

# A portal for cataloging worked examples extracted from open source software

Simone de França Tonhão  
State University of Maringá  
siimone.franca@gmail.com

Thelma Elita Colanzi  
State University of Maringá  
thelma@din.uem.br

Igor Steinmacher  
Northern Arizona University  
igor.steinmacher@nau.edu

## ABSTRACT

Finding real examples that follow the evolution of Software Engineering is not an easy task for instructors in the field. Open Source Software (OSS) projects have been an alternative to support education in Software Engineering, as they allow contact with real development projects and environments. However, the adoption of these projects in the classroom presents some obstacles, such as choosing the appropriate project, and the time and effort needed to prepare the classes and get used to the open source environment. In order to reduce the difficulties faced by instructors, we are working on developing a portal to catalog worked examples extracted from OSS projects to assist instructors. In this paper, we present the first steps towards creating this catalog, which is creating a pattern for guiding the structuring the worked examples. We are applying the Design Science Research paradigm. The activities reported in this paper regards the first Design Cycle, and were divided into three stages related to the elaboration and evaluation of the pattern for cataloging the worked examples. In these stages, we carried out focus groups and interviews with Software Engineering instructors, including the instantiation of the pattern to catalog actual worked examples. After each stage, we had a new and improved version of the pattern for cataloging the worked examples. Such pattern served as a basis to create the first version of the worked examples portal, which will be evaluated in the next stages of the research.

## CCS CONCEPTS

• **Software and its engineering** → **Open source model**; • **Social and professional topics** → **Software engineering education**.

## KEYWORDS

Open Source Software, Software Engineering Education, Worked Examples

### ACM Reference Format:

Simone de França Tonhão, Thelma Elita Colanzi, and Igor Steinmacher. 2020. A portal for cataloging worked examples extracted from open source software. In *34th Brazilian Symposium on Software Engineering (SBES '20)*, October 21–23, 2020, Natal, Brazil. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3422392.3422471>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*SBES '20, October 21–23, 2020, Natal, Brazil*

© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-8753-8/20/09...\$15.00  
<https://doi.org/10.1145/3422392.3422471>

## 1 INTRODUÇÃO

Os cursos típicos de Engenharia de Software (ES) são tradicionalmente baseados em exposição de conteúdo teórico e na execução de pequenos projetos práticos [19]. Em geral, os requisitos para a implementação de tais projetos são definidos pelo professor e alunos envolvidos, e seguem um modelo de desenvolvimento em cascata, sendo elaborados em um curto período acadêmico, baseados em requisitos adaptados [18, 19]. Essa estratégia pode apresentar benefícios, contudo, o uso desses projetos, também conhecidos como “*toy projects*”, pode não oferece treinamento suficiente para que os alunos aprendam a lidar com problemas reais, aplicados em projetos semelhantes aos da indústria de software.

Encontrar exemplos reais e que acompanhem a evolução da ES não é uma tarefa fácil para os professores da área [23]. Muitas vezes, os professores se deparam com desafios, como a falta de recursos e materiais para evidenciar a aplicação de conceitos ou técnicas [22]. Dessa maneira, os professores podem acabar utilizando exemplos muito simples, e que não demonstram a aplicação dos conceitos em um ambiente real de desenvolvimento.

Os projetos de Software Livre (SL) são uma alternativa de apoio ao ensino de ES, e têm sido adotados por muitos educadores [7]. O uso de SL, além de contribuir para o conhecimento, pode ajudar os alunos a se alinharem com o desenvolvimento profissional de software. Isso acontece pois os projetos oferecem um contexto no qual os alunos podem praticar habilidades de *design*, desenvolvimento, e teste, tendo contato com projetos de complexidade, ambientes e tamanhos reais [22, 27].

Entretanto, a literatura oferece evidências de que existem obstáculos para adoção de projetos de SL no ensino de ES, entre eles a escolha dos projetos com os quais trabalhar [16, 22, 27]. Os projetos não podem ser muito complexos, pois podem gerar problemas de compreensão; e não devem ser muito simples, pois os alunos podem não ver as formas de aplicar os conceitos [12, 27]. Tais problemas culminam em um outro desafio: o esforço e tempo necessário para os professores prepararem as aulas baseadas em SL [3, 22]. Parte desse esforço está relacionado à necessidade de conhecer e familiarizar-se com os projetos de SL e encontrar os exemplos que se adequem à aula.

Dado esse contexto, este trabalho tem por objetivo apoiar os professores a superar as barreiras para identificar e utilizar exemplos trabalhados (*worked examples*) extraídos de projetos de SL no ambiente educacional. Para tal, está sendo construído um portal para acesso facilitado a exemplos trabalhados extraídos de projetos de SL. Assim, neste artigo é apresentado o padrão para catalogação dos exemplos trabalhados que serão inseridos no portal, bem como o processo de criação do mesmo.

