as etapas seguintes poderia servir de catalisador:

- Identifique a tecnologia ou estrutura com que começar.
- Encontre formação online, tutoriais e código de amostra para experimentar.
- Construa um protótipo da solução final.
- Identifique peritos para desafiar e orientar na implementação.

### Recursos

- [What is a Competency Matrix](#): uma leitura introdutória rápida.
- [How to Make a Skills Matrix for your Team](#): um modelo com comentários.
- [MOOC on Free \(libre\) culture](#) (apenas em francês): este é um curso de 6 partes sobre a cultura livre, introdução aos direitos de autor, propriedade intelectual, licenciamento de código aberto

## 4.3 Supervisão de código aberto

ID da atividade: [GGI-A-19](#).

### Descrição

Esta atividade consiste em controlar a utilização de software de código aberto e assegurar que o software de código aberto é gerido proativamente. Isto diz respeito a várias perspetivas, quer para utilizar ferramentas OSS e soluções empresariais, quer para incluir o OSS como componentes em desenvolvimentos próprios ou modificar uma versão de um software adaptando-o às necessidades próprias, etc. Trata-se também de identificar áreas onde o software de código aberto tornou-se uma solução (por vezes oculta) e avaliar a sua adequação.

Poderá ser necessário esclarecer o seguinte:

- A funcionalidade necessária é fornecida?
- Existe uma funcionalidade adicional fornecida que não é necessária, mas aumenta a complexidade nas fases CONSTRUÇÃO e EXECUÇÃO?
- O que exige a licença, quais são as restrições legais?
- Quanto a decisão torna a sua organização independente do fornecedor?
- Existe uma opção de apoio pronta para as suas necessidades comerciais e quanto custa?
- TCO (Custo Total de Propriedade).
- A administração conhece as vantagens de código aberto, por exemplo, para além da "economia de custos de licença"? Estar confortável com o código aberto ajuda a obter o máximo benefício de trabalhar com as comunidades de projeto e fornecedores.
- Veja se faz sentido partilhar os custos de desenvolvimento a dar os próprios desenvolvimentos e todas as implicações, como o cumprimento da licença, à comunidade.
- Verifique a disponibilidade de apoio à comunidade ou suporte profissional.

### Avaliação de oportunidades

A definição de um processo de decisão especificamente dirigido ao código aberto é uma forma de maximizar benefícios.

- Evita o uso descontrolado e os custos ocultos das tecnologias OSS.
- Conduz a decisões estratégicas e organizacionais informadas e conscientes do OSS.

Custos: a atividade pode desafiar e reconsiderar o uso subideal de código aberto como ineficiente, arriscado, etc.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- O OSS tornou-se uma opção confortável quando escolher OSS não é visto como uma exceção ou uma escolha perigosa.
- OSS tornou-se a opção "corrente dominante".
- Os intervenientes principais estão suficientemente convencidos de que a solução de código aberto tem vantagens estratégicas nas quais vale a pena investir.
- Pode ser demonstrado que o "TCO" da solução baseada no código aberto dá à sua organização um valor mais elevado do que a alternativa.
- Há uma avaliação de como a independência do fornecedor poupa dinheiro ou potencialmente pode poupar dinheiro no futuro.
- Há uma avaliação de que a independência da solução reduz os riscos de ser demasiado cara para alterar a solução ( nenhuns formatos de dados fechados possíveis).

### Ferramentas

Nesta fase não podemos pensar em nenhuma ferramenta relevante ou preocupada com essa atividade.

## Recomendações

- Gerir proativamente a utilização de código aberto requer níveis básicos de consciencialização e compreensão dos fundamentos do código aberto, porque devem ser considerados em qualquer decisão do OSS.
- Compare a funcionalidade necessária em vez de procurar uma alternativa para uma solução de código fechado conhecida.
- Assegure-se de ter apoio e desenvolvimento futuro.
- Considere os efeitos da licença da solução na sua organização.
- Convença todos os atores-chave sobre o valor das vantagens do código aberto, além da "economia de custos de licença".
- Seja honesto, não exagere o efeito da solução de código aberto.
- No processo de tomar decisões é igualmente importante avaliar soluções de código aberto diferentes de modo a evitar o desapontamento através de expectativas erradas, para esclarecer o que a organização é obrigada a fazer e todas as vantagens que a abertura das soluções traz. Isto deve ser identificado para que a organização possa avaliá-lo para o próprio contexto.

## Recursos

- [Top 5 Benefits of Open Source](#): Blog patrocinado, mas ainda interessante, de leitura rápida.
- [Weighing The Hidden Costs Of Open Source](#): uma revisão patrocinada pela IBM sobre os custos de suporte do OSS.

## 4.4 Software de empresa de código aberto

ID da atividade: [GGI-A-20](#).

### Descrição

Esta atividade é sobre selecionar proativamente soluções de OSS, quer sejam apoiadas por vendedores ou pela comunidade, em áreas orientadas a negócios. Pode também abranger a definição de políticas de preferência para a seleção de software de aplicações de código aberto para empresas.

Embora o software de código aberto seja mais frequentemente utilizado por profissionais de TI -- sistemas operativos, middleware, SGBD, administração de sistemas, ferramentas de desenvolvimento -- tem ainda de ser reconhecido em áreas onde profissionais de negócios são os utilizadores principais.

A atividade abrange áreas como: suítes Office, ambientes de colaboração, gestão de utilizadores, gestão de fluxos de trabalho, gestão de relações com clientes, correio eletrónico, comércio eletrónico, etc.

### Avaliação de oportunidades

À medida que o código aberto se torna a corrente dominante, alcança muito além dos sistemas operativos e ferramentas de desenvolvimento e assim cada vez mais encontra o seu caminho às camadas superiores dos sistemas de informação, no cerne das aplicações de negócio. É relevante identificar quais aplicações de OSS são usados com sucesso para atender às necessidades da organização e como podem tornar-se a escolha preferida de uma organização que economiza custos.

A atividade pode trazer alguns custos de retreino e migração.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe uma lista de soluções OSS recomendadas para responder a necessidades pendentes em aplicações comerciais.
- Uma política de preferência para a seleção de software de aplicações de negócios de código aberto é elaborada.



- Aplicações comerciais proprietárias em uso são avaliadas em relação a OSS equivalentes.
- O processo de aquisição e solicitações especificam a preferência de código aberto (se legalmente viável).

### Ferramentas

Ferramentas para mapear software e aplicações de negócios?

Nesta fase, não podemos pensar em nenhuma ferramenta relevante ou em questão com esta atividade.

### Recomendações

- Converse com colegas, aprenda com o que outras empresas comparáveis à sua fazem.
- Visite eventos locais do setor para saber mais sobre as soluções de OSS e apoio profissional.
- Experimente primeiro as edições comunitárias e o apoio comunitário antes de se comprometer com planos de apoio pagos.

### Recursos

- [What is enterprise open source?](#): uma leitura rápida sobre código aberto para empresas.
- [101 Open Source Apps to Help your Business Thrive](#): uma lista indicativa de soluções de código aberto para empresas.

## 4.5 Gerir habilidades e recursos de código aberto

ID da atividade: [GGI-A-42](#).

### Descrição

Esta atividade centra-se nas habilidades e nos recursos do **desenvolvimento de software**. Inclui as tecnologias e habilidades específicas de desenvolvimento dos programadores, bem como os processos, métodos e ferramentas globais de desenvolvimento.

Uma vasta quantidade de documentação, fóruns e discussões decorrentes do ecossistema e recursos públicos está disponível para as tecnologias de código aberto. Para beneficiar plenamente da abordagem de código aberto delas, é necessário estabelecer um roteiro dos seus recursos e alvos desejados atuais para estabelecer um programa consistente de habilidades, métodos e ferramentas de desenvolvimento nas equipas.

**Domínios de aplicação** É necessário estabelecer os domínios onde o programa será aplicado e como irá melhorar a qualidade e eficiência do código e das práticas. Como exemplo, o programa não terá os mesmos benefícios se houver apenas um único programador a trabalhar em componentes de código aberto ou se todo o ciclo de vida do desenvolvimento for otimizado à inclusão das práticas recomendadas de código aberto.

É necessário definir o âmbito a abranger para o desenvolvimento de código aberto: componentes técnicos, aplicações, modernização ou criação de novos desenvolvimentos. São exemplos de práticas de desenvolvimento que podem beneficiar do código aberto:

- Administração de plataformas cloud.
- Aplicações nativas da cloud, como inovar com estas tecnologias.
- DevOps, Integração Contínua / Entrega Contínua.

### Categorias

- Habilidades e recursos necessários para desenvolver software de código aberto: PI, licenciamento, práticas.
- Habilidades e recursos necessários para desenvolver software a utilizar componentes de código aberto, línguas, tecnologias.
- Habilidades e recursos necessários para utilizar métodos e processos de código aberto.

### Avaliação de oportunidades

Ferramentas de código aberto são cada vez mais populares entre criadores de software. Esta atividade aborda a necessidade de evitar a proliferação de ferramentas heterogêneas numa equipa de desenvolvimento. Ajuda a definir uma política neste domínio. Ajuda a otimizar a formação e a construção de experiência. Um inventário de competências é utilizado para recrutamento, formação e planeamento de sucessão no caso de um funcionário-chave abandonar a empresa.

Precisaríamos de uma metodologia para mapear habilidades de desenvolvimento de software de código aberto.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe uma descrição da cadeia de produção de código aberto (a "cadeia logística de software"),
- Existe um plano (ou uma lista de desejos) para a racionalização dos recursos de desenvolvimento,
- Existe um inventário de habilidades que resume as competências, educação e experiência dos criadores atuais,
- Existe uma lista de desejos de formação e um programa que trata das lacunas de habilidades,
- Há uma lista de práticas recomendadas de desenvolvimento de código aberto em falta e um plano para as aplicar.

### Recomendações

- Comece de forma simples, faça a análise e o roteiro crescer consistentemente.
- Ao recrutar, dê uma forte ênfase às habilidades e experiência de código aberto. É sempre mais fácil quando as pessoas já têm conhecimento de código aberto do que treinar pessoas.
- Verifique programas de formação de vendedores de software e escolas de código aberto.

### Recursos

Mais informações:

- Uma introdução a [o que é um Inventário de Habilidades? \(what is a Skills Inventory?\)](#) de Robert Tanner.
- Um artigo sobre habilidades de código aberto: [5 Open Source Skills to Up Your Game and Your Resume](#)

Esta atividade pode incluir recursos e habilidades técnicas como, por exemplo:

- **Linguagens populares** (tais como Java, PHP, Perl, Python).
- **Frameworks de código aberto** (Spring, AngularJS, Symfony) e ferramentas de teste.
- **Métodos de desenvolvimento e práticas recomendadas** de Agile, DevOps e código aberto.

