

Métricas de Open Source - Resumo

O que é Open Source?

- Antes de entender o que é uma métrica de open source, precisamos entender primeiro o que é Open Source*:
- *Open Source:
 - Segundo a Amazon Web Service (AWS), Open Source, ou em português, Código Aberto, é definido como: “um modelo de produção descentralizado que permite a qualquer pessoa modificar e compartilhar tecnologias porque seu projeto é acessível ao público. [...] que se baseia nos princípios de troca gratuita de informações, prototipagem rápida e desenvolvimento colaborativo.”
 - Sendo assim, é um código em que a comunidade pode interagir com ele, buscando sempre melhorar o projeto e trazer novas alterações. O Linux, é um dos exemplos mais conhecidos de software de código aberto. A intenção de seu autor, Linus Torvalds, era de que, com contribuição externa, o Sistema operacional pudesse ter evolução constante. Fato é que hoje temos variações do Linux, conhecidas como distros. Exemplos : Arch Linux, Ubuntu, Fedora e outros.

Princípios de código aberto

Ainda segundo a AWS, os princípios de OS baseiam-se em:

- Comunidade: Desenvolvimento baseado na contribuição de várias pessoas em prol de um objetivo comum;
- Transparência: As informações do código são disponibilizadas de forma livre para favorecer “tomadas de decisão eficazes e ampliam as ideias e descobertas uns dos outros.”

- Colaboração aberta: Criação de projetos que buscam incentivar o “trabalho em equipe” e não permitir que uma pessoa só faça todo trabalho, o que é inviável.
- Prototipagem rápida: O OS baseia-se em uma criação constante de protótipos por cada membro, buscando evolução;
- Meritocracia inclusiva: “As melhores ideias recebem mais apoio e esforço da comunidade de código aberto.” Isso não quer dizer que apenas as ideias de uma pessoa serão levadas em consideração, mas tudo é levado em consideração até que se chegue a um consenso.

Pontos principais:

- “Inspeção, modificação e aprimoramento”

Distribuição do código aberto: Para que possamos criar um código aberto, precisamos de:

- “Nenhuma restrição à venda ou distribuição do software como um componente da distribuição agregada de softwares
- Deve incluir e permitir a distribuição do código-fonte
- Deve permitir modificações e trabalhos derivados
- Os direitos associados ao programa devem ser aplicáveis a todos, sem discriminação.”

Métricas de Open Source

Por que utilizar métricas de Open Source ?

- Já que entendemos o que é Open Source (OS. Não confunda com sistema operacional), vamos agora entender o tópico principal deste resumo, que são as Métricas de Open Source. Primeiro, precisamos entender o porquê utilizá-los.
 - A razão de usar padrões no “O.S/C.A” é de medir o impacto que o seu código está gerando, tanto de visualizações quanto de utilização do seu aplicativo. Segundo o Open Source Guide: “pode ajudá-lo a tomar decisões melhores como um mantenedor open source”

- A métrica é aplicável/tem utilidade, segundo o Open Source Guide: “Se você está interessado em entender seu projeto em um nível mais profundo.”
- Agora que entendemos o porquê de usarmos a métrica de open source, vamos ver como podemos utilizá-la;

Como Utilizar?

1. O Open Source Guide diz que a métrica de C.A é necessária para, por exemplo:
 - “Entender como usuários respondem a uma nova funcionalidade;
 - Descobrir de onde os novos usuários vêm;
 - Identificar e decidir se se deve suportar um caso de uso ou uma funcionalidade sugerida;
 - Quantificar a popularidade do seu projeto;
 - Entender como seu projeto é usado;
 - Arrecadar dinheiro através de patrocínios e doações”
2. Algumas ferramentas são utilizadas para obter estes dados, algumas delas são:
 - Google Analytics
 - GitHub
 - No Github, temos a função “*traffic*”, dentro da função “*Insights*”. Nela, podemos ver a visualização da página, de onde os usuários vieram (qual site referenciou o seu, por exemplo), e mostra qual o conteúdo popular na sua página;
 - Existem também as **Github Stars**, que também ajudam a saber da popularidade da página;
3. Existem ainda ferramentas que nos auxiliam a saber da visualização comercial da sua página, como o **npm** ou o [rubygems.org](https://rvm.pubgems.org). Estas, fornecem informações de quantos downloads foram feitos sobre o seu site.
4. Blz, e onde a métrica entra nisso?
 - A métrica nos ajuda a analisar a aceitação do projeto, nos mostrando se o projeto está recebendo o público errado, ou não está conseguindo “segurar” seus visitantes.
 - Isso nos ajuda também a verificar a contribuição da comunidade. Caso você tenha muitos contribuidores inativos, ou com frequência baixa de contribuição, você pode ter um projeto insustentável. Portanto é preciso que estejamos abertos a receber novos contribuidores. Para isso, podemos ver, por meio de infográficos disponíveis no menu *contributors* do GitHub, quantos commits cada pessoa está fazendo. Se há muitos que estão fazendo commits frequentes, isso é um bom sinal.

5. Se o seu código está recebendo alta demanda, você precisa de ajuda. Portanto, ver o número de issues, e se ele aumenta, e ver a quantidade de pull requests, serve de índice para saber se o seu projeto está em destaque. Se há muitas issues e há muito tempo elas não são solucionadas, então seu projeto precisa ser revisado.
6. Precisamos também ficar atentos às novas ideias, portanto, responder, ainda que apenas de forma escrita, às ideias e contribuições dos usuários. Para saber se tudo está fluindo como esperado, saber o tempo médio para fechar as issues, e o tempo médio para merges de pull requests é essencial.
7. Resumo: Commits e participação da comunidade (parâmetros técnicos e qualitativos) são essenciais para saber a aceitação do projeto.

Fontes

1. “O que é open source?": <https://aws.amazon.com/pt/what-is/open-source>
2. “Métricas de open source": <https://opensource.guide/pt/metrics>