O portal em desenvolvimento será disponibilizado abertamente, com o intuito de possibilitar que professores da ES possam consultar e utilizar os exemplos trabalhados catalogados, diminuindo o tempo de busca por materiais e recursos. Nesse trabalho, considera-se como exemplo trabalhado a declaração de um problema e a explicação passo a passo para se chegar a solução desse problema que facilita a construção de esquemas cognitivos [5, 28].

Para atingir o objetivo desta pesquisa, optou-se pelo paradigma de *Design Science Research*, e os três ciclos que o compõem: Ciclo de Relevância, Ciclo de *Design* e Ciclo de Rigor [11]. Neste artigo são apresentadas atividades preliminares, relacionadas ao primeiro Ciclo de *Design* da pesquisa. Tal ciclo, teve como foco a elaboração do padrão para catalogação dos exemplos trabalhados no portal.

O principal resultado apresentado nesse artigo é o padrão criado para catalogar os exemplos trabalhados extraídos de projetos de SL no portal. Tal padrão foi gerado por meio de atividades de elaboração, avaliação, e refinamento, e serviu como base para a criação de uma primeira versão do portal.

## 2 EXEMPLOS TRABALHADOS (WORKED EXAMPLES)

De acordo com Sweller, Van Merriënboer e Pass [28], um exemplo trabalhado é um artefato focado na declaração de um problema e nas etapas de solução desse problema. Já para Atkinson, Renkl e Merrill [1] um exemplo trabalhado consiste na formulação de um problema, nas etapas de solução, e na resposta final.

Geralmente, os estudantes que aprendem utilizando exemplos trabalhados tendem a cometer menos erros, resolver com mais facilidade e rapidez os problemas semelhantes, e a exigir menos assistência do professor [4]. Skudder e Luxton-Reilly [26] encontraram evidências que sugerem que os exemplos trabalhados são uma melhoria em relação às abordagens de solução de problemas, tanto no aspecto de tempo de aprendizagem, quanto no desempenho em problemas semelhantes. Isso ocorre porque os exemplos trabalhados diminuem a carga cognitiva imposta pela resolução de problemas, visto que, eliminam a busca por meios e fins, e facilitam na construção de esquemas cognitivos [28, 29]. O princípio dos exemplos trabalhados mostra que é melhor substituir alguns dos problemas práticos por exemplos que demonstrem a solução de determinado problema, e permita aos alunos primeiramente estudar essas soluções antes de resolver problemas sozinhos [2].

Essa abordagem pode trazer benefícios para o aprendizado, principalmente para alunos que não possuem um conhecimento prévio, podendo ser mais eficiente do que a aprendizagem através da resolução de problemas [20, 29]. A partir dos exemplos trabalhados é possível guiar os alunos para o desenvolvimento de uma compreensão sobre princípios e conceitos, focando nos estados problemáticos e nas etapas de solução, permitindo que os alunos construam soluções e esquemas generalizados, diminuindo o esforço cognitivo [24, 28].

Aprendendo através dos exemplos trabalhados, os estudantes dedicam mais recursos cognitivos aos conceitos que apoiam soluções corretas, e realizam aquisições de esquemas relevantes para o conhecimento. Quando os alunos tentam resolver problemas práticos sozinhos eles supõem procedimentos para realizar a resolução, esses procedimentos podem não ser apropriados, em alguns casos ineficientes e não generalizáveis, e até mesmo incorretos [28].

## 3 TRABALHOS RELACIONADOS

Reproduzir situações do mundo real na academia é muito difícil, a literatura descreve várias abordagens para aproximar os cursos de ES de ambientes realísticos de desenvolvimento, uma dessas abordagens é a utilização de projetos de SL no ensino [7–9, 16, 19, 21, 25]. Esses projetos podem ajudar a diminuir as lacunas entre a academia e o mercado de trabalho.

O estudo de Muller, Schindler e Slany [17] discute que a utilização dos projetos de SL no ensino pode ser benéfica para os alunos, tendo impacto positivo na futura carreira dos estudantes, visto que, os coloca de frente com problemas do ambiente real. Nos trabalhos de Morgan e Jensen [16], Papadopoulou, Stamelos e Meiszner [21], e Nascimento, Chavez e Bittencourt [19] são discutidas experiências nas quais os alunos tiveram contato direto com os projetos de SL. Nessas experiências, a abordagem baseada nos projetos de SL se mostrou uma possibilidade interessante no ensino. No entanto, foram encontrados alguns desafios, como a complexidade dos projetos trabalhados e a dificuldade do professor em acompanhar os alunos envolvidos nos projetos de SL.

Em Dorodchi e Dehbozorgi [7] os alunos também tiveram contato direto com os projetos de SL, porém, com foco no desenvolvimento colaborativo. Além disso, nesse trabalho, foram apresentados aos estudantes os projetos de SL humanitários. Os trabalhos de Ellis, Morelli e Hislop [8] e Ellis et al. [9] também abordaram os projetos de SL humanitários na educação em ES. A utilização desses projetos trouxe benefícios no ensino, mas também houve alguns desafios, como a inexperiência dos estudantes, duração limitada do curso, e complexidade dos projetos [8].

