Relatório de Avaliação do Software Software Piloto

Daniela Soares Feitosa

Contexto

Realizamos uma avaliação do software `Software Piloto` no contexto da minha pesquisa de mestrado, orientada pela Profa. Dra. Christina von Flach G. Chavez.

O objetivo principal da pesquisa é investigar o estado-da-prática no que se refere ao *software de pesquisa*, considerando aspectos de sustentabilidade e grau de abertura (*FAIRness*), com o propósito de identificar conceitos e práticas utilizados e compreender barreiras enfrentadas no desenvolvimento de software de pesquisa sustentável e aberto sob a perspectiva de pesquisadores de grupos de pesquisa de diferentes áreas do conhecimento da Universidade Federal da Bahia.

Neste relatório, apresento os resultados da avaliação de sustentabilidade e abertura do software `Software Piloto`, destacando pontos positivos e recomendar boas práticas para torná-lo mais sustentável e aberto.

1 Critérios de Avaliação

Consideramos a avaliação do software de pesquisa sob duas perspectivas: (1) sustentabilidade ou direcionada a sua longevidade, e (2) abertura (*openness*) ou aderência aos princípios FAIR (*FAIRness*).

1.1 Sustentabilidade

A avaliação da sustentabilidade do *software de pesquisa* busca determinar se o mesmo é sustentável com base em critérios relevantes para a engenharia de software e para o ecossistema científico. A avaliação pode considerar atributos ou práticas usadas no desenvolvimento do software para oferecer uma visão geral ou detalhada da sustentabilidade do *software de pesquisa* –

por exemplo, disponibilidade do software, organização do código fonte, uso de padrões abertos, revisão por pares e testes, portabilidade, documentação, planos para o futuro, suporte, comunidade ativa, política de contribuições, identidade e licenças.

Em geral, a avaliação de sustentabilidade do *software de pesquisa* não gera um resultado objetivo *é sustentável/não é sustentável*, mas tende a valorizar pontos fortes do software e identificar pontos para sua melhoria.

A tabela 1 lista as práticas observadas no software de pesquisa.

Tabela 1: Avaliação de sustentabilidade do *software de pesquisa* baseada em práticas.

P	**Descrição**
P1	O software está hospedado em um repositório público
P2	O software utiliza controle de versão
P3	O software adota explicitamente uma licença
P4	O software está registrado e apresenta um DOI
P5	A estrutura de arquivos do projeto comunica a finalidade de seus elementos
P6	O software usa formato de dados e interfaces padronizadas
P7	A documentação apresenta uma visão geral do software
P8	O software possui testes
P9	O código é revisado antes de ser publicado
P10	O projeto de software utiliza rastreador de tarefas e *bugs*
P11	Tarefas repetitivas são automatizadas
P12	Há integração e implantação contínuas
P13	Há lançamento de versões do software
P14	Há evidência de uma comunidade (presente ou futuro)
P15	O software é divulgado para a comunidade acadêmica
P16	Há uma forma recomendada para citação do software

1.2 FAIRness

A avaliação de *FAIRness* do *software de pesquisa* pode ser definida como um processo de avaliação do seu grau de aderência aos princípios FAIR, originalmente definidos para dados de pesquisa, adaptados para software. Os princípios FAIR buscam oferecer uma visão para melhorar o compartilhamento e a reutilização de dados ou software por pessoas e máquinas.

A avaliação de *FAIRness* para software pode servir para apoiar a decisão de usuários e desenvolvedores sobre uma eventual adoção do *software de pesquisa* em seu projeto de pesquisa.

A tabela 2 lista as práticas observadas no software de pesquisa.

Tabela 2: Avaliação de FAIRness - Template.