Atividades associadas:

- [Perspetiva de RH GGI-A-28](#)

## 5 Atividades de objetivos de confiança

### 5.1 Gerir a conformidade legal

Activity ID: [GGI-A-21](#).

#### Descrição

As organizações precisam de implementar um processo de conformidade legal para garantir o seu uso e participação em projetos de código aberto.

A gestão madura e profissional do cumprimento da lei, na organização e em toda a cadeia de abastecimento, está em causa:

- Realizar uma análise exaustiva da propriedade intelectual que inclua a identificação da licença e a verificação da compatibilidade.
- Assegurar que a organização possa utilizar, integrar, modificar e redistribuir componentes de código aberto como parte dos seus produtos ou serviços em segurança.
- Proporcionar aos empregados e contratantes um processo transparente sobre como criar e contribuir ao software de código aberto.

*Análise da Composição do Software (SCA):* Uma parte significativa das questões legais e de PI resulta do uso de componentes disponibilizados sob licenças que são incompatíveis entre eles ou incompatíveis com a forma como a organização pretende utilizar e redistribuir os componentes. A SCA é o primeiro passo para resolver essas questões, pois "é preciso conhecer o problema para resolvê-lo". O processo consiste em identificar todos os componentes envolvidos num projeto num documento de lista de material, incluindo dependências de construção e testes.

*Verificação de licenças:* Um processo de verificação de licenças utiliza um instrumento para analisar a base de códigos e identificar as licenças e os direitos de autor dentro da mesma automaticamente. Se executado regularmente e idealmente integrado em cadeias de construção e integração contínuas, isto permite detetar problemas de PI antecipadamente.

#### Avaliação de oportunidades

Com a crescente utilização do OSS nos sistemas de informação de uma organização, é essencial avaliar e gerir a potencial exposição legal.

No entanto, a verificação de licenças e direitos de autor pode ser complicada e dispendiosa. Os programadores precisam de poder verificar a PI e as questões legais rapidamente. Ter uma equipa e um responsável empresarial dedicado a questões de PI e legais assegura uma gestão proativa e consistente das questões legais, ajuda a assegurar o uso de componentes e contribuições de código aberto e fornece uma visão estratégica clara.

#### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe um processo de verificação de licenças fácil de usar para projetos.
- Existe um processo de verificação de PI fácil de usar para projetos.
- Existe uma equipa ou pessoa responsável pelo cumprimento legal na organização.
- Estão previstas auditorias regulares para avaliar a conformidade legal.

Outras formas de estabelecer pontos de verificação:

- Existe um processo de verificação de licenças fácil de usar.
- Existe uma equipa jurídica/PI fácil de usar, como na atividade #13.
- Todos os projetos fornecem a informação necessária para que as pessoas utilizem e contribuam para o projeto.
- Existe um contacto na equipa para questões relacionadas com PI e licenciamento.
- Existe um responsável corporativo dedicado à PI e licenciamento.
- Existe uma equipa dedicada a questões relacionadas com PI e licenciamento.

## Ferramentas

- [ScanCode](#)
- [Fossology](#)
- [SW360](#)
- [Fossa](#)
- [OSS Review Toolkit](#)

## Recomendações

- Informe as pessoas sobre os riscos associados ao licenciamento em conflito com objetivos comerciais.
- Proponha uma solução fácil para os projetos criarem a verificação de licenças no seu código.
- Comunique a sua importância e ajude projetos a adicioná-la aos sistemas de IC deles.
- Forneça um modelo ou orientações oficiais para a estrutura do projeto.
- Instale verificações automatizadas para garantir que todos os projetos cumprem as diretrizes.
- Considere a realização de uma auditoria interna para identificar as licenças da infraestrutura da empresa.
- Forneça formação básica em PI e licenciamento para pelo menos uma pessoa por equipa.
- Forneça formação completa em PI e licenciamento ao oficial.
- Estabelecimento de um processo para levantar questões de PI e licenciamento com o responsável.

Lembre-se que o cumprimento não é apenas legal; é também uma questão de PI. Portanto, eis algumas perguntas para ajudar a compreender as consequências do cumprimento legal:

- Se distribuir um componente de código aberto e não respeitar as condições da licença, infrinjo a licença --> implicações legais.
- Se utilizar um componente de código aberto num projeto que desejo distribuir/publicar, essa licença pode obrigar à visibilidade sobre elementos de código que não quero fazer de código aberto --> Impacto de confidencialidade para a vantagem tática da minha empresa e com terceiros (implicações legais).
- É uma discussão aberta sobre a utilização de uma licença de código aberto para um projeto que pretendo publicar, concedendo direitos de propriedade intelectual relevantes --> implicações de propriedade intelectual.
- Se fizer um projeto de código aberto *antes* de qualquer processo de patentes, *provavelmente* exclui a criação de patentes relativas ao projeto --> implicações de PI.
- Se fizer um projeto open source *após* qualquer processo de patentes, *provavelmente* permite a criação de patentes (defensivas) relativas a esse projeto --> potencial de PI.
- Em projetos complexos que trazem muitos componentes com muitas dependências, a multiplicidade de licenças de código aberto pode exibir incompatibilidades entre licenças --> implicações legais (compare problema #23).

## Recursos

- Existe uma extensa lista de ferramentas na [página de grupo de conformidade OSS existente](#).
- [Recommended Open Source Compliance Practices for the enterprise](#). Um livro de Ibrahim Haddad, da Fundação Linux, sobre práticas de conformidade de código aberto para a empresa. [OpenChain Project](#)

## 5.2 Gerir vulnerabilidades de software

Activity ID: [GGI-A-22](#).

### Descrição

O código de uma pessoa é tão seguro quanto a sua parte menos segura. Casos recentes (por exemplo, [heartbleed<sup>1</sup>](#), [equifax<sup>2</sup>](#)) demonstraram a importância de verificar as vulnerabilidades

<sup>1</sup><https://pt.wikipedia.org/wiki/Heartbleed>

<sup>2</sup><https://arstechnica.com/information-technology/2017/09/massive-equifax-breach-caused-by-failure-to-patch-two-month-old-bug/>

em partes do código que não são diretamente desenvolvidas pela entidade. As consequências da exposição vão desde perdas de dados (com tremendo impacto na reputação) a pedidos de resgate e indisponibilidade de serviços, ameaçando as empresas.

O software de código aberto é conhecido por ter uma melhor gestão de vulnerabilidades do que o software proprietário, principalmente porque:

- Mais olhos procuram e corrigem problemas em códigos e processos abertos.
- Projetos de código aberto corrigem vulnerabilidades e lançam patches e novas versões muito mais rapidamente.

Por exemplo, um [estudo da WhiteSource](#) sobre software proprietário mostrou que 95% das vulnerabilidades encontradas nos seus componentes de código aberto já tinham lançado uma correção na altura da análise. A questão, portanto, é de **gerir melhor as vulnerabilidades tanto no código desenvolvido como nas dependências do mesmo**, independentemente de serem de código fechado ou de código aberto.

Para mitigar estes riscos, é necessário criar um programa de avaliação dos seus ativos de software e um processo de verificação de vulnerabilidade executado regularmente. Implementar ferramentas que alertem as equipas afetadas, gerir as vulnerabilidades conhecidas e prevenir ameaças de dependências de software.

### Avaliação de oportunidades

Qualquer empresa que usa software deve observar as vulnerabilidades dele em:

- a sua infraestrutura (por exemplo, infraestrutura cloud, infraestrutura de rede, armazéns de dados),
- as suas aplicações comerciais (RH, ferramentas de CRM, gestão de dados internos e relacionados com os clientes),
- o seu código interno: por exemplo, o sítio web da empresa, projetos de desenvolvimento interno, etc.,
- e todas as dependências diretas e indiretas de software e serviços.

O RSI das vulnerabilidades é pouco conhecido até que algo de mau aconteça. Há que considerar as consequências de uma grande perda de dados ou indisponibilidade de serviços para estimar o verdadeiro custo das vulnerabilidades.

Da mesma forma, uma cultura de sigilo e ocultação para questões relacionadas com a segurança na empresa deve ser evitada a todo o custo. Em vez disso, a informação sobre o estado de vulnerabilidade deve ser partilhada e discutida para encontrar as melhores respostas por parte das pessoas certas, desde os desenvolvedores até à administração superior.

Os benefícios de prevenir ataques cibernéticos gerindo cuidadosamente as vulnerabilidades de software são bastantes:

- Evitar riscos à reputação,
- Evitar perdas por abuso (DDoS, Ransomware, Tempo para reconstruir um sistema informático alternativo após um ataque),
- Cumprir os regulamentos de proteção de dados.

A gestão de vulnerabilidades de software OSS é apenas uma parte do processo maior de cibersegurança que aborda globalmente a segurança dos sistemas e serviços na organização.

### Avaliação do progresso

Deve haver uma pessoa ou equipa dedicada a monitorizar vulnerabilidades e processos fáceis de usar em que os programadores possam confiar. A avaliação das vulnerabilidades é uma parte padrão do processo de integração contínua e as pessoas conseguem monitorizar o estado atual dos riscos num painel de instrumentos dedicado.

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- A atividade é abrangida quando todo o software e serviços internos são avaliados e monitorizados quanto a vulnerabilidades conhecidas.

- A atividade é abrangida quando uma ferramenta e um processo dedicados são implementados na cadeia de produção de software para evitar a introdução de problemas nas rotinas diárias de desenvolvimento.
- Uma pessoa ou equipa é responsável pela avaliação de exposição a CVEs/riscos de vulnerabilidade.
- Uma pessoa ou equipa é responsável pelo envio de CVE/vulnerabilidade às pessoas interessadas (SysOps, DevOps, programadores, etc.).

### Ferramentas

- Ferramentas do GitHub
  - GitHub fornece diretrizes e ferramentas para assegurar o código alojado na plataforma. Veja [GitHub docs](#) para mais informações.
  - GitHub fornece [Dependabot](#) para identificar vulnerabilidades nas dependências automaticamente.
- [Eclipse Steady](#) é uma ferramenta gratuita e de código aberto que analisa projetos de Java e Python em busca de vulnerabilidades e ajuda os programadores a atenuá-las.
- [OWASP dependency-check](#): um scanner de vulnerabilidades em código aberto.
- [OSS Review Toolkit](#): um orquestrador de código aberto capaz de recolher conselhos de segurança para dependências usadas por serviços de dados de vulnerabilidade configurados.

### Recursos

- A [base de dados de vulnerabilidades](#) dos CVEs. Veja também [base de dados de segurança do NIST](#) do NVD, e recursos de satélite como [CVE Details](#).
- Consulte também esta nova iniciativa da Google: as [vulnerabilidades do código aberto](#).
- O grupo de trabalho do OWASP publica uma lista de scanners de vulnerabilidades [no website dele](#), tanto do mundo comercial como de código aberto.
- J. Williams and A. Dabirsiaghi. The unfortunate reality of insecure libraries, 2012.
- [Detection, assessment and mitigation of vulnerabilities in open source dependencies](#), Serena Elisa Ponta, Henrik Plate & Antonino Sabetta, Empirical Software Engineering volume 25, páginas 3175–3215(2020).
- [A Manually-Curated Dataset of Fixes to Vulnerabilities of open source Software](#), Serena E. Ponta, Henrik Plate, Antonino Sabetta, Michele Bezzi, Cédric Dangremont. Existe também um [conjunto de ferramentas em desenvolvimento para implementar o conjunto de dados em epígrafe](#).