No trabalho de Silva et al. [25] os projetos de SL foram combinados com o uso de exemplos trabalhados no ensino de diagramas UML. Tal estudo focou apenas no reconhecimento inicial do projeto e ensino de UML, e pode não ter gerado os resultados esperados, mas o professor participante dos estudos se mostrou interessado em participar da experiência abordando outros tópicos da ES.

Os trabalhos supracitados evidenciam que a utilização dos projetos de SL traz benefícios para o ensino de ES, possibilitando que os alunos desenvolvam habilidades e experiências necessárias para a indústria [7, 19, 21]. Existem diversos trabalhos, com abordagens diferentes, que buscam melhorar o ensino de ES por meio do uso de projetos de SL. No entanto, apesar do aumento da utilização desses projetos na educação, a escolha de projetos apropriados, duração limitada do curso, o tempo e o esforço investidos para planejar, e a forma de conduzir as aulas, são desafios que podem dificultar a adoção de tal abordagem pelos professores [8, 16, 19, 25].

## 4 PROCEDIMENTOS METODOLÓGICOS

O objetivo deste artigo é apresentar as etapas iniciais do desenvolvimento de um portal para catalogação de exemplos trabalhados. Especificamente, apresenta-se o processo de criação e avaliação de um *padrão*—definido como o modelo elaborado para catalogação e organização dos exemplos de forma estruturada. Tal padrão é parte de uma proposta para apoiar professores na identificação de exemplos trabalhados extraídos de projetos de SL. A pesquisa está sendo desenvolvida utilizando o paradigma de *Design Science Research* (DSR), que objetiva o desenvolvimento de artefatos que

resolvam problemas reais em determinados domínios [30]. O artefato proposto na pesquisa é o portal para catalogação dos exemplos trabalhados extraídos dos projetos de software livre.

O DSR é dividido em três ciclos: Ciclo de Relevância, Ciclo de Design e Ciclo de Rigor. O foco deste artigo são as atividades do Ciclo de Design, que é o ciclo responsável pelas atividades de pesquisa relacionadas a construção do artefato, sua avaliação, e coleta de feedback para refinamento [10]. Os resultados preliminares apresentados neste artigo, trazem as atividades do Ciclo de Design referentes à elaboração e avaliação do padrão para catalogação dos exemplos trabalhados no portal.

Antes de definir a organização do portal, foi preciso projetar a forma de representação dos exemplos trabalhados e quais informações são necessárias para catalogação. Idealizou-se, então, um padrão que possa guiar os professores no processo de organização e catalogação das informações. O processo de elaboração e avaliação desse padrão foi dividido em três etapas, conforme apresentado na Figura 1.

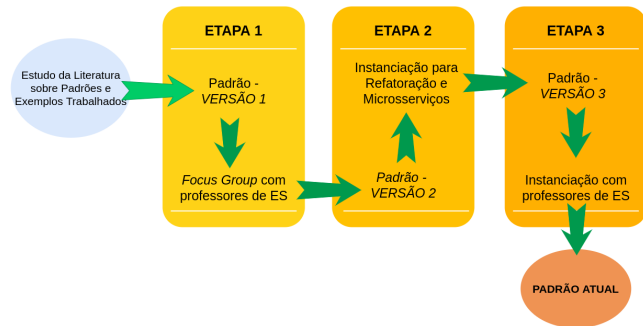


Figura 1: Etapas do processo de elaboração e avaliação do padrão para catalogação de exemplos trabalhados.

A primeira etapa consistiu na definição e elaboração da primeira versão do padrão, que teve como base estudos da literatura sobre padrões e exemplos trabalhados. Foi realizado também um grupo focal (*focus group*) com professores da área de ES para avaliar tal padrão. Na segunda etapa o padrão foi utilizado para criar exemplos trabalhados para os tópicos de refatoração e extração de microserviços, esta etapa visou verificar a flexibilidade do padrão. Na terceira etapa o padrão foi utilizado por professores de ES para criar exemplos trabalhados de refatoração, tal etapa teve como objetivo, obter *feedback* dos profissionais.

## 5 RESULTADOS E DISCUSSÕES

É importante fazer com que os exemplos trabalhados sejam fáceis de criar, entender, e utilizar, para que os professores não encontrem muitas dificuldades na adoção de tais exemplos, para isso criamos um padrão para estruturar e catalogar as informações no portal. Nesta seção, descrevemos as três etapas para criação e avaliação do padrão para catalogação, apresentamos também os resultados referentes a cada etapa.

### 5.1 Etapa 1 - Estudo da Literatura e Focus Group

Esta etapa foi conduzida de acordo com a Figura 2. Tais atividades envolveram estudos da literatura relacionados ao uso de exemplos trabalhados no ensino, e padrões na área da Educação, para elaboração da primeira versão do padrão. Realizou-se ainda um grupo focal com professores de ES para avaliar o padrão proposto.