Princ.	Descrição
F1	O software recebe um identificador globalmente único e persistente.
F1.1	Aos componentes do software que representam diferentes níveis de granularidade são atribuídos identificadores distintos.
F1.2	As diferentes versões do software recebem identificadores distintos.
F2	O software é descrito com metadados detalhados.
F3	Os metadados contém de forma clara e explícita o identificador do software que descrevem.
F4	Os metadados seguem os princípios FAIR, são pesquisáveis e indexáveis.
A1	O software é pode ser obtido por meio de seu identificador utilizando um protocolo de comunicação padronizado.
A1.1	O protocolo é aberto, gratuito e universalmente implementável.
A1.2	O protocolo permite procedimentos de autenticação e autorização, quando necessário.
A2	Os metadados são acessíveis, mesmo quando o software não está mais disponível.
I1	O software lê, escreve e troca dados de acordo com padrões da comunidade relacionados ao domínio.
I2	O software inclui referências qualificadas a outros objetos.
R1	O software é descrito com uma variedade de atributos precisos e relevantes.
R1.1	É atribuída uma licença clara e acessível ao software.
R1.2	O software possui informações detalhadas de procedência.
R2	O software inclui referências qualificadas a outros softwares.
R3	O software atende aos padrões relevantes da comunidade do domínio.

2 Avaliação de Sustentabilidade

2.1 Resumo

A tabela 3 apresenta um resumo da avaliação da sustentabilidade do software `Software Piloto`, indicando se o mesmo atende, não atende ou atende parcialmente às práticas, e comentários sobre como foram usadas.

Pontos Positivos

- O software está disponível em um repositório público com controle de versão;
- Organização bem descritiva da estrutura de pastas;
- Possibilidade de utilizar a ferramenta de rastreamento de *bugs*.

Sugestões de Melhorias

Licença Definir a licença de cada arquivo no repositório, incluindo as dependências e bibliotecas externas. Por exemplo, os arquivos nas pastas *external* e *plugins* estão sob a mesma licença que o repositório?

Registro de software Gerar um DOI para o software. Por exemplo, registrando uma versão do software no Zenodo [1] ou figshare [2];

Documentação Revisar os *links* listados na página do GitHub [3]. Alguns links redirecionam para uma página não existente;

Revisão de código Praticar a revisão de código, colocando como requisito para incorporação de código a aprovação de outro desenvolvedor. A revisão realizada por outros desenvolvedores podem perceber problemas na qualidade do código do *software de pesquisa* que testes automatizados podem não detectar;

Testes automatizados Não há no repositório arquivos de testes. A implementação de testes automatizados facilita a identificação de problemas no software e a manutenção, aumenta a confiança na estabilidade e fornecem o suporte para refatoração;

[1] `https://zenodo.org/`
[2] `https://figshare.com/`
[3] <software-site>

Tabela 3: Sustentabilidade do software `Software Piloto`.

P	Atende?	Comentário
P1	Sim	O software está disponibilizado em um repositório público no GitHub
P2	Sim	O controle de verão é implementado por utilizar o GitHub como plataforma de hospedagem
P3	Parcial	Um arquivo declarando a licença MIT. Porém não está claro se as permissões se aplicam a qualquer arquivo fonte no repositório
P4	Não	O repositório não menciona um DOI associado a ele mas pode ser encontrado no GitHub pelo nome
P5	Sim	A estrutura de pastas está organizada de forma descritiva e permite inferir o conteúdo
P6	Sim	Apesar de não haver documentação explícita, o software utiliza o formato de entrada e saída que facilita a integração com o MATLAB
P7	Parcial	O projeto utiliza o *GitHub pages* mas as informações estão incompletas e algumas URLs direcionam para um destino não válido.
P8	Não	Não há testes automatizados para o software
P9	Parcial	Todos os *pull requests* listados no repositórios foram aprovados pelo próprio autor, sugerindo que não houve revisão de código por outra pessoa
P10	Sim	O projeto aproveita a funcionalidade de rastreamento de *bugs* e tarefas disponível no GitHub
P11	Não	Não encontramos tarefas automatizadas no projeto
P12	Não	Não há integração contínua. Por ser um plugin instalado manualmente no MATLAB, a implantação contínua não é viável
P13	Não	O projeto não utiliza a funcionalidade de lançamento de versões no repositório do GitHub
P14	Não	Há apenas um desenvolvedor como autor
P15	Não	Não encontramos divulgação em eventos científicos
P16	Não	Não encontramos citação do software em publicações.