## 5.3 Gerir as dependências de software

Activity ID: [GGI-A-23](#).

### Descrição

Um programa *de identificação de dependências* procura as dependências efetivamente utilizadas no código. O resultado é que a organização deve estabelecer e manter uma lista de dependências conhecidas para o seu código e observar a evolução dos fornecedores identificados.

Estabelecer e manter uma lista de dependências conhecidas é um facilitador e um pré-requisito para:

- Verificação de PI e licenças: algumas licenças não podem ser misturadas, mesmo como dependência. É preciso conhecer as dependências para avaliar os riscos legais associados deles.
- Gestão das vulnerabilidades: todo o software é tão fraco como a sua parte mais fraca: veja o exemplo do [bug Heartbleed](#). É preciso conhecer as suas dependências para avaliar os riscos de segurança que lhe são associados.
- Ciclo de vida e sustentabilidade: uma comunidade ativa no projeto de dependência é um sinal brilhante para correções de bugs, otimizações e novas características.
- Seleção atenciosa de dependências utilizadas, de acordo com critérios de "maturidade" - o objetivo é usar componentes de código aberto seguros, com um código são e bem conservado e uma comunidade viva, ativa e reativa que aceitará contribuições externas, etc.

### Avaliação de oportunidades

A identificação e monitorização de dependências é um passo necessário para mitigar os riscos associados a qualquer reutilização de código. Além disso, a implementação de ferramentas e processos para gerir as dependências de software é um pré-requisito para gerir a qualidade, conformidade e segurança adequadamente.

Considere as questões seguintes:

- Qual é o risco para a empresa (custo, reputação, etc), se o software for corrompido, atacado ou processado?
- O código desenvolvido é considerado crítico para as pessoas, a organização ou o negócio?
- E se um componente de que uma aplicação depende alterar o seu repositório?

O mínimo e primeiro passo é implementar uma ferramenta de análise da composição do software (SCA). O apoio de empresas de consultoria especializadas poderá ser necessário para uma SCA completa ou um mapeamento de dependências.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- As dependências são identificadas em todo o código desenvolvido internamente.
- As dependências são identificadas em todo o código externo executado na empresa.
- Uma análise de composição de software fácil de configurar ou um procedimento de identificação de dependências está disponível para os projetos adicionarem ao seu processo de Integração Contínua.
- São utilizadas ferramentas de análise de dependências.

### Ferramentas

- [OWASP Dependency check \(Verificação de dependências do OWASP\)](#): a verificação de dependências é uma ferramenta de Análise de Composição de Software (SCA) que tenta detetar vulnerabilidades divulgadas publicamente contidas nas dependências de um projeto.
- [OSS Review Toolkit \(Caixa de ferramentas de revisão de OSS\)](#): um conjunto de ferramentas para ajudar na revisão das dependências de software de código aberto.
- [Fossa](#): análise de dependência rápida, portátil e fiável. Suporta a análise de licenças e vulnerabilidades. Agnóstico de linguagem; integra-se em mais de 20 sistemas de construção.
- [Software 360](#).
- [Eclipse Dash license tool](#): leva uma lista de dependências e solicita [ClearlyDefined](#) a verificar as suas licenças.
- [The FOSSology Project](#): FOSSology é um projeto de código aberto com a missão de promover a conformidade com licenças de código aberto.

### Recomendações

- Realize auditorias regulares sobre as dependências e os requisitos de PI para mitigar riscos legais.
- Idealmente, integre a gestão de dependências no processo de integração contínua de modo que questões (nova dependência, incompatibilidade de licença) sejam identificadas e corrigidas o mais rápido possível.
- Mantenha o controlo das vulnerabilidades relacionadas à dependência, mantenha os utilizadores e desenvolvedores informados.
- Informe as pessoas sobre os riscos associados ao licenciamento errado.
- Proponha uma solução fácil para os projetos criarem a verificação de licenças no seu código.
- Comunique a sua importância e ajude projetos a adicioná-la aos sistemas de IC deles.
- Estabeleça um KPI visível sobre riscos relacionados a dependências.

### Recursos

- Página de grupo sobre [ferramentas existentes com licenças de OSS de conformidade com licenças de OSS](#).

- [Free and Open Source Software licence Compliance: Tools for Software Composition Analysis](#), por Philippe Ombredanne, nexB Inc.
- [Software Sustainability Maturity Model](#).
- [CAOS](#): Software de análise da comunidade de código aberto da saúde.

## 5.4 Gerir indicadores chave

Activity ID: [GGI-A-24](#).

### Descrição

Esta atividade recolhe e monitoriza um conjunto de indicadores que informam as decisões de gestão administrativas e as opções estratégicas relativas ao software de código aberto gerido profissionalmente.

As métricas principais relacionadas com software de código aberto constituem o cenário como os programas de boa governação são implementados. A atividade abrange a seleção de alguns indicadores, a publicação às equipas e à administração e o envio regular de atualizações sobre a iniciativa, por exemplo, por um boletim informativo ou notícias empresariais.

Esta atividade requer:

- que partes interessadas discutem e definem os objetivos do programa,
- a implementação de um instrumento de medição e recolha de dados ligado à infra-estrutura de desenvolvimento,
- a publicação de pelo menos um painel para os interessados e para todas as pessoas envolvidas na iniciativa.

Os indicadores são baseados em dados que devem ser recolhidos de fontes relevantes. Felizmente, há muitas fontes para engenharia de software de código aberto. Exemplos incluem:

- o ambiente de desenvolvimento, a cadeia de produção da IC/DC,
- o departamento de RH,
- as ferramentas de teste e de análise da composição do software,
- os repositórios.

Exemplos de indicadores incluem:

- Quantidade de dependências resolvidas, indicadas por tipo de licença.
- Quantidade de dependências desatualizadas/vulneráveis.
- Quantidade de problemas de licenciamento/PI detetados.
- Contribuições feitas a projetos externos.
- Tempo de abertura de bugs.
- Quantidade de contribuintes num componente, quantidade de commits, etc.

Essa atividade trata-se de definir esses requisitos e necessidades de medição e implementar um painel que mostre os principais indicadores do programa de forma simples e eficiente.

### Avaliação de oportunidades

Os indicadores-chave ajudam a compreender e a gerir melhor os recursos dedicados ao software de código aberto e medem os resultados para comunicar eficazmente e desfrutar de todos os benefícios do investimento. Ao comunicar de forma ampla, mais pessoas podem seguir a iniciativa e sentir-se envolvidas, tornando-a, ultimamente, uma preocupação e um objetivo a nível da organização.

Embora cada atividade tenha critérios de avaliação que ajudam a responder a perguntas sobre os progressos alcançados, ainda há necessidade de monitorização feita com números e indicadores quantitativos.

Quer seja numa pequena startup ou numa grande empresa global, as métricas principais ajudam a manter as equipas concentradas e a monitorizar o desempenho. As métricas são cruciais porque apoiam a tomada de decisões e são a base para monitorizar decisões já tomadas.

Com números e gráficos simples e práticos, todos os membros da organização conseguirão acompanhar e sincronizar esforços em relação ao código aberto, a fazê-lo uma preocupação e ação



partilhada. Isto também permite que os vários atores entrem melhor no curso, contribuam para o projeto e obtenham os benefícios globais.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- Uma lista de métricas e como recolhê-las foi estabelecida.
- São utilizadas ferramentas para recolher, armazenar, processar e exibir indicadores.
- Há um painel generalizado disponível para todos os participantes que mostra os progressos feitos na iniciativa.

### Ferramentas

- [GrimoireLab](#) de Bitergia.
- As ferramentas genéricas de IB (Elasticsearch, Grafana, visualizações de R/Python...) são também uma boa opção, quando os conectores adequados são configurados segundo os objetivos definidos.

### Recomendações

- Anote os objetivos e o roteiro da governança de código aberto.
- Comunique internamente sobre as ações e o estado da iniciativa.
- Envolver pessoas na definição dos KPIs para garantir que
  - são bem compreendidos,
  - fornecem uma visão completa das necessidades e
  - são considerados e seguidos.
- Construa pelo menos um painel que possa ser exibido para todos (por exemplo, num ecrã na sala), com indicadores essenciais para mostrar o progresso e a situação geral.

### Recursos

- A [comunidade CHAOSS](#) tem muitas e boas referências e recursos relacionados com indicadores de código aberto.
- Verifique métricas para [atributos de projetos](#) da [metodologia](#) dos Níveis de Preparação para o Mercado do OW2.
- [A New Way of Measuring Openness: The Open Governance Index](#) por Liz Laffan é uma leitura interessante sobre a abertura em projetos de código aberto.
- [Governance Indicators: A Users' Guide]([https://anfrel.org/wp-content/uploads/2012/02/2007\\_UNDP\\_governanceindicators.pdf](https://anfrel.org/wp-content/uploads/2012/02/2007_UNDP_governanceindicators.pdf)) é o guia da ONU sobre indicadores de governação. Embora seja aplicado à democracia, corrupção e transparência das nações, os princípios básicos de medição e indicadores aplicados à governação valem bem a pena ser lidos.

## 5.5 Executar revisões de código

Activity ID: [GGI-A-44](#).

### Descrição

A revisão de código é uma tarefa de rotina que envolve a revisão manual e/ou automatizada do código-fonte de uma aplicação antes de lançar um produto ou entregar um projeto ao cliente. No caso de software de código aberto, a revisão de código é mais que apanhar oportunisticamente erros; é uma abordagem ao nível de equipa integrada no desenvolvimento colaborativo realizado.

As revisões de código devem se aplicar ao código desenvolvido internamente, bem como ao código reutilizado de fontes externas, uma vez que melhora a confiança geral no código e reforça a propriedade. É também uma excelente forma de melhorar as competências e conhecimentos globais na equipa e fomentar a colaboração da equipa.

### Avaliação de oportunidades

As revisões de código são valiosas sempre que a organização desenvolve software ou reutiliza peças de software externas. Apesar de ser uma etapa padrão no processo de engenharia de software, as revisões de código no contexto de software de código aberto são benéficos específicos, tais como:

- Ao publicar o código-fonte interno, verificar se as diretrizes de qualidade adequadas são respeitadas.
- Ao contribuir para um projeto de código aberto existente, verificar se as diretrizes do projeto visado são respeitadas.
- A documentação disponível publicamente é atualizada em conformidade.