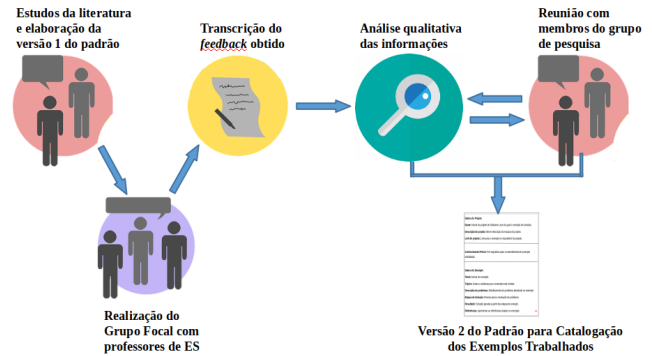


Figura 2: Atividades desenvolvidas na Etapa 1.

A primeira fase desta etapa, consistiu na definição da forma de representação dos exemplos trabalhados e na elaboração do padrão para catalogação. A primeira versão do padrão foi elaborada tendo como base estudos da literatura [2, 15, 25, 28]. O foco foi fornecer diretrizes para identificar, caracterizar, e estruturar as informações dos exemplos trabalhados. Tal padrão foi baseado no formato típico de padrão de reengenharia descrito em Demeyer, Ducasse e Niers-trasz [6]. Na Figura 3 é apresentada a primeira versão do padrão para catalogação dos exemplos trabalhados, dividido em três seções: dados do projeto, conhecimento prévio e dados do exemplo.

Com a primeira versão do padrão estruturada, iniciou-se a fase de avaliação. Tal avaliação consistiu na realização de um grupo focal, baseado nas etapas apresentadas em Kontio, Lehtola e Bragge [14], envolvendo professores da área de ES, a fim de avaliar a viabilidade do padrão. A partir do grupo focal obteve-se *feedback* sobre o padrão proposto para a estruturação e catalogação dos exemplos trabalhados (Figura 3).

O grupo focal contou com a participação de quatro professores (P0, P1, P2, e P3), conforme Tabela 1.

Tabela 1: Informações sobre os participantes dos estudos.

Professor	Universidade	Etapa 1	Etapa 3	Ensino de ES	Uso de SL
P0	PUC-Rio	X		15 anos	Pesquisa
P1	UFPA	X	X	-	Pesquisa
P2	UTFPR	X	X	14 anos	Pesquisa e contribuição
P3	UNESPAR	X	X	5 anos	Pesquisa e contribuição
P4	UTFPR		X	20 anos	Pesquisa
P5	UEMS		X	5 anos	Nunca
P6	UTFPR		X	13 anos	Pesquisa e contribuição

Os professores receberam o padrão, e um exemplo trabalhado criado seguindo o padrão, e foram convidados a discuti-los. O grupo focal foi gravado com o consentimento dos participantes, e, posteriormente, as informações foram transcritas, para realização de

<p><b>Dados do Projeto</b></p> <p><b>Nome:</b> Nome do projeto de Software Livre do qual o exemplo foi extraído.</p> <p><b>Descrição do projeto:</b> Breve descrição do escopo do projeto.</p> <p><b>Link do projeto:</b> Link para o exemplo no repositório do projeto.</p>
<p><b>Conhecimento Prévio:</b> Pré-requisitos para o entendimento do exemplo trabalhado.</p>
<p><b>Dados do Exemplo</b></p> <p><b>Título:</b> Nome do exemplo.</p> <p><b>Tópico:</b> Áreas e subáreas que o exemplo está contido.</p> <p><b>Descrição do problema:</b> Detalhamento do problema abordado no exemplo.</p> <p><b>Etapas de Solução:</b> Passos para a resolução do problema.</p> <p><b>Resultado:</b> Solução gerada a partir das etapas de solução.</p> <p><b>Referências:</b> Apresentar as referências citadas no exemplo. ▶</p>

Figura 3: Primeira versão do padrão de catalogação dos exemplos trabalhados.

análise qualitativa. Com essas informações, foram realizadas reuniões com os membros do grupo de pesquisa, para analisar quais melhorias poderiam ser realizadas no padrão.

Por meio da análise da *feedback*, foi possível aplicar melhorias no padrão. Um dos problemas levantados foi a **falta de elementos de cunho pedagógico** no padrão, nas palavras de P2 “*senti falta de características pedagógicas, falta de coisas para linkar o exemplo com a disciplina ou com o conteúdo específico da disciplina [...] Onde poderia ser usado e como isso poderia ser aplicado em sala de aula*”. Com o intuito de resolver tal problema foram criados dois elementos pedagógicos no padrão, sendo sugestões de uso e restrições de uso do exemplo trabalhado.