Integração contínua A implementação de integração contínua permite que as mudanças sejam automaticamente verificadas e integradas, facilitando a detecção e correção de problemas mais cedo no processo de desenvolvimento. Por exemplo, o código pode ser incorporado logo após a revisão do código e a execução com sucesso dos testes automatizados;

Lançamento de versões Fazer lançamento de versões;

Citação de software Criar um arquivo "CITATION.cff" no repositório definindo como o software deve ser citado em publicações[4]. Também é importante que o software seja citado em publicações e divulgado em eventos científicos.

2.2 Avaliação completa

Na avaliação de sustentabilidade, consideramos 16 práticas organizadas em cinco grupos.

Práticas Básicas

O software `Software Piloto` esteve hospedado em um repositório público desde o início de seu desenvolvimento (**P1**). Apesar da hospedagem do *software de pesquisa* no GitHub e uso das facilidades da plataforma, um backup periódico tem sido realizado em um dispositivo de armazenamento do grupo de pesquisa, mantido nas instalações do grupo. No GitHub a identidade do software é clara e única: o nome público do software é `Software Piloto`. Apesar de ter um arquivo README.md, não há uma descrição do projeto que facilite sua indexação nos mecanismos de busca.

Por estar hospedado no GitHub, `Software Piloto` possui suporte para controle de versão (**P2**). Inicialmente, o repositório do projeto não apresentava a licença de software (**P3**). Após quatro meses da realização da entrevista com o pesquisador sênior do grupo, houve um *commit* no repositório do software para a inclusão do arquivo *LICENSE*, descrevendo a licença escolhida, porém sem deixar claro se as permissões se aplicam a qualquer arquivo fonte do repositório. A licença atribuída ao software `Software Piloto` é a MIT[5], um licença usada em projetos de software livre e em projeto de software proprietário. Quanto ao registro do software (**P4**), o repositório não menciona se há um DOI associado a ele. Além disso, não encontramos uma referência para o software `Software Piloto` no Zenodo.

[4] `https://citation-file-format.github.io/`

[5] `https://opensource.org/license/mit/`

Organização do Projeto

O repositório do `Software Piloto` possui uma estrutura de arquivos (**P5**) bem definida, e usa nomes auto-explicativos para pastas e arquivos que facilitam a compreensão do propósito e utilidade do *software de pesquisa*. O software `Software Piloto` também se preocupa com padronização (**P6**) visto que é uma aplicação que depende do software MATLAB[6], uma plataforma paga para programação e computação numérica usada por engenheiros e cientistas para analisar dados, desenvolver algoritmos e criar modelos. Assim, o `Software Piloto` utiliza formatos de entrada e saída que facilitam a integração com o MATLAB.

No `Software Piloto`, a preocupação com documentação do software (**P7**) ainda é incipiente e voltada para usuários do software. Inicialmente, o repositório não tinha qualquer documentação. Quatro meses após a entrevista realizada com o pesquisador, um arquivo *README* foi adicionado ao repositório com descrição da ferramenta, lista de funcionalidades, informações sobre utilização e contribuição e licença utilizada pelo software (**P7**). Os autores do software não estão listados em um arquivo, mas é possível identificar as pessoas que contribuíram com o software a partir de informações sobre autores das mudanças registradas pelo sistema de controle de versão (*commits*). Finalmente, os desenvolvedores deram início à construção de um site para publicação de informações sobre o projeto de pesquisa e o software usando o *GitHub Pages*.

Qualidade

O *MATLAB Test*[7] é um conjunto de ferramentas para o desenvolvimento, gerenciamento, análise e teste de aplicações MATLAB. Entretanto, o *software de pesquisa* `Software Piloto` ainda não implementa testes de software automatizados (**P8**). Vale destacar que as ferramentas do *MATLAB Test*, assim como o MATLAB, são produtos de software fechados e que requerem assinaturas pagas.

Considerando que o software `Software Piloto` está publicado no GitHub, é possível realizar a revisão de código (**P9**) colaborativamente, utilizando a infraestrutura oferecida pela plataforma. Porém, até o dia da avaliação do *software de pesquisa*, todos os pedidos de mudança no código (*pull requests*) listados no repositório foram aprovados pelo próprio autor, sugerindo que não houve revisão de código por outra pessoa.