É também uma excelente oportunidade para partilhar e aplicar algumas das regras da política de conformidade legal da sua empresa, como, por exemplo:

- Nunca remova cabeçalhos de licenças existentes ou direitos de autor encontrados em código aberto reutilizado.
- Não copie e cole o código-fonte do Stack Overflow sem permissão prévia da equipa jurídica.
- Inclua a linha de direitos autorais correta quando necessário.

As revisões de código trarão confiança ao código. Se as pessoas não estiverem seguras sobre a qualidade ou os riscos potenciais da utilização de um produto de software, devem realizar revisões por pares e de código.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- A revisão do código aberto é reconhecida como um passo necessário.
- Revisões de código aberto são planeadas (regularmente ou em momentos críticos).
- Um processo para a realização de revisões de código aberto foi definido e aceite coletivamente.
- As revisões de código aberto são uma parte padrão do processo de desenvolvimento.

### Recomendações

- A revisão de código é uma tarefa coletiva que funciona melhor num bom ambiente de colaboração.
- Não hesite em utilizar ferramentas e padrões existentes do mundo de código aberto, no qual as revisões de código são uma norma há anos (décadas).

### Recursos

- [What is Code Review?](#): uma leitura didática sobre revisão de código encontrada na Open Practice Library do Red Hat.
- [Best Practices for Code Reviews](#): outra perspetiva interessante sobre o que a revisão de códigos é.

## 6 Atividades do objetivo cultura

### 6.1 Promover práticas recomendadas de desenvolvimento de código aberto

Activity ID: [GGI-A-25](#).

#### Descrição

Esta atividade consiste em definir, promover ativamente e implementar as práticas recomendadas de código aberto no âmbito do equipas de desenvolvimento.

Como ponto de partida, os tópicos seguintes podem ser considerados à atenção:

- Documentação do utilizador e programador.
- Organização adequada do projeto num repositório acessível ao público.
- Promover e implementar a reutilização controlada.
- Fornecimento de uma documentação completa e actualizada do produto.
- Gestão da configuração: fluxos de trabalho de git, padrões de colaboração.
- Gestão de lançamento: lançamento cedo & lançamento frequente, versões estáveis versus versões de desenvolvimento, etc.

Projetos de OSS têm um modus operandi especial do [tipo bazar](#). Para permitir e fomentar esta colaboração e mentalidade, práticas que facilitem o desenvolvimento colaborativo e descentralizado e as contribuições de desenvolvedores de terceiros são recomendadas...

**Documentos comunitários** Assegure que todos os projetos na empresa proponham os documentos seguintes:

- README -- descrição rápida do projeto, como interagir, ligações a recursos.
- Contribuir -- introduções para pessoas dispostas a contribuir.
- Código de Conduta -- o que é aceitável -- ou não -- como se comportar na comunidade.
- LICENSE -- a licença padrão do repositório.

**Práticas recomendadas do REUSE** [REUSE](#) é uma iniciativa da [Free Software Foundation Europe](#) para melhorar a reutilização de software e racionalizar o OSS e o cumprimento de licenças.

#### Avaliação de oportunidades

Embora dependa fortemente do conhecimento comum do OSS entre a equipa, a formação de pessoas e a criação de processos que imponham estas práticas é sempre benéfica. É ainda mais importante quando:

- os utilizadores e contribuintes potenciais não são conhecidos,
- os programadores não estão habituados ao desenvolvimento de código aberto.

#### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- O projeto estabelece uma lista de práticas recomendadas de código aberto a cumprir.
- O projeto monitoriza o seu alinhamento com as práticas recomendadas.
- A equipa de desenvolvimento criou consciência sobre o cumprimento das práticas recomendadas do OSS.
- As práticas recomendadas novas são avaliadas regularmente e um esforço é feito para as implementar.

#### Ferramentas

- A ferramenta [REUSE helper](#) auxilia na realização de um repositório conforme as práticas recomendadas do [REUSE](#). Pode ser incluído em muitos processos de desenvolvimento para confirmar o estado atual.

- [ScanCode](#) consegue listar todos os documentos comunitários e legais no repositório: veja [descrição da características](#).
- GitHub tem uma funcionalidade agradável para [verificar se faltam documentos comunitários](#). Encontra-se na Página do Repositório > "Insights" > "Community".

### Recomendações

- A lista das práticas recomendadas depende do contexto e do domínio do programa e deve ser reavaliada regularmente, para a melhorar continuamente. As práticas devem ser monitorizadas e avaliadas regularmente para rastrear o progresso.
- Forme pessoas sobre a reutilização do OSS (como consumidores) e ecossistemas (como contribuintes).
- Implemente o software REUSE.software como na atividade #14.
- Estabeleça um processo para gerir os riscos legais associados à reutilização e às contribuições.
- Incentive explicitamente as pessoas a contribuírem a projetos externos.
- Forneça um modelo ou orientações oficiais para a estrutura do projeto.
- Instaure verificações automatizadas para garantir que todos os projetos cumprem as diretrizes.

### Recursos

- [OW2's list of open source best practices](#) da metodologia de avaliação dos Níveis de Aptidão para o Mercado.
- [Site oficial do REUSE](#) com especificação, tutorial e FAQ.
- [Guias de código aberto](#).
- Um exemplo de [configuração de práticas recomendadas de gestão utilizando GitHub](#).

## 6.2 Contribuir a projetos de código aberto

Activity ID: [GGI-A-26](#).

### Descrição

Contribuir para projetos de código aberto que são livremente utilizados é um dos princípios-chave da boa governação. O objetivo é evitar ser um simples consumidor passivo e restituir aos projetos. Quando as pessoas acrescentam uma característica ou corrigem um bug para fins próprios, devem torná-lo suficientemente genérico para contribuir ao projeto. Os programadores devem dispor de tempo para contribuir.

Esta atividade abrange o âmbito seguinte:

- Trabalhar com projetos de código aberto a montante.
- Relatar bugs e pedidos de funcionalidades.
- Contribuir com código e correções de bugs.
- Participar em listas de correio da comunidade.
- Partilhar experiências.

### Avaliação de oportunidades

Os benefícios principais desta atividade são:

- Aumenta o conhecimento geral e o compromisso com o código aberto na empresa, à medida que as pessoas começam a contribuir e a envolver-se em projetos de código aberto. Têm um sentimento de utilidade pública e melhoram a própria reputação pessoal.
- A empresa aumenta a sua visibilidade e reputação, à medida que as contribuições atravessam o projeto contribuído. Isto mostra que a empresa está realmente envolvida no código aberto, contribui de volta, e promove a equidade e a transparência.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe um caminho claro e oficial para as pessoas dispostas a contribuir.
- Os promotores são encorajados a contribuir de volta aos projetos de código aberto que utilizam.
- Existe um processo para assegurar a conformidade legal e a segurança das contribuições dos programadores.
- KPI: Volume de contribuições externas (código, listas de correio, edições...) por indivíduo, equipa ou entidade.

### Ferramentas

Pode ser útil seguir as contribuições, tanto para acompanhar o que é contribuído como para poder comunicar sobre o esforço da empresa. Painéis de controlo e software de localização de actividades podem ser utilizados para este fim. Verifique:

- [GrimoireLab](#) do Bitergia
- [ScanCode](#)

### Recomendações

Encorajar as pessoas na entidade a contribuir para projetos externos, ao:

- Dar-lhes tempo para escrever correções e características genéricas e bem testadas de bugs e para os contribuir de volta à comunidade.
- Fornecer formação às pessoas sobre como contribuir de volta às comunidades de código aberto. Trata-se tanto de competências técnicas (melhorar os conhecimentos da sua equipa) como da comunidade (pertencentes às comunidades de código aberto, código de conduta, etc.).
- Fornecer formação sobre questões jurídicas, de PI, técnicas e estabelecer um contacto na empresa para ajudar com estes tópicos, caso alguém tenha dúvidas.
- Fornecer incentivos para trabalhos publicados.
- Note que as contribuições da empresa/entidade reflectirão a qualidade do código e o envolvimento dela, por isso certifique-se de que a sua equipa de desenvolvimento fornece um código que seja suficientemente bom.

### Recursos

- A iniciativa [CHAOSS](#) da Linux Foundation tem ferramentas e indicações sobre como acompanhar as contribuições em desenvolvimento.

## 6.3 Pertencer à comunidade de código aberto

Activity ID: [GGI-A-27](#).

### Descrição

Esta atividade trata-se de desenvolver um maior sentimento de pertença a uma comunidade de código aberto entre os programadores. Como em qualquer comunidade, pessoas e entidades têm de participar e contribuir de volta para o coletivo. Reforça as ligações entre profissionais e traz sustentabilidade e atividade ao ecossistema. Numa vertente mais técnica, permite a escolha de prioridades e do roteiro dos projetos, melhora o nível de conhecimento geral e a consciência técnica.

Esta atividade abrange o seguinte:

- **Identificar eventos** que valham a pena participar. Conectar pessoas, aprender sobre novas tecnologias e construir uma rede são fatores-chave para obter todos os benefícios do software livre.

- Ponderar **associações a fundações\***. As fundações e organizações de código aberto são uma componente chave do ecossistema de código aberto. Fornecem recursos técnicos e organizacionais a projetos e são um bom lugar neutro para os patrocinadores discutirem questões e soluções comuns, ou trabalharem em normas.
- Observar **grupos de trabalho\***. Os grupos de trabalho são um espaço de trabalho colaborativo neutro onde os peritos interagem num domínio específico como a Internet das Coisas, a modelização ou a ciência. São um mecanismo muito eficiente e rentável para abordar preocupações comuns em conjunto, apesar de específicas do domínio.
- **Participação no orçamento\***. No final da viagem, o dinheiro é o facilitador. Planear as despesas necessárias, dar tempo às pessoas pagas para estas atividades, antecipar os próximos movimentos, para que o programa não tenha de parar após alguns meses sem financiamento.

### Avaliação de oportunidades

O código aberto funciona melhor quando feito com a comunidade de código aberto em geral. Facilita a correção de bugs, partilha de soluções, etc.

É também uma boa forma de as empresas mostrarem o seu apoio aos valores do código aberto. A comunicação sobre o envolvimento da empresa é importante tanto para a reputação da empresa como para o ecossistema de código aberto.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- É elaborada uma lista de eventos a que as pessoas poderiam assistir.
- Há monitorização dos discursos públicos dados pelos membros da equipa.
- As pessoas podem enviar solicitações de participação em eventos.
- As pessoas podem apresentar solicitações de participação em eventos.

### Recomendações

- Pesquise as pessoas para conhecer que eventos gostam ou seriam as mais benéficas para o trabalho.
- Crie comunicação interna (boletim informativo, centro de recursos, convites...) para que as pessoas conheçam as iniciativas e possam participar.
- Certifique-se de que essas iniciativas podem beneficiar vários tipos de pessoas (programadores, administradores, suporte...), não apenas a administração superior.