Os professores P1 e P2 sugeriram a criação de um elemento para **separar o contexto e o problema** no exemplo. Para isso, criamos um novo elemento chamado “contexto”, antes da “descrição do problema”, para descrever separadamente o contexto onde o exemplo está aplicado dentro do projeto de SL. Outro ponto discutido foi a **falta de mais elementos relacionados aos projetos de SL**. P2 apontou que poderia existir um link para o repositório do projeto e outro diretamente para o exemplo, P0 citou a atividade do projeto de SL, se é um projeto ativo ou não, e P1 ressaltou a importância de se ter um elemento para data de extração do exemplo, caso o projeto seja descontinuado ou seus links quebrados. Diante disso, criamos os elementos “link do exemplo”, “status do projeto”, e “data da extração do exemplo”. Foram definidos e marcados também quais elementos são obrigatórios na estrutura. A partir de tais mudanças obteve-se como resultado a segunda versão do padrão para catalogação.

## 5.2 Etapa 2 - Criação de exemplos trabalhados

Na Figura 4 são apresentadas as atividades de desenvolvimento da segunda etapa. Essas atividades envolveram a busca e seleção de exemplos para os tópicos de refatoração e extração de microsserviços, e a utilização do padrão para criação de exemplos para esses tópicos. Foi realizada também a análise, feita pelos membros do grupo de pesquisa, dos exemplos criados, com o intuito de verificar a flexibilidade do padrão.

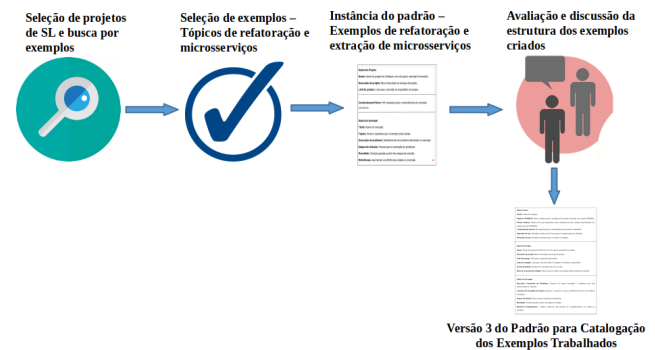


Figura 4: Atividades desenvolvidas na Etapa 2.

Com as melhorias da primeira etapa implementadas, seguiu-se para uma segunda iteração de avaliação do padrão. O primeiro passo foi a busca e seleção de exemplos para os tópicos de refatoração e extração de microsserviços. Optou-se por esses tópicos, tendo em vista que, um trata do sistema em nível de código, e o outro em nível de projeto de software, gerando assim, exemplos de formatos diferentes.

As buscas por exemplos foram realizadas por membros do grupo de pesquisa, em repositórios do GitHub. A busca para o tópico de refatoração foi realizada manualmente. Foram selecionados três projetos de SL, sendo eles: Arduino<sup>1</sup>, CodeIgniter4<sup>2</sup>, e OpenRCT2<sup>3</sup>. As buscas foram conduzidas analisando *pull requests* e *commits*. Três refatorações (exemplos) foram selecionadas, uma de cada projeto. Para o tópico de extração de microsserviços selecionamos dois repositórios da loja JPetStore<sup>4</sup>, um dos repositórios tratava do sistema na forma monolítica, e no outro o sistema estava organizado utilizando microsserviços. Para esse tópico foi selecionado apenas um exemplo.

Com os exemplos selecionados, seguiu-se para a segunda fase, na qual foram criados exemplos trabalhados seguindo o padrão para catalogação. Para o tópico de refatoração foram produzidos três exemplos trabalhados<sup>5</sup> seguindo o padrão. Tais exemplos abordaram as operações de refatoração *inline method*, *rename variable* e *remove redundancy*. A criação do exemplo trabalhado para o tópico de extração de microsserviços<sup>6</sup> teve como base o processo de extração de microsserviços de uma arquitetura monolítica, comparando o sistema nos dois repositórios selecionados da loja JPetStore.

<sup>1</sup><https://github.com/arduino/Arduino>

<sup>2</sup><https://github.com/codeigniter4/CodeIgniter4>

<sup>3</sup><https://github.com/OpenRCT2/OpenRCT2>

<sup>4</sup><https://github.com/mybatis/jpetstore-6>

<sup>5</sup>Disponível em <https://figshare.com/s/ad87727b39cae679ab7b>

<sup>6</sup>Disponível em <https://figshare.com/s/ad87727b39cae679ab7b>

Os exemplos trabalhados criados nesta etapa passaram por uma avaliação para verificar a efetividade do padrão para catalogação dos exemplos trabalhados e a adaptação do padrão para tópicos diferentes. A avaliação foi conduzida pelos membros da equipe de pesquisa, em busca de novas melhorias.

Visando maior fluidez na leitura dos exemplos, a ordem dos campos “contexto” e “descrição do problema” foi invertida no padrão, e os elementos foram renomeados para: “descrição conceitual do problema” e “contexto do problema no projeto”. Além disso, o padrão foi dividido em três seções: dados gerais, dados do projeto e dados do *worked example*, para que as informações fossem melhor apresentadas e divididas.