[6] `https://www.mathworks.com/products/matlab.html`

[7] `https://www.mathworks.com/products/matlab-test.html`

Gerência

O software `Software Piloto` está hospedado no GitHub e conta com o seu rastreador de tarefas e *bugs* nativo (**P10**). O *GitHub Docs*[8], apresenta uma breve introdução sobre como reportar um problema usando o rastreador de tarefas. Entretanto, o rastreador nativo do GitHub ainda não foi utilizado no projeto `Software Piloto`. Não foram encontradas tarefas automatizadas ou indício de automatização de tarefas (**P11**). Por fim, não há suporte para integração e implantação contínuas (**P12**). Por ser um *plugin* instalado manualmente pelo usuário no MATLAB, tais práticas não são viáveis.

O projeto não segue a prática de lançamento de versões (**P13**) no repositório do GitHub. As funcionalidades de 'Releases' e 'Tags' que facilitariam a referência e a descrição das funcionalidades e *bugs* incluídos naquela versão específicas não são utilizadas. Apesar de não realizarem lançamentos oficiais de versões, se fosse necessário citar o *software de pesquisa* em um artigo, os autores poderiam fazer referência a um *commit* específico no repositório do projeto. No histórico do projeto, vimos que um *commit* com a mensagem "Software Piloto 2.0" foi incorporado quatro meses após a entrevista. Essa mensagem sugere uma intenção de indicar uma alteração significativa no projeto e associar um número de versão.

Reconhecimento

O projeto ainda não possui uma comunidade (**P14**). Apenas o usuário dono do repositório submeteu *commits* e *pull requests* no projeto. A página do projeto no GitHub não apresenta outros usuários que adicionaram o projeto como favorito e não mostra usuários que fizeram uma cópia independente (*fork*) do projeto.

Não encontramos artigos ou outras formas de divulgação do *software de pesquisa* em eventos científicos (**P15**). O software `Software Piloto` / MATLAB é mencionado e foi usado em uma dissertação de mestrado na área de Saúde[9]. Entretanto, não encontramos no texto da dissertação um identificador ou referência para o software ou para a versão usada. Também não encontramos citação do software em publicações e na dissertação mencionada (**P16**), nem informações sobre como o software deveria ser citado.

[8] `https://docs.github.com/pt`

[9] <repositorio-dissertacao-url>

3 Avaliação de FAIRness

3.1 Resumo

A Tabela 4 apresenta um resumo da avaliação de *FAIRness* do `Software Piloto`. Cada linha da tabela apresenta um princípio, indicação se o software atende (S), atende parcialmente (P) ou não atende (N) ao princípio, e um comentário, se procedente.

Pontos Positivos

- Disponibilização em um repositório aberto de código com controle de versão;
- Adoção de uma licença de software;

Sugestões de Melhorias

Facilitar a busca no repositório Descrever o software na seção de edição dos detalhes do repositório. Incluir descrição e tópicos relacionados ao software. Essas informações facilitam a busca pelo software;

Preservar o software O software está armazenado na conta do desenvolvedor do software. Caso, por exemplo, o desenvolvedor apague a sua conta ou o projeto, o software não estaria mais disponível e as publicações que o citam ficariam com os *links* quebrados. Uma forma de preservar o software é disponibilizar em uma plataforma como o *Software Heritage*[10];

Identificação das dependências O software menciona que o software MATLAB é uma dependência e menciona a versão, mas poderia também inserir o *link* de onde obter o software. É mencionado no README a utilização de bibliotecas externas mas os nomes e versões não são mencionados;

3.2 Avaliação completa

F: O software e seus metadados associados são facilmente encontrados tanto por humanos quanto por máquinas

O software `Software Piloto` está hospedado com uma identidade clara e única no GitHub, não globalmente e o repositório não garante a persistência

[10] `https://www.softwareheritage.org`

Tabela 4: *FAIRness* no Software `Software Piloto`.