### Recursos

- [What motivates a developer to contribute to open source software?](#) Um artigo de Michael Sweeney no clearcode.cc.
- [Why companies contribute to open source](#) Um artigo de Velichka Atanasova da VMWare.
- [Why your employees should be contributing to open source](#) Uma boa leitura por Robert Kowalski de CloudBees.
- [7 ways your company can support open source](#) Um artigo de Simon Phipps para a InfoWorld.
- [Events: the life force of open source](#) Um artigo de Donna Benjamin da RedHat.

## 6.4 Perspetiva do RH

Activity ID: [GGI-A-28](#).

### Descrição

A mudança à cultura de código aberto tem impactos profundos nos RH:

- **Novos processos e contratos:** contratos têm de ser adaptados para permitir e promover contribuições externas. Isto inclui questões de PI e licenciamento para o trabalho realizado na empresa, mas também a capacidade do empregado ou contratante ter os seus próprios projetos.

- **Tipos de pessoas diferentes:** pessoas que trabalham com código aberto têm frequentemente incentivos e mentalidades diferentes das pessoas puramente proprietárias e empresariais. Processos e mentalidades precisam de se adaptar a este paradigma orientado para a reputação da comunidade, de modo a atrair novos tipos de talento e a mantê-los ao longo do tempo.
- **Desenvolvimento de carreira:** necessidade de oferecer um percurso profissional que alimente e valorize empregados pelas suas habilidades técnicas e interpessoais, bem como as competências esperadas pela sua organização (colaboração para impulsionar os esforços da comunidade, comunicação para atuar como porta-voz da sua empresa, etc.). Por todos os meios, o RH tem um papel fundamental na viabilização do código aberto como um objetivo cultural.

**Força de trabalho** Para um programador que trabalha há muito tempo na mesma solução proprietária, a mudança para o código aberto pode parecer uma grande mudança e requerer adaptação. Mas para a maioria dos programadores, o software de código aberto traz apenas benefícios.

Os desenvolvedores que saem da escola ou da universidade hoje sempre trabalharam com código aberto. Numa empresa, a maioria dos desenvolvedores usa idiomas de código aberto e importam bibliotecas ou trechos de código aberto diariamente. Na verdade, é muito mais fácil colar linhas de código aberto num programa do que acionar o processo de aprovisionamento interno, que se intensifica por múltiplas validações através da linha de gestão.

O código aberto torna o trabalho do programador mais interessante porque com o código aberto, um programador está sempre atento a descobrir o que os pares fora da empresa inventaram e por isso permanece na vanguarda da tecnologia.

Para uma organização, tem de haver uma estratégia de RH para 1/ qualificar ou requalificar a força de trabalho existente, 2/ refletir e posicionar a empresa na contratação de novos talentos, qual é a atratividade da empresa quando se trata de código aberto.

Obter pessoas com uma boa mentalidade de FLOSS, que já entendem o código e sabem como trabalhar bem com outros é maravilhoso. A alternativa de evangelizar/treinar/estagiar vale a pena fazer, mas é mais cara e demorada.

— CEO do fornecedor de software OSS

Isto ilustra que contratar pessoas com conhecimento de código aberto é um caminho de aceleração a considerar na estratégia de RH.

### Processos

- Estabelecer ou rever descrições de funções (competências técnicas, habilidades interpessoais, competências e experiências)
- Programas de treino: auto-treino, treino formal, coaching de gestão, mapeamento de pares, comunidades
- Estabelecer ou rever o percurso profissional: competências, resultados principais/impacto e etapas da carreira

### Avaliação de oportunidades

1. Quadro de práticas de desenvolvimento: o problema provavelmente não é tanto a incentivar os programadores a utilizar mais código aberto, mas para garantir que o utilizam com segurança, conforme os termos de licenciamento de cada tecnologia de código aberto e sem abandonar as verificações de segurança tradicionais (código aberto pode conter códigos maliciosos),
2. Revir as práticas de colaboração: com práticas de desenvolvimento, a oportunidade é alargar a agilidade e colaboração a outras linhas de negócio na sua organização. O fornecimento interno é frequentemente utilizado para fomentar estes comportamentos, embora isto possa ser metade do caminho para a cultura de código aberto,
3. Cultura da organização: no final, trata-se da cultura da sua organização: o código aberto pode ser a estandarte de valores como a abertura, a colaboração, a ética, a sustentabilidade.

## Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- É disponibilizada formação para apresentar tanto os benefícios como as limitações (conformidade com os termos de licenciamento de propriedade intelectual) relacionados ao código aberto.
- Cada programador, cada arquiteto, cada líder de projeto (ou proprietário do produto/proprietário da empresa), compreende os benefícios e as limitações (conformidade com os termos de licenciamento de propriedade intelectual) relacionados ao código aberto.
- Programadores são encorajados a contribuir às comunidades de código aberto e a assumir a responsabilidade por eles e poderiam receber formação adequada para o fazer.
- Habilidades e competências são refletidas em descrições de trabalho de organização e etapas de carreira.
- A experiência adquirida em código aberto pelos programadores (contribuições às comunidades de código aberto, participação no processo de conformidade interno, pessoas externas a falar em nome da empresa, ...) é tida em conta no processo de avaliação de RH.

## Ferramentas

- Matriz de habilidades.
- Programas públicas de formação (p.ex. open source school).
- Aprovisionamento: GitHub, GitLab, LinkedIn, Meetups, Epitech, Epita, ...
- Modelos de contrato (cláusula de lealdade).
- Descrições de trabalho (modelos) & etapas da carreira (modelos).

## Recomendações

Na maioria das vezes, os programadores já conhecem alguns princípios de código aberto e estão dispostos a trabalhar com, e em, software de código aberto. No entanto, ainda existem algumas ações que a administração deve tomar:

- Preferência pela experiência OSS na contratação, mesmo que o trabalho para o qual o programador é contratado esteja relacionado apenas a tecnologia proprietária. É provável que, com a transformação digital, o programador tenha que trabalhar em código aberto eventualmente.
- Programa de treino de OSS: cada programador, cada arquiteto, cada líder de projeto (ou proprietário de produto/proprietário de negócios), deve ter acesso a recursos de treino (treino por vídeos ou presencial) que apresentem os benefícios do código aberto e também as restrições em termos de propriedade intelectual e conformidade com o licenciamento.
- O treino deve ser disponibilizado para programadores que desejam contribuir para comunidades de código aberto e fazer parte dos órgãos de governança dessas comunidades (certificações Linux).
- Reconhecimento nos processos de avaliação pessoal de RH da contribuição do colaborador (programador ou arquiteto) para temas relacionados a código aberto, como contribuições para comunidades de código aberto e cumprimento dos termos de licenciamento de propriedade intelectual. A maioria dos tópicos são compartilhados e encaixam-se em caminhos técnicos de carreira, enquanto alguns podem ou devem ser específicos.
- Segredo melhor guardado e postura da empresa: necessidade de abordar os aspetos de comunicação (como é essencial para a sua organização que possa ser refletido no seu relatório anual), como isso tem impacto na sua postura de comunicação (um colaborador de código aberto pode ser uma pessoa que fala pela sua empresa, incluindo contactos com a imprensa).

## Recursos

- Quanto à capacidade das pessoas de falarem fora da empresa em eventos, consulte a atividade 31: "(objetivo de envolvimento) Afirmar publicamente a utilização de código aberto".

## 6.5 Primeiro a montante

Activity ID: [GGI-A-39](#).



## Descrição

Esta atividade trata-se de desenvolver a consciencialização em relação aos benefícios de contribuir de volta e aplicar o princípio de montante primeiro.

Com a primeira abordagem a montante, todo o desenvolvimento num projeto de código aberto deve ser feito com o nível de qualidade e abertura necessários para serem submetidos aos desenvolvedores principais de um projeto e publicados por eles.

## Avaliação de oportunidades

Escrever código com uma mentalidade de montante em primeiro:

- código de melhor qualidade,
- código que está pronto para ser submetido a montante,
- código mesclado no software principal,
- código que será compatível com a versão futura,
- reconhecimento pela comunidade do projeto e uma cooperação melhor e mais rentável.

Montante primeiro é mais que apenas "ser gentil". Significa ter algo a dizer no projeto. Significa previsibilidade. Significa que está no controlo. Significa que age em vez de reagir. Significa que compreende o código aberto. ([Maximilian Michels](#))

## Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade: o montante primeiro está implementado?

- Aumento significativo na quantidade de solicitações de pull/merge submetidas a projetos de terceiros.
- Uma lista de projetos de terceiros foi elaborada para os quais primeiro a montante devem ser aplicados.

## Recomendações

- Identifique programadores com mais experiência em interagir com programadores a montante.
- Facilite a interação entre programadores e programadores principais (eventos, hackathons, etc.)

## Recursos

- A explicação clara do princípio montante primeiro e porque se encaixa no objetivo da cultura: <https://maximilianmichels.com/2021/upstream-first/>.

Montante primeiro significa que sempre que resolver um problema na sua cópia do código a montante do qual outros podem beneficiar, contribui com essas alterações de volta a montante, ou seja, envia um patch ou abre um pull request para o repositório a montante.

- [What is Upstream and Downstream in Software Development?](#) Uma explicação clara.
- Explicado dos documentos de desenho do Chromium OS: [Upstream First](#).
- Red Hat sobre a montante e as vantagens de [upstream first](#).

## 7 Atividades de objetivos de envolvimento

### 7.1 Envolver-se em projetos de código aberto

Activity ID: [GGI-A-29](#).

#### Descrição

Esta atividade trata-se de cometer contribuições significativas para alguns projetos do OSS que são importantes para si. As contribuições são aumentadas e comprometidas a nível da organização (não a nível pessoal como em #26). Podem assumir várias formas, desde o financiamento direto à alocação de recursos (por exemplo, pessoas, servidores, infraestruturas, comunicação, etc.), desde que beneficiem o projeto ou o ecossistema de forma sustentável e eficiente.

Esta atividade é um seguimento da atividade #26 e traz as contribuições dos projetos de código aberto ao nível da organização, fazendo-as mais visíveis, poderosas e benéficas. As contribuições nesta actividade devem trazer uma melhoria substancial e a longo prazo ao projeto de OSS: por exemplo, um programador ou uma equipa que desenvolve uma nova característica muito desejada, bens de infraestrutura, servidores para um novo serviço, assunção da manutenção de um ramo amplamente utilizado.

A intenção é de reservar uma percentagem dos recursos para patrocinar programadores de código aberto que escrevem e mantêm bibliotecas ou projetos que utilizamos.

Esta atividade implica ter um mapeamento do software de código aberto utilizado e uma avaliação da criticidade dele para decidir qual deles apoiar.