Os dados gerais tratam das informações relacionadas ao exemplo trabalhado, como título do exemplo trabalhado, e elementos pedagógicos, como conhecimento prévio e sugestões de uso, e tópicos do guia SWEBOK (*Software Engineering Body of Knowledge*) nos quais o exemplo está contido. Na seção de dados do projeto, são apresentados nome do projeto, links para o repositório, exemplo abordado, e algumas informações adicionais, como descrição, status do projeto, e data da extração do exemplo. A seção de dados do *worked example* apresenta a definição do exemplo trabalhado, tendo elementos como descrição do problema abordado, contexto do problema no projeto de SL, etapas de solução e resultado.

A partir da aplicação das melhorias levantadas nesta etapa, tivemos como resultado uma terceira versão do padrão.

### 5.3 Etapa 3 - Utilização do padrão por professores de ES

Na Figura 5 são esquematizadas as atividades que desenvolvemos na Etapa 3. Nesta etapa foi realizada uma nova busca e seleção de exemplos nos projetos de SL, para complementar a busca realizada na etapa anterior, no entanto, foram realizadas buscas somente para o tópico de refatoração. Após a seleção desses exemplos, convidamos professores de ES para transformá-los em exemplos trabalhados seguindo o padrão para catalogação. A partir do *feedback* apresentado pelos professores nesta etapa, foram realizadas novas melhorias no padrão.

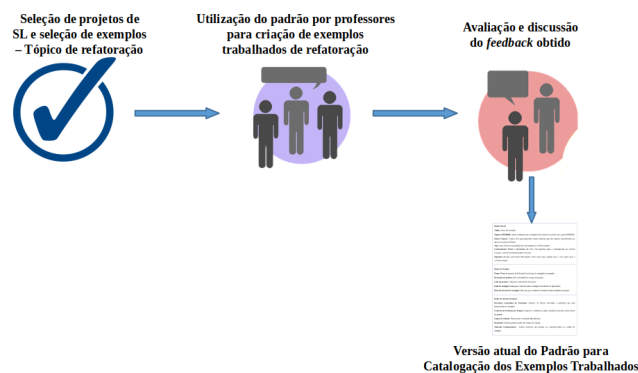


Figura 5: Atividades desenvolvidas na Etapa 3.

A primeira fase desta etapa consistiu no levantamento de exemplos de refatoração nos projetos de SL, conduzido pelos membros do grupo de pesquisa. A busca pelos exemplos foi conduzida tal

qual na Etapa 2, através da análise de *pull requests* e *commits* no GitHub. Os cenários selecionados nessa fase foram retirados dos seguintes projetos: Arduino, OpenMRS <sup>7</sup>, e Apache Dubbo <sup>8</sup>.

Com os exemplos selecionados, convidamos seis professores de ES: P1, P2, P3, P4, P5 e P6 (Tabela 1), para utilizarem o padrão e criarem exemplos trabalhados para os cenários selecionados. Cada professor recebeu um dos exemplos de refatoração, o padrão para catalogação, e um link para um protótipo do portal para catalogação dos exemplos trabalhados.

A coleta de informações dos estudos foi conduzida por meio do protocolo *think aloud* (ou pensar em voz alta). Nesse método, o participante é requisitado a falar em voz alta durante a execução de uma tarefa, ou resolução de um problema [13], e suas observações são anotadas para posterior análise. Sendo assim, durante o processo de criação do exemplo trabalhado o professor foi verbalizando seu pensamento e levantando possíveis problemas e melhorias no padrão. Cada criação de um novo exemplo trabalhado seguindo o padrão foi conduzida individualmente. Todas as considerações feitas pelos professores foram anotadas e discutidas pelos membros do grupo de pesquisa. A partir disso, foram realizadas novas melhorias no padrão.

Foi criado um novo elemento aberto chamado “outros tópicos”, para complementar os tópicos do guia SWEBOK. Além disso, o campo de “referências” do exemplo trabalhado foi renomeado para “material complementar”, e foi inserido um novo atributo chamado “tags”, no qual os professores poderão inserir tags de atributos técnicos e pedagógicos relacionados ao exemplo trabalhado.

Os campos “conhecimento prévio” e “restrições de uso” foram transformados em apenas um elemento, pois apresentam informações semelhantes. Além do mais, foram transformados em obrigatórios os campos “conhecimento prévio e restrições de uso”, “sugestões de uso” e “resultado” do exemplo trabalhado, pois foram considerados elementos essenciais para catalogação dos exemplos trabalhados, e o elemento “status do projeto” foi removido, dado que não foi considerado um elemento fundamental.

A partir dessas mudanças, obteve-se a versão atual do padrão para catalogação dos exemplos trabalhados extraídos de projetos de SL. Tal padrão é apresentado na Figura 6, e serviu como base para a criação de uma primeira versão do portal <sup>9</sup>.

## 6 CONCLUSÃO

Neste artigo foram apresentados os passos preliminares para a criação de um portal para catalogação de exemplos trabalhados extraídos de projetos de SL, para auxiliar os professores na utilização dos projetos de SL no ambiente educacional, e na adoção dos exemplos trabalhados no ensino de ES. Foi executado o primeiro Ciclo de *Design* da pesquisa, no qual foram realizadas iterações relacionadas a elaboração e avaliação do padrão para catalogação dos exemplos trabalhados.

