

Aprendendo com os Erros dos Outros: Um relato sobre a adoção de Exemplos Errôneos como ferramenta de Aprendizagem de Casos de Uso

Williamson Silva^{1,2}, André L. M. Mendes¹, Renato Garcia^{1,2},
Matheus de Oliveira Ribeiro^{1,2}, Ricardo Vilela³, Walter Nakamura⁴,
Gabriel M. Lunardi⁵, Pedro Valle⁶

¹Universidade Federal do Pampa - UNIPAMPA (Campus Alegrete), Alegrete, RS, Brasil

²PPGES (UNIPAMPA - Campus Alegrete), Alegrete, RS, Brasil

³Universidade Federal do Cariri – UFCA, Juazeiro do Norte, CE, Brasil

⁴Universidade Tecnológica Federal do Paraná – UTFPR, Campo Mourão, PR, Brasil

⁵Universidade Federal de Santa Maria – UFSM, Cachoeira do Sul, RS, Brasil

⁶Universidade Federal de Juiz de Fora – UFJF, Juiz de Fora, MG, Brasil

^{1,2}{williamsonsilva, andremiranda.aluno}@unipampa.edu.br

^{1,2}{renatogarcia.aluno, matheusdor.aluno}@unipampa.edu.br

³ricardo.ferreira@ufca.edu.br, ⁴waltertakashi@professores.utfpr.edu.br

⁵gmlunardi@gmail.com, ⁶pedrohenrique.valle@ufjf.br

Abstract. *Use Cases (use case diagram and textual description of the use case) are widely recognized and used to capture systems requirements. However, software engineers and students need help understanding the syntax and semantics of these models. These gaps can affect the final quality of the software since these models would represent the software incompletely and incorrectly. Therefore, guiding the students while they are learning the diagrams is necessary. Instructors can use alternative pedagogical strategies to teach software modeling effectively in Computing courses, such as Learning from Erroneous Examples (ErrEx). This paper presents an experience report that shows the students' perception of using ErrEx to support software modeling learning. The results show that ErrEx helps to understand some concepts, identify defects, and avoid repeating them in the future.*

Resumo. *Os Casos de Uso (diagrama de caso de uso e descrição textual do caso de uso) são amplamente reconhecidos e usados como um meio de capturar requisitos de sistemas. No entanto, engenheiros de software e estudantes têm lacunas de aprendizado na compreensão da sintaxe e semântica desses modelos. Essas lacunas podem afetar a qualidade final do software, uma vez que esses modelos representariam o software de forma incompleta e incorreta. Logo, faz-se necessário orientar os estudantes enquanto estão aprendendo os diagramas. Para ensinar a modelagem de software de maneira eficaz nos cursos de*

Computação, os professores podem utilizar estratégias pedagógicas alternativas, como Learning from Erroneous Examples (ErrEx). Este artigo apresenta um relato de experiência que mostra a percepção dos estudantes sobre o uso do ErrEx como apoio na aprendizagem de modelagem de software. Os resultados mostram que o ErrEx ajuda a entender alguns conceitos, na identificação de defeitos e a evitar repeti-los no futuro.

1. Introdução

Os Casos de Uso (*Use Cases*- UC) tornaram-se uma técnica de especificação de requisitos amplamente aceita na indústria de software [Misbhauddin and Alshayeb 2015]. Segundo Tiwari e Gupta (2013), é uma estratégia de documentação de requisitos que auxilia na especificação das funcionalidades que devem ser implementadas pelo engenheiro de software [Misbhauddin and Alshayeb 2015, Tiwari and Gupta 2015], além de auxiliar na comunicação entre os membros da equipe do projeto e outras partes interessadas. Geralmente, os casos de uso são compostos por duas partes: o diagrama e a descrição textual das UCs [Tiwari and Gupta 2015, Anda et al. 2009]. O diagrama mostra a estrutura das UCs, os atores e as relações entre eles. A descrição textual complementa o diagrama, pois descreve o comportamento do UC de forma estruturada, conforme recomendado por muitas metodologias de desenvolvimento de software orientadas a modelos [Tiwari and Gupta 2015]. Na descrição textual, pode-se representar um conjunto de cenários, que são descritos em linguagem natural (narrativa) e obedecem a uma estrutura sintática ou utilizam uma estrutura pré-definida, conhecida como *template* [Tiwari and Gupta 2015, Alistair 2001].

Alguns estudos relatam que, embora UC seja um dos modelos mais simples da *Unified Modeling Language* (UML), engenheiros de software e estudantes ainda têm lacunas de aprendizado para entendê-los, tais como [Misbhauddin and Alshayeb 2015, Nascimento et al. 2017]: dificuldade em identificar as relações entre as UCs em um diagrama, dificuldade em definir o nível de granularidade do diagrama, dificuldade em identificar ou descrever os fluxos na descrição textual. Se as lacunas de aprendizado não forem abordadas ou sanadas, a má qualidade dos UCs ameaça todo o processo de desenvolvimento de software, pois esses modelos podem representar o software de forma incompleta e incorreta [Tiwari and Gupta 2015, Anda and Sjøberg 2002, Nascimento et al. 2016]. Assim, estratégias rentáveis para melhorar a aprendizagem de estudantes e especialistas, e para diminuir o número de erros nas UCs são cruciais [Oliveira et al. 2015].