#### Avaliação de oportunidades

Se todas as empresas que utilizam código aberto contribuíssem pelo menos um pouco, teríamos um ecossistema saudável. <https://news.ycombinator.com/item?id=25432248>

Apoiar projetos ajuda a assegurar a sustentabilidade dos mesmos e proporciona acesso à informação, podendo assim mesmo ajudar a influenciar e priorizar alguns desenvolvimentos (embora esta não deveria ser a razão principal para apoiar projetos).

Benefícios potenciais desta atividade: assegurar que os relatórios de bugs sejam priorizados e que os desenvolvimentos sejam integrados na versão estável. Custos possíveis associados à atividade: dedicação de tempo aos projetos, dedicação de dinheiro.

#### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- O projeto beneficiário foi identificado.
- Opção de apoio foi decidida, tal como contribuição monetária direta ou contribuição em código.
- O líder de tarefa foi nomeado.
- Aconteceu alguma contribuição.
- O resultado da contribuição foi avaliado.

Pontos de verificação emprestados do formulário [auto-certificação da OpenChain](#):

- Temos uma política de contribuição para projetos de código aberto em nome da organização.
- Temos um procedimento documentado que rege as contribuições de código aberto.
- Temos um procedimento documentado para sensibilizar todo o pessoal de software para a política de contribuições de código aberto.

#### Ferramentas

Algumas organizações oferecem mecanismos para financiar projetos de código aberto (poderia ser conveniente se o seu projeto alvo estiver nos portfólios delas).

- [Open Collective](#).

- [Software Freedom Conservancy](#).
- [Tidelift](#).

### Recomendações

- Concentre-se nos projetos que são críticos para a organização: estes são os projetos que mais deseja ajudar com as suas contribuições.
- Dirija-se a projetos comunitários.
- Esta atividade requer um mínimo de conhecimentos sobre um projeto alvo.

### Recursos

- [How to support open source projects now](#): uma breve página com ideias sobre o financiamento de projetos de código aberto.
- [Sustain OSS: a space for conversations about sustaining open source](#)

## 7.2 Apoiar comunidades de código aberto

Activity ID: [GGI-A-30](#).

### Descrição

Esta atividade tem a ver com o envolvimento de representantes institucionais do mundo do código aberto.

É conseguida através de:

- Adesão a fundações de OSS (incluindo o custo financeiro da adesão).
- Apoiar e defender as atividades das fundações.

Esta atividade envolve a atribuição de tempo e orçamento às equipas de desenvolvimento e TI para participarem em comunidades de código aberto.

### Avaliação de oportunidades

Comunidades de código aberto estão na vanguarda da evolução do ecossistema de código aberto. O envolvimento nas comunidades de código aberto tem várias vantagens:

- ajuda a estar informado e em dia,
- melhora o perfil da organização,
- a adesão vem com benefícios,
- fornece estruturas adicionais e motivação para a equipa de TI de código aberto.

Os custos incluem:

- taxas de adesão,
- tempo de pessoal e orçamento de viagem atribuído para participar em actividades comunitárias,
- controlo do compromisso de PI.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- A organização é um membro assinado de uma fundação de código aberto.
- A organização contribui para a governação.
- O software desenvolvido pela organização é submetido a / foi adicionado à base de códigos de uma fundação.
- A adesão é reconhecida tanto nos sítios web da organização como na comunidade.
- Uma avaliação de custo/benefício da adesão foi feita.
- Um contato para a comunidade foi nomeado.

**Recomendações**

- Junte-se a uma comunidade compatível com o seu tamanho e recursos, ou seja, uma comunidade que possa ouvir o que diz e onde possa ser um colaborador reconhecido.

**Recursos**

- Veja esta [página útil](#) da Fundação Linux sobre o porquê e como aderir a uma comunidade de código aberto.

**7.3 Afirmar publicamente a utilização de código aberto**

Activity ID: [GGI-A-31](#).

**Descrição**

Esta atividade consiste em reconhecer a utilização do OSS num sistema de informação, em aplicações e em produtos novos.

- Fornecer histórias de sucesso.
- Apresentar em eventos.
- Financiamento da participação em eventos.

**Avaliação de oportunidades**

É agora geralmente aceite que a maioria dos sistemas de informação funcionam no OSS, e que as novas aplicações são, na maioria, feitas através da reutilização do OSS.

O benefício principal desta atividade é de criar condições equitativas entre o OSS e o software proprietário, para garantir que o OSS é alvo de igual atenção e gerido de forma tão profissional como o software proprietário.

Um benefício secundário é que ajuda muito a elevar o perfil do ecossistema OSS e, uma vez que os utilizadores do OSS são identificados como "inovadores", também aumenta a atratividade da organização.

**Avaliação do progresso**

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- Os vendedores comerciais de código aberto recebem autorização para utilizar o nome da organização como cliente de referência.
- Contribuidores podem fazê-lo e expressar-se sob o nome da organização.
- A utilização do OSS é abertamente mencionada no relatório anual do departamento de TI.
- Não há nenhum obstáculo para a organização explicar o seu uso de OSS na média (entrevistas, OSS e eventos do setor, etc.).

**Recomendações**

- O objetivo desta atividade não é que a organização se torne um organismo de ativismo do OSS, mas que se certifique de que não existe nenhum obstáculo para que o público reconheça a utilização dela do OSS.

**Recursos**

- Exemplo de [CERN](#) que afirma publicamente a utilização do OpenStack deles

**7.4 Envolver-se com vendedores de código aberto**

Activity ID: [GGI-A-33](#).

**Descrição**

Proteja contratos com fornecedores de código aberto que lhe fornecem software crítico para si. As empresas e entidades que produzem software de código aberto precisam prosperar para fornecer manutenção e desenvolvimento de novas funcionalidades. As competências específicas destas empresas são necessárias para o projeto e a comunidade de utilizadores conta com a continuidade dos negócios e contribuições dos mesmos.

O envolvimento com vendedores de código aberto assume várias formas:

- Subscrição de planos de apoio.
- Contratação de empresas de serviços locais.
- Patrocínio de desenvolvimentos.
- Pagar uma licença comercial.

Esta atividade implica considerar projetos de código aberto como produtos completos pelo qual vale a pena pagar, como qualquer outro produto proprietário - embora normalmente muito menos dispendioso.

**Avaliação de oportunidades**

O objetivo desta atividade é de assegurar o apoio profissional do software de código aberto utilizado na organização. Tem várias vantagens:

- Continuidade do serviço através de correções de bugs rápidas.
- Desempenho do serviço através de uma instalação otimizada.
- Clarificação do estado jurídico/comercial do software utilizado.
- Acesso antecipado à informação.
- Previsão de orçamento estável.

O custo é obviamente o dos planos de apoio escolhidos. Outros custos podem ser abandonar a subcontratação a granel a grandes integradores de sistemas em favor de contratos de granulação fina com PME's especializadas.

**Avaliação do progresso**

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- O código aberto utilizado na organização é secundado por apoio comercial.
- Planos de suporte para alguns projetos de código aberto foram contratados.
- O custo dos planos de apoio de código aberto é uma entrada legítima no orçamento de TI.

**Recomendações**

- Encontre, sempre que possível, PME especializadas locais.
- Cuidado com os grandes integradores de sistemas que revendem conhecimentos especializados de terceiros (planos de apoio à revenda realmente fornecidos por PME's especializadas de código aberto).

**Recursos**

Algumas ligações que ilustram a realidade comercial do software de código aberto:

- [An investor's view of the community to business evolution of open source projects.](#)
- [A quick read to understand commercial open source.](#)

**7.5 Política de provisionamento de código aberto**

Activity ID: [GGI-A-43](#).

## Descrição

Esta atividade trata-se de implementar um processo de seleção, aquisição, compra de software e serviços de código aberto. Trata-se também de considerar o custo real do software de código aberto e o provisionamento para o mesmo. OSS pode ser “gratuito” à primeira vista, mas não é isento de custos internos e externos, tais como integração, formação, manutenção e apoio.

Tal política exige que tanto as soluções de código aberto como as soluções proprietárias sejam simetricamente consideradas ao avaliar a relação custo-benefício como a combinação ótima do custo total de propriedade e qualidade. Por esse motivo, o departamento de aquisição de TI deve considerar de forma ativa e justa as opções de código aberto, assegurando simultaneamente que as soluções proprietárias sejam consideradas numa base de igualdade nas decisões de compra.

A preferência pelo código aberto pode ser explicitamente declarada com base na flexibilidade inerente à opção de código aberto quando não há diferença significativa de custo global entre soluções proprietárias e de código aberto.

Os departamentos de provisionamento devem compreender que empresas que oferecem apoio ao OSS tipicamente não dispõem dos recursos comerciais para participar em concursos de provisionamento, e adaptar as políticas e processos de provisionamento de código aberto deles em conformidade.

## Avaliação de oportunidades

Várias razões justificam os esforços para estabelecer políticas específicas de provisionamento de código aberto:

- O fornecimento de software e serviços comerciais de código aberto está a crescer e não pode ser ignorado, e requer a implementação de políticas e processos de aquisição dedicados.
- Existe uma variedade crescente de soluções comerciais de código aberto altamente competitivas para sistemas de informação empresarial.
- Mesmo após a adoção de uma componente OSS gratuita e integração dela numa aplicação, recursos internos ou externos devem ser fornecidos para manter esse código fonte.
- O Custo Total de Propriedade (TCO) é frequentemente (embora não necessariamente) mais baixo para soluções de FOSS: sem taxas de licença a pagar na compra/atualização, mercado aberto para fornecedores de serviços, opção de fornecer você mesmo uma parte ou a totalidade da solução.

## Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- Novas aberturas de concurso solicitam proactivamente a apresentação de propostas de código aberto.
- O departamento de provisionamento tem uma forma de avaliar soluções de código aberto versus soluções proprietárias.
- Um processo simplificado de aquisição de software e serviços de código aberto foi implementado e documentado.
- Um processo de aprovação baseado na perícia inter-funcional foi definido e documentado.

## Recomendações

- “Assegure-se de que utiliza os conhecimentos especializados das suas TI, DevOps, cibersegurança, gestão de riscos e equipas de provisionamento ao criar o processo”. (de [5 Open Source Procurement Best Practices](#)).
- Leis de concorrência podem exigir que o “código aberto” não seja especificamente mencionado.
- Selecione a tecnologia antecipadamente e acesse as aberturas de concurso para serviços de personalização e suporte.

### **Recursos**

- [Decision factors for open source software procurement](#): não novo, mas ainda uma grande leitura pelos nossos colegas da OSS-watch no Reino Unido. Veja as [slides](#).
- [5 Open Source Procurement Best Practices](#): uma obra recente sobre o provisionamento de código aberto com dicas úteis.

## 8 Atividades de objetivos de estratégia

### 8.1 Estabelecer uma estratégia para a governação empresarial de código aberto

Activity ID: [GGI-A-16](#).