Obteve-se uma versão atual desse padrão por meio da execução de avaliações baseadas em grupo focal, *feedback* de especialistas, e estudos realizados pelos membros do grupo de pesquisa. Tal padrão, representa um modelo com elementos de um exemplo trabalhado,

<sup>7</sup><https://github.com/openmrs/openmrs-core>

<sup>8</sup><https://github.com/apache/dubbo>

<sup>9</sup><https://portalworkedexamples.000webhostapp.com/>

<p><b>Dados Gerais</b></p> <p><b>Título:</b> Nome do exemplo.</p> <p><b>Tópicos SWEBOK:</b> Áreas e subáreas que o exemplo está contido de acordo com o guia SWEBOK.</p> <p><b>Outros Tópicos:</b> Tópico livre para apresentar outras subáreas que não estejam especificadas nos tópicos do guia SWEBOK.</p> <p><b>Tags:</b> Tags técnicas ou pedagógicas relacionadas ao <i>worked example</i>.</p> <p><b>Conhecimento Prévio e Restrições de Uso:</b> Pré-requisitos para o entendimento do <i>worked example</i>, e possíveis restrições para o seu uso.</p> <p><b>Sugestões de uso:</b> Apresentar informações sobre como usar, quando usar, e com quem usar o <i>worked example</i>.</p>
<p><b>Dados do Projeto</b></p> <p><b>Nome:</b> Nome do projeto de Software Livre do qual o exemplo foi extraído.</p> <p><b>Descrição do projeto:</b> Breve descrição do escopo do projeto.</p> <p><b>Link do projeto:</b> Link para o repositório do projeto.</p> <p><b>Link do exemplo:</b> Link para o local de onde o exemplo foi retirado no repositório.</p> <p><b>Data de extração do exemplo:</b> Data em que os dados do exemplo foram retirados do projeto.</p>
<p><b>Dados do Worked Example</b></p> <p><b>Descrição Conceitual do Problema:</b> Explicar de forma conceitual o problema que será demonstrado no exemplo.</p> <p><b>Contexto do Problema no Projeto:</b> Explicar o contexto no qual o problema descrito ocorre dentro do projeto.</p> <p><b>Etapas de solução:</b> Passos para a resolução do problema.</p> <p><b>Resultado:</b> Solução gerada a partir das etapas de solução.</p> <p><b>Material Complementar:</b> Indicar materiais que possam ser complementares ao estudo do exemplo.</p>

**Figura 6: Padrão para catalogação dos exemplos trabalhados.**

e informações consideradas importantes para o seu entendimento, além de informações de cunho pedagógico, importantes para auxiliar os professores na adoção desses exemplos. Esse padrão serviu também como base para a criação de uma primeira versão do portal, que será avaliado nas próximas etapas da pesquisa em estudos realizados com professores de ES.

Nos estudos realizados o padrão foi instanciado apenas para os tópicos de refatoração e extração de microsserviços, gerando exemplos trabalhados que envolvem dois estágios: “antes” e “depois” de determinada situação. Para tópicos de ES que não possuam esses dois estágios, o padrão proposto pode necessitar de eventuais adaptações, o que será avaliado em trabalhos futuros.

É importante ressaltar que o foco do trabalho é investigar como o portal pode ser útil aos professores que desejam adotar exemplos reais baseados em projetos de Software Livre. O objetivo final é que o portal seja alimentado colaborativamente, no entanto, está fora do escopo do presente trabalho abordar como o portal será populado, ficando para trabalhos futuros o objetivo de estudar atratividade e tornar o ambiente colaborativo.