	S/N/P	**Comentário**
Localizável **Findable**		
F	*P*	
F1	*P*	O software possui uma identidade única apenas no GitHub e não é garantida a persistência.
F1.1	S	
F1.2	S	
F2	P	O software menciona alguns metadados, mas sem muitos detalhes que permitam encontrar o software facilmente
F3	P	Os metadados não apresentam de forma clara, mas é possível obter a partir do *commit*
F4	P	Há informações nos arquivos do repositório, mas não há descrição na configuração do projeto para facilitar indexação
Acessível		
A:	P	
A1	S	
A1.1	S	
A1.2	S	
A2	S	Caso o software não esteja mais disponível, os metadados também ficarão inacessíveis
Interoperável		
I:	P	
I1	P	Não está explícito como a troca de dados ocorre com o MATLAB
I2	P	As referências se resumem a menção sobre utilização, sem citar nomes, websites ou detalhes dos outros objetos
Reusável		
R:	P	
R1	P	Não há muitas informações sobre atributos e como reutilizar o software
R1.1	S	*MIT License.*
R1.2	P	Não há informações sobre como o software foi desenvolvido ou quais foram as intenções originais
R2	P	Não há informações relevantes sobre dependências
R3	S	

do identificador se o software for movido, por exemplo (**F1**). Os componentes do software, como classes e bibliotecas, apresentam identificadores distintos (**F1.1**). Cada versão do software pode ser unicamente identificada por um *hash de commit* e, a partir dele, é possível recuperar os metadados da versão específica (**F1.2**). O software apresenta metadados, mas não descreve as dependências, informações detalhadas sobre utilização e configuração e como deve ser a entrada e saída de arquivos (**F2**). Ao analisar os metadados a partir de um commit, é possível verificar qual versão específica do software o metadado está se referindo (**F3**). Apesar de incluir um arquivo README com informações do projeto, não há uma descrição na configuração do projeto que facilite a indexação nos mecanismos de busca (**F4**).

A: O software e seus metadados podem ser obtidos através de protocolos padronizados

O software `Software Piloto` pode ser obtido a partir do repositório do projeto no GitHub (**A1**). Não há restrições para baixar o código-fonte, como taxas os custos relacionados à licença (**A1.1**). É possível configurar o repositório para ser acessado apenas por pessoas autorizadas (**A1.2**). Os metadados estão descritos em arquivos no repositório, então ele não seriam acessíveis caso o repositório do software não esteja mais disponíveis (**A2**).

I: O software interopera com outros software trocando dados e/ou metadados através da interação via interface de programação de aplicativos (APIs), descrito por meio de padrões

O software `Software Piloto` é uma aplicação para MATLAB e, portanto, interopera seguindo seu padrão, mas a forma como a interação ocorre não está explicitamente descrito (**I1**). O repositório menciona o MATLAB como pré-requisito para executar a aplicação e inclui agradecimentos pela utilização de bibliotecas e recursos externos e recursos, mas não inclui referências qualificadas, como websites ou os nomes da bibliotecas e recursos externos utilizados (**I2**).

R: O software é tanto utilizável (pode ser executado) quanto reutilizável (pode ser compreendido, modificado, aprimorado ou incorporado a outros softwares)

O software `Software Piloto` não disponibiliza muitas informações descrevendo como reutilizar o software (**R1**). O software está licenciado sob a licença de código aberto MIT, que permite a reutilização, mas não deixa claro se todas as partes do software seguem a mesma licença (**R1.1**). A partir dos

commits e da lista de contribuidores do projeto no GitHub é possível saber quais pessoas contribuíram com o software, mas não há informações explícitas sobre como o software foi desenvolvido ou quais foram as intenções originais (**R1.2**). Os nomes e informações sobre as bibliotecas utilizadas não são mencionadas na documentação (**R2**). A comunidade MATLAB recomenda que sejam seguidas as melhores práticas de desenvolvimento de software em relação a correção, clareza e generalização e o software, em geral, atende a esses padrões (**R3**).

Considerações Finais

A avaliação do software `Software Piloto` proporcionou insights valiosos para a pesquisa em andamento. Agradeço ao Prof. Dr. Pesquisador Entevistado pela disponibilidade e cooperação na realização da pesquisa.