#### Descrição

A definição de uma estratégia de alto nível para a governação de código aberto numa empresa assegura a consistência e a visibilidade das abordagens tanto para a utilização interna como para as contribuições e o envolvimento externos. Faz a comunicação da empresa mais eficaz, oferecendo uma visão e liderança claras e estabelecidas.

A mudança para o código aberto implica inúmeros benefícios, bem como algumas tarefas e uma mudança na cultura da empresa. Pode ter um impacto nos modelos de negócio e influenciar a forma como uma organização apresenta o valor e oferta dela e na posição em relação aos clientes e concorrentes.

Esta atividade inclui as seguintes tarefas:

- Criar um oficial do OSS, com patrocínio e apoio da direcção (de topo).
- Estabelecer e publicar um roteiro claro para o código aberto, com objetivos declarados e benefícios esperados.
- Assegurar que toda a gestão de alto nível o conhece e atua de acordo com ele.
- Promover o OSS na empresa: encorajar as pessoas a utilizá-lo, fomentar iniciativas internas e nível de conhecimento.
- Promover o OSS fora da empresa: através de declarações e comunicação oficiais, e envolvimento visível nas iniciativas do OSS.

A definição, publicação e aplicação de uma estratégia clara e consistente também ajuda à adesão de todas as pessoas na empresa e facilita outras iniciativas das equipas.

#### Avaliação de oportunidades

É uma boa altura para trabalhar nesta actividade se:

- Não houver um esforço coordenado por parte da administração e o código aberto ainda for visto como uma solução improvisada.
- Já existem iniciativas internas, mas que não penetram até aos níveis superiores da administração.
- A iniciativa foi iniciada há algum tempo, mas enfrenta muitos obstáculos e ainda não produz os resultados esperados.

#### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe uma carta clara de governação de código aberto para a empresa. A carta deve conter:
  - o que alcançar,
  - por quem o fazemos,
  - qual o poder do(s) estratega(s) e o que não é.
- Um roteiro de código aberto está amplamente disponível e é aceite em toda a empresa.

#### Recomendações

- Estabeleça um grupo de pessoas e processos para definir e monitorizar a governação de código aberto na empresa. Certifique-se de que haja um compromisso claro da gestão de alto nível com as iniciativas de código aberto.
- Comunique sobre a estratégia de código aberto na organização, faça-o uma grande preocupação e um verdadeiro compromisso empresarial.
- Assegure que o roteiro e a estratégia são bem compreendidos por todos, desde as equipas de desenvolvimento até ao pessoal da administração e infraestrutura.



- Comunique sobre progressos, para que as pessoas saibam onde a organização está relativamente aos compromissos feitos. Publique atualizações e indicadores frequentemente.

### Recursos

- [Lista de verificação e referências para a governação aberta.](#)
- [L'open source comme enjeu de souveraineté numérique, by Cédric Thomas, OW2 CEO, Workshop nos Orange Labs, Paris, 28. janeiro 2020 \(em francês\).](#)
- [A series of guides to manage open source within the enterprise, by the Linux Foundation.](#)
- [A fine example of open source strategy document, by the LF Energy group](#)

## 8.2 Consciência da administração superior

Activity ID: [GGI-A-34](#).

### Descrição

A iniciativa de código aberto da organização só produzirá benefícios estratégicos se for aplicada ao nível mais elevado, integrando o ADN de código aberto na estratégia da empresa e no trabalho interno. Tal compromisso não pode acontecer se os executivos ao nível mais elevado e a administração superior não fizerem parte dela. A formação e a mentalidade de código aberto devem também ser aplicadas àqueles que moldam as políticas, decisões e estratégia global, tanto dentro como fora da empresa.

Este compromisso assegura que melhorias práticas, mudanças de mentalidade e novas iniciativas sejam cumpridas com um apoio consistente da hierarquia, benevolente e sustentável, introduzindo uma participação mais fervorosa dos trabalhadores. Molda a forma como os actores externos vêem a organização, introduzindo benefícios em termos de reputação e ecossistema. É também um meio de estabelecer a iniciativa e seus benefícios a médio e longo prazo.

### Avaliação de oportunidades

Esta atividade torna-se essencial se/quando:

- A organização definiu objetivos globais relevantes para a gestão de código aberto, mas luta para os atingir. É improvável que a iniciativa possa conseguir alguma coisa sem um bom conhecimento e um compromisso claro por parte dos executivos de nível elevado.
- A iniciativa já começou e está a progredir, mas os níveis superiores da hierarquia não lhe dão o seguimento adequado.

Esperemos que venha a ser evidente que tudo menos o uso ad hoc do código aberto requer uma abordagem consistente e bem pensada, devido à variedade de equipas e à mudança cultural que pode trazer.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe um departamento/oficial de governação mandatado para definir uma estratégia uniforme de código aberto para toda a empresa e assegura que o âmbito é claro.
- Existe um compromisso claro e obrigatório da hierarquia para com a estratégia do OSS.
- Há uma comunicação transparente por parte da hierarquia sobre o compromisso dela com o programa.
- A hierarquia está disponível para discutir o software de código aberto. Pode ser solicitada e desafiada nas promessas que fez.
- Existe um orçamento e financiamento apropriados para a iniciativa.

### Recomendações

Exemplos de ações associadas a esta actividade incluem:

- Realizar formação para desmistificar a gestão do OSS para a administração superior.
- Obter apoio explícito e prático para a utilização e estratégia do OSS.

- Mencionar explicitamente e endossar o programa OSS em comunicações internas.
- Mencionar explicitamente e endossar o programa OSS em comunicações públicas.

Open source is a *strategic enabler* that embarks *enterprise culture*. What does this mean?

- O código aberto pode ser alavancado como um mecanismo para incomodar os fornecedores e reduzir os custos de aquisição de software.
  - Should open source come under the purview of *Software Asset Managers* or *purchasing departments*?
- Open source licences enshrine the freedoms that deliver the benefits of open source, but they also carry *obligations*. If not met appropriately, responsibilities can create legal, commercial and image risks to an organisation.
  - As condições de licenças conferirão visibilidade a áreas de código que devem permanecer confidenciais?
  - Terá impacto na carteira de patentes da minha organização?
  - Como devem as equipas de projetos ser formadas e apoiadas neste assunto?
- Contribuir de volta aos projetos externos de código aberto é onde reside o maior valor do código aberto.
  - Como deve a minha empresa encorajar (e acompanhar) isto?
  - Como devem os programadores utilizar GitHub, GitLab, Slack, Discord, Telegram ou qualquer outra ferramenta que os projetos de código aberto habitualmente utilizam?
  - Pode o código aberto ter impacto nas políticas de RH da empresa?
- Claro que não se trata apenas de contribuir de volta, e então os meus próprios projetos de código aberto?
  - Estou pronto para fazer inovação *aberta*?
  - Como é que os meus projetos irão gerir as contribuições *recebidas*?
  - Devo gastar o esforço de alimentar uma comunidade para um determinado projeto?
  - Como devo liderar a comunidade, que papel devem ter os membros da comunidade?
  - Estou pronto a ceder as decisões do roteiro a uma comunidade?
  - Pode o código aberto ser uma ferramenta valiosa para reduzir a siloização entre as equipas da empresa?
  - Preciso de tratar da transferência de código aberto de uma entidade da empresa para outra?

### 8.3 Código aberto e soberania digital

Activity ID: [GGI-A-35](#).

#### Descrição

A soberania digital pode ser definida como a

”Capacidade e oportunidade de indivíduos e instituições executarem o(s) seu(s) papel(s) no mundo digital de forma independente, intencional e segura.” — Centro de Competência de TI Público, Alemanha

Para efectuar os negócios adequadamente, todas as entidades têm de contar com outros parceiros, serviços, produtos e ferramentas. A revisão dos laços e das restrições destas dependências permite à organização avaliar e controlar a sua dependência face a factores externos, melhorando assim a autonomia e a resiliência da mesma.

Como exemplo, a dependência de fornecedores é um forte fator de dependência que pode impedir os processos e o valor acrescentado da organização e, como tal, deve ser evitada. O código aberto é uma das formas de sair desta tranca. O código aberto desempenha um papel significativo na soberania digital, ao permitir uma maior escolha entre soluções, fornecedores e integradores e um maior controlo sobre os roteiros de TI.

É de notar que a soberania digital não é uma questão de confiança: é óbvio que precisamos de confiar nos nossos parceiros e fornecedores, mas a relação melhora ainda mais quando se baseia no consentimento mútuo e no reconhecimento, em vez de contratos e tensões forçados.

Eis algumas vantagens de uma melhor soberania digital:

- Melhorar a capacidade da organização de fazer as próprias escolhas sem constrangimentos.

- Melhorar a resiliência da empresa em relação a protagonistas e fatores externos.
- Melhorar a posição negocial ao lidar com parceiros e provedores de serviços.

### Avaliação de oportunidades

- Quanto difícil/barato é afastar-se de uma solução?
- Poderiam os fornecedores de soluções aumentar unilateralmente preços, simplesmente porque não temos escolha?
- Poderiam os fornecedores de soluções impor condições indesejáveis ao serviço deles (por exemplo, mudança de licença, atualizações de contratos)?

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta Atividade:

- Existe uma avaliação das dependências críticas para os fornecedores e parceiros da organização.
- Existe um plano de reserva para estas dependências identificadas.
- Existe um requisito declarado de soberania digital quando novas soluções são investigadas.

### Recomendações

- Identifique os riscos principais de dependência dos provedores de serviços e de entidades terceiras.
- Mantenha uma lista de alternativas de código aberto para serviços críticos.
- Adicione um requisito ao selecionar novas ferramentas e serviços utilizados na entidade, declarando a necessidade de soberania digital.

### Recursos

- [A Primer on Digital Sovereignty & Open Source: part I](#) e [A Primer on Digital Sovereignty & Open Source: part II](#), do sítio web Open-Sourcerers.
- Um excelente artigo no [superuser.openstack.org](#) sobre [The Role of Open Source in Digital Sovereignty](#). Aqui está um pequeno extrato:
 

A soberania digital é uma preocupação fundamental para o século XXI, especialmente para a Europa. O código aberto tem um papel importante a desempenhar na viabilização da soberania digital, permitindo a todos o acesso à tecnologia necessária, mas também a proporcionar a transparência da governação e a interoperabilidade necessárias para que essas soluções sejam bem sucedidas.
- A União Europeia assume a soberania digital, do [Open Source Observatory \(OSOR\)](#): Open Source, soberania digital e interoperabilidade: A Declaração de Berlim.
- A posição da UNICEF sobre [Open Source for Digital Sovereignty](#).

## 8.4 Código aberto permitindo a inovação

Activity ID: [GGI-A-36](#).

### Descrição

A inovação é a implementação prática de ideias que resultam na introdução de bens ou serviços novos ou na melhoria da oferta de bens ou serviços.

— Schumpeter, Joseph A.