## REFERÊNCIAS

- [1] Robert K Atkinson, Alexander Renkl, and Mary M Merrill. 2003. Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *J. Educ. Psychol.* 95, 4 (2003), 774.
- [2] Julie L Booth, Kelly M McGinn, Laura K Young, and Christina Barbieri. 2015. Simple practice doesn't always make perfect: Evidence from the worked example effect. *Policy Insights from the Behavioral and Brain Sciences* 2, 1 (2015), 24–32.
- [3] Joseph Buchta, Maksym Petrenko, Denys Poshyvanyk, and Václav Rajlich. 2006. Teaching evolution of open-source projects in software engineering courses. In *2006 22nd IEEE International Conference on Software Maintenance*. IEEE, 136–144.
- [4] William M Carroll. 1994. Using worked examples as an instructional support in the algebra classroom. *Journal of educational psychology* 86, 3 (1994), 360.
- [5] Xingliang Chen, Antonija T Mitrovic, and Moffat Matthews. 2019. Learning from worked examples, erroneous examples and problem solving: towards adaptive selection of learning activities. *IEEE Trans. Learn. Tecnologia* (2019).
- [6] Serge Demeyer, Stéphane Ducasse, and Oscar Nierstrasz. 2008. *Object-oriented reengineering patterns*. <https://boris.unibe.ch/104439/1/OORP.pdf>
- [7] Mohsen Dorodchi and Nasrin Dehbozorgi. 2016. Utilizing open source software in teaching practice-based software engineering courses. In *FIE 2016*. IEEE, 1–5.
- [8] Heidi JC Ellis, Ralph A Morelli, and Gregory W Hislop. 2008. Work in progress-challenges to educating students within the community of open source software for humanity. In *2008 38th Annual Frontiers in Education Conference*. IEEE, S3H–7.
- [9] Heidi Ellis JC, Becka Morgan, Gregory W Hislop, Ben Coleman, and S Monisha Pulimood. 2015. Software engineering learning in HFOSS: A multi-institutional study. *age* 26 (2015), 1.
- [10] Alan Hevner and Samir Chatterjee. 2010. Design science research in information systems. In *Design research in information systems*. Springer, 9–22.
- [11] Alan R Hevner. 2007. A three cycle view of design science research. *Scandinavian journal of information systems* 19, 2 (2007), 4.
- [12] Letizia Jaccheri and Thomas Osterlie. 2007. Open source software: A source of possibilities for software engineering education and empirical software engineering. In *FLOSS'07: ICSE Workshops 2007*. IEEE, 5–5.
- [13] Monique WM Jaspers, Thiemo Steen, Cor Van Den Bos, and Maud Geenen. 2004. The think aloud method: a guide to user interface design. *Int. J. Med. Inform* 73, 11-12 (2004), 781–795.
- [14] Jyrki Kontio, Laura Lehtola, and Johanna Bragge. 2004. Using the focus group method in software engineering: obtaining practitioner and user experiences. In *Symposium on Empirical Software Engineering ISESE'04*. IEEE, 271–280.
- [15] Kelly M McGinn, Karin E Lange, and Julie L Booth. 2015. A worked example for creating worked examples. *Mathematics Teaching in the Middle School* 21, 1 (2015), 26–33.
- [16] Becka Morgan and Carlos Jensen. 2014. Lessons learned from teaching open source software development. In *International Conference on Open Source Systems*. Springer, 133–142.
- [17] Matthias Müller, Christian Schindler, and Wolfgang Slany. 2019. Engaging students in open source: Establishing foss development at a university. In *HICSS*.
- [18] Jagadeesh Nandigam, Venkat N Gudivada, and Abdelwahab Hamou-Lhadj. 2008. Learning software engineering principles using open source software. In *FIE 2008*. IEEE, S3H–18.
- [19] Debora Nascimento, Christina Chavez, and Roberto Bittencourt. 2018. The adoption of open source projects in engineering education: a real software development experience. In *FIE 2018*. IEEE, 1–9.
- [20] Fleurie Nievelstein, Tamara Van Gog, Gijs Van Dijck, and Henny Boshuizen. 2013. The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemp. Educ. Psychol.* 38, 2 (2013), 118–125.
- [21] Pantelis M Papadopoulos, Ioannis G Stamelos, and Andreas Meiszner. 2013. Enhancing software engineering education through open source projects: Four years of students' perspectives. *Educ Inf Technol (Dordr)* 18, 2 (2013), 381–397.
- [22] Gustavo Pinto, Fernando Figueira Filho, Igor Steinmacher, and Marco Gerosa. 2017. Training software engineers using open-source software: the professors' perspective. In *CSEE&T 2017*. IEEE, 117–121.
- [23] Václav Rajlich. 2013. Teaching developer skills in the first software engineering course. In *ICSE 2013*. IEEE, 1109–1116.
- [24] Rolf Schwonke, Alexander Renkl, Carmen Krieg, Jörg Wittwer, Vincent Alevan, and Ron Salden. 2009. The worked-example effect: Not an artefact of lousy control conditions. *Comput. Hum. Behav* 25, 2 (2009), 258–266.
- [25] Fernanda G. Silva, Moara S. Brito, Jenifer V.T. Tavares, and Christina V.F. Chavez. 2019. FLOSS in Software Engineering Education: Supporting the Instructor in the Quest for Providing Real Experience for Students. In *SBES 2019*. ACM, 234–243.
- [26] Ben Skudder and Andrew Luxton-Reilly. 2014. Worked examples in computer science. In *ACE 2014*. Australian Computer Society, Inc., 59–64.
- [27] Therese Mary Smith, Robert McCartney, Swapna S Gokhale, and Lisa C Kaczmarczyk. 2014. Selecting open source software projects to teach software engineering. In *Technical symposium on Computer science education*. ACM, 397–402.
- [28] John Sweller, Jeroen Van Merriënboer, and Fred Paas. 1998. Cognitive architecture and instructional design. *Educ. Psychol. Rev.* 10, 3 (1998), 251–296.
- [29] Tamara Van Gog and Liesbeth Kester. 2012. A test of the testing effect: acquiring problem-solving skills from worked examples. *Cogn. Science* 36, 8 (2012), 1532–1541.
- [30] Roel Wieringa. 2009. Design science as nested problem solving. In *4<sup>th</sup> Intl. Conf. on Design Science Research in Information Systems and Technology*. 1–12.