O código aberto pode ser um fator chave para a inovação através de diversidade, colaboração e uma partilha fluente de ideias. Pessoas de origens e domínios diferentes podem ter perspetivas diferentes e fornecer respostas novas, melhoradas ou mesmo disruptivas a problemas conhecidos. É possível ativar a inovação ao ouvir opiniões diferentes e promover ativamente a colaboração aberta em projetos e tópicos.

Da mesma forma, participar na elaboração e implementação de normas abertas é um grande promotor de boas práticas e ideias para melhorar o trabalho diário da empresa. Também permite

à entidade conduzir e influenciar a inovação para onde e o que necessita e aumenta a visibilidade e reputação global dela.

Através da inovação, o código aberto possibilita não só transformar os bens ou serviços que a sua empresa comercializa, mas também criar ou modificar todo o ecossistema em que a sua empresa quer prosperar.

Como exemplo, ao lançar o Android como código aberto, a Google convida centenas de milhares de empresas a construir os próprios serviços com base nesta tecnologia de código aberto. A Google está assim a criar um ecossistema inteiro do qual todos os participantes poderiam beneficiar. É claro que muito poucas empresas são suficientemente poderosas para criar um ecossistema pela própria decisão. Mas há muitos exemplos de alianças entre empresas para criar um ecossistema desses.

### Avaliação de oportunidades

É importante avaliar a posição da sua empresa em comparação com os seus concorrentes e parceiros e clientes porque muitas vezes seria arriscado para uma empresa afastar-se demasiado das normas e tecnologias utilizadas pelos próprios clientes, parceiros e concorrentes. Inovação significa obviamente ser diferente, mas o que é diferente não deve representar um alcance demasiado grande; caso contrário, a sua empresa não beneficiaria dos desenvolvimentos de software feitos pelas outras empresas do ecossistema e do impulso comercial que o ecossistema proporciona.

### Avaliação do progresso

Os seguintes **pontos de verificação** demonstram o progresso nesta atividade:

- As tecnologias - e as comunidades de código aberto que as desenvolvem - que têm influência sobre o negócio foram identificadas.
- O progresso e as publicações destas comunidades de código aberto são monitorizados -- estou mesmo ciente da estratégia delas antes dos lançamentos serem publicados.
- Os funcionários da organização são membros de (algumas) destas comunidades de código aberto e influenciam os respectivos roteiros e escolhas técnicas, por contribuírem com linhas de códigos e participarem nos órgãos de governação destas comunidades.

### Recomendações

De todas as tecnologias que são necessárias para gerir o seu negócio, deve identificar:

- as tecnologias que podem ser as mesmas que os seus concorrentes,
- as tecnologias que devem ser específicas para a sua empresa.

Mantenha-se actualizado sobre as tecnologias emergentes. O código aberto impulsionou a inovação durante a última década e muitas ferramentas poderosas do dia-a-dia vêm de lá (pense em Docker, Kubernetes, projetos de Apache Big Data ou Linux). Não é preciso saber tudo sobre tudo, mas é preciso conhecer o estado da arte o suficiente para identificar novas tendências interessantes.

Permita, e encoraje, as pessoas a submeter ideias inovadoras, e a apresentá-las. Se possível, depende recursos nestas iniciativas e fazê-las crescer. Confie na paixão e vontade das pessoas de criar e fomentar ideias e tendências emergentes.

### Recursos

- [4 innovations we owe to open source.](#)
- [The Innovations of Open Source](#), do professor Dirk Riehle.
- [Open source technology, enabling innovation.](#)
- [Can Open Source Innovation Work in the Enterprise?.](#)
- [Europe: Open source software strategy.](#)
- [Europe: Open source software strategy 2020-2023.](#)

## 8.5 Código aberto permitindo a transformação digital

Activity ID: [GGI-A-37](#).

### Descrição

“Transformação digital é a adoção de tecnologia digital para transformar serviços ou empresas, através da substituição de processos não digitais ou manuais por processos digitais ou da substituição de tecnologia digital mais antiga por tecnologia digital mais recente.” (Wikipédia, de um artigo antigo)

Quando as organizações mais avançadas na transformação digital impulsionam em conjunto a mudança através dos seus negócios, TI e finanças para ancorar o digital no caminho, reconsideraram:

- Modelo de negócios: cadeia de valor com ecossistemas, “as a Service”, SaaS.
- Finanças: despesas operacionais/despesas de capital, pessoas, terceirização.
- TI: inovação, modernização de legados/ativos.

O código aberto está no centro da transformação digital:

- Tecnologias, práticas ágeis, gestão de produtos.
- Pessoas: colaboração, comunicação aberta, ciclo de desenvolvimento/decisão.
- Modelos de negócios: experimentar & comprar, inovação aberta.

Em termos de competitividade, os processos mais visíveis são provavelmente os processos que têm um efeito direto na experiência do cliente. Temos de reconhecer que os intervenientes maiores, bem como as empresas na fase de start-up, ao proporcionarem uma experiência de cliente totalmente sem precedentes, alteraram drasticamente as expectativas dos clientes.

A experiência do cliente, bem como todos os outros processos dentro de uma empresa, dependem inteiramente das TI. Cada empresa tem de transformar as suas TI, é do que se trata na transformação digital. As empresas que ainda não o fizeram, têm agora de conseguir a transformação digital delas o mais rápido possível, caso contrário o risco é que possam ser eliminadas do mercado. A transformação digital é uma condição de sobrevivência. Desde que as apostas são tão altas, uma empresa não pode deixar inteiramente a transformação digital a um fornecedor. Todas as empresas têm de adaptar a informática delas, o que significa que todas as empresas têm de trabalhar com software de código aberto porque não há informática sem software de código aberto.

As vantagens esperadas da transformação digital incluem:

- Simplificar, automatizar processos nucleares, fazê-los agir em tempo real.
- Permitir respostas rápidas às mudanças competitivas.
- Tirar partido da inteligência artificial e dos grandes dados.

### Avaliação de oportunidades

A transformação digital poderia ser dirigida por:

- Segmentos da TI: TI de produção, TI de suporte empresarial (GRC, faturação, aquisição, ...), TI de suporte (RH, finanças, contabilidade...), macrodados.
- Tipo de tecnologia ou processo de apoio à TI: infraestrutura (nuvem), Inteligência Artificial, processos (Make-or-Buy, DevSecOps, SaaS).

Injetar código aberto num determinado segmento ou tecnologia das suas TI revela que quer trabalhar neste segmento ou tecnologia, porque avaliou que este segmento ou tecnologia específica das suas TI é importante para a competitividade da sua empresa. É importante avaliar a posição da sua empresa em comparação não só com os seus concorrentes, mas também com outras indústrias e com intervenientes-chave em termos de experiência do cliente e de soluções de mercado.

### Avaliação do progresso

1. Nível 1: Avaliação da situação

- Identifiquei:
  - os segmentos das TI que são importantes para a competitividade da minha empresa, e
  - as tecnologias de código aberto necessárias para desenvolver aplicações nestes segmentos.
- E assim decidi:
  - em que segmentos quero gerir internamente o desenvolvimento de projetos, e
  - em que tecnologias de código aberto preciso de criar conhecimentos especializados internos.

#### 1. Envolvimento

- Em algumas tecnologias de código aberto selecionadas utilizadas na empresa, vários programadores foram treinados e reconhecidos como contribuintes valiosos pela comunidade de código aberto.

Projetos baseados em tecnologias de código aberto foram lançados em alguns segmentos selecionados.

#### 1. Nível 3: Generalização

- Para todos os projetos, uma alternativa de código aberto está a ser sistematicamente investigada durante a fase de início do projeto. Para facilitar o estudo dessa alternativa de código aberto por parte da equipa do projeto, um orçamento central e uma equipa central de arquitetos, alojada no departamento de TI, dedica-se a prestar assistência aos projetos.

#### **KPIs:**

- KPI 1. Taxa para a qual uma alternativa de código aberto foi investigada: (quantidade de projetos / quantidade total de projetos).
- KPI 2: \* Taxa para a qual a alternativa de código aberto foi escolhida: (quantidade de projetos / quantidade total de projetos).

#### **Recomendações**

Para além da palavra de ordem, a transformação digital é uma mentalidade que envolve algumas mudanças fundamentais e deve também (ou mesmo principalmente) provir das camadas superiores da organização. A administração deve promover iniciativas, novas ideias, gerir riscos e potencialmente atualizar os procedimentos existentes para os ajustar a novos conceitos.

A paixão é um enorme fator de sucesso. Um dos meios desenvolvidos pelos principais protagonistas no setor é a criação de aberturas para novas ideias, onde as pessoas possam submeter e trabalhar livremente, as ideias deles sobre a transformação digital. A administração deve encorajar tais iniciativas.

#### **Recursos**

- [Eclipse Foundation: Enabling Digital Transformation in Europe Through Global Open Source Collaboration.](#)
- [Europe: Open source software strategy.](#)
- [Europe: Open source software strategy 2020-2023.](#)

## 9 Conclusão

Como já dissemos antes, a boa governança de código aberto não é um destino; é uma jornada. Precisámo-nos preocupar com os nossos ativos comuns, com as comunidades e o ecossistema que o dificultam, porque o nosso próprio sucesso comum, portanto, individual, depende disso.

**Nós**, como profissionais de software e entusiastas de código aberto, estamos empenhados em continuar a melhorar o manual da Iniciativa de Boa Governança e trabalhar na divulgação e alcance dela. Acreditamos firmemente que organizações, indivíduos e comunidades precisam trabalhar lado a lado para construir um conjunto crescente de bens comuns, disponíveis e benéficos para todos.

**Você** é bem-vindo para se juntar à OSPO Alliance, contribuir para o nosso trabalho, divulgar a ideia e ser o embaixador de uma melhor consciencialização e governança de código aberto dentro do seu próprio ecossistema. Há uma variedade de recursos disponíveis, desde postagens de blog e artigos de pesquisa até a conferências e cursos de treino on-line. Também fornecemos um conjunto de material útil no [nosso site](#) e estaremos felizes em ajudar o máximo que pudermos.

**Vamos definir e construir juntos o futuro da Good Governance Initiative!**

### 9.1 Contato

A maneira preferida de entrar em contacto com a OSPO Alliance é postar uma mensagem na nossa lista de discussão pública em <https://accounts.eclipse.org/mailling-list/ospo.zone>. Também pode vir e discutir connosco nos eventos habituais de código aberto, participar dos nossos webinars mensais OSPO OnRamp ou entrar em contacto com qualquer membro - eles gentilmente redirecionarão-no à pessoa certa.

### 9.2 Apêndice: Modelo personalizado do Quadro de pontuação de atividades personalizados

A versão mais recente do modelo do Quadro de pontuação de atividades personalizado está disponível na secção recursos do [Good Governance Initiative GitLab](#) no OW2.

--	--	--

		07/28/21
• •		• •
• •	•	• •
1.		
2.		
3.		
4.		
•	• •	•
• •		



--	--	--

--