A literatura vem relatando que uma possível causa para estas lacunas de aprendizagem dos estudantes pode estar relacionada com a forma como a modelagem de software, com diagramas UML, vem sendo ensinada [Ma 2017]. Os métodos tradicionais de ensino não tem se mostrado eficazes, pois focam em um ambiente de aprendizagem centrado no docente para apresentar os conceitos de modelagem aos estudantes. Silva *et al.* (2019) argumentam que os docentes necessitam utilizar diferentes estratégias pedagógicas em sala de aula afim de obter resultados de aprendizagem mais eficazes. Para isso, os estudantes devem ser orientados especialmente durante a fase inicial de aprendizado.

Uma estratégia que os docentes usam e que tem ganhado destaque entre os docentes é o uso de exemplos em sala de aula. Usar exemplos em sala de aula ajuda os estudantes a entenderem o passo a passo sobre como resolver determinado pro-

blema e a estruturar o aprendizado, desenvolvendo habilidades de resolução de problemas [Van Gog and Rummel 2018]. Os docentes fazem uso de exemplos, principalmente exemplos errôneos / incorretos, quando querem testar o nível de atenção e aprendizado dos estudantes. Ao utilizar esta estratégia, os docentes estão fazendo uso da Aprendizagem Baseada em Exemplos Errôneos (*Learning from Erroneous Examples - ErrEx*) [McLaren et al. 2016]. A partir desta estratégia, os estudantes tentam resolver um problema observando um ou mais exemplos que estão incorretos. Adotar a estratégia ErrEx em sala de aula ajuda os estudantes principalmente na aquisição inicial de habilidades no processo de aprendizagem [McLaren et al. 2016]. Além disso, ErrEx ajuda os estudantes a estudar mais cuidadosamente os exemplos e a lembrar e evitar erros encontrados e aprendidos em atividades futuras [Große and Renkl 2007]. No entanto, estudos experimentais ou relatos de experiência evidenciando a percepção dos estudantes sobre o uso do ErrEx durante a aprendizagem são bastante escassos, especialmente no domínio da ES e em modelagem de software.

Dado o contexto, este artigo apresenta um estudo experimental realizado com 47 estudantes de graduação sobre o uso do ErrEx como forma de apoio na aprendizagem de modelagem de software, mais especificamente diagrama de Casos de Uso. Para isso, foram analisadas as percepções dos estudantes a partir de sessões de Grupo Focal [Kontio et al. 2004]. A análise dos dados foi feita qualitativamente empregando alguns procedimentos de codificação. Isso permitiu entender melhor as dificuldades e os benefícios percebidos pelos estudantes após o uso da ErrEx como forma de apoio a aprendizagem de casos de uso.

2. Fundamentação Teórica e Trabalhos Relacionados

ErrEx (*Learning from Erroneous Examples*) faz uso de exemplos trabalhados errôneos que incluem um ou mais erros nas etapas de solução e os estudantes são responsáveis por detectar, explicar e/ou corrigir o erro [Tsovaltzi et al. 2010]. Os exemplos errôneos podem ser apresentados de várias maneiras, com diferentes tipos de *feedback*, diversas estratégias tutoriais e sequenciamento de material de aprendizagem [Tsovaltzi et al. 2010]. Nesse sentido, docentes podem determinar quais partes do material de aprendizagem são propensas a erros e podem, então, compilar os exemplos errôneos, de acordo com as necessidades de cada estudante, contribuindo para uma melhor aprendizagem.

Nota-se que o ErrEx é uma estratégia pedagógica empregada para evitar a “passividade” dos estudantes durante as aulas, uma vez que permite que os estudantes estudem os exemplos com mais cuidado, além de lembrar e evitar cometer os erros percebidos no futuro [McLaren et al. 2016]. As descobertas sobre a eficácia do ErrEx mostraram que aprender com os erros pode fomentar a aquisição de “conhecimento negativo” que fornece proteção importante contra decisões errôneas [McLaren et al. 2016, Adams et al. 2014]. Dessa forma, os estudantes podem entender claramente o que está errado e por que algo está errado em uma determinada situação [Curry 2004, Silva et al. 2017].

Apesar das vantagens mencionadas, poucas pesquisas sobre o uso do ErrEx vem sendo conduzidas pela comunidade de Educação em Computação [Skudder and Luxton-Reilly 2014, Silva et al. 2017]. Bolloju *et al.* (2011) realizaram um experimento controlado comparando a utilidade do uso de exemplos positivos versus negativos no ensino de modelagem de software, de modo a permitir

melhores resultados de aprendizagem para os estudantes. Como resultados, os autores perceberam que orientar os estudantes sobre o que não fazer parece mais eficaz durante a modelagem. É porque a direção positiva tem muitas formas possíveis de implementação, enquanto a direção negativa é mais focada e específica. Nesse sentido, a direção negativa é mais clara e memorável.

Balaban *et al.* (2015) apresentaram aos estudantes um catálogo de antipadrões de diagramas de classes. Além disso, os autores realizaram dois experimentos a fim de examinar até que ponto os antipadrões de modelagem ajudam os estudantes a identificar erros nos diagramas de classes. Ambos os experimentos mostraram que os antipadrões tiveram um impacto positivo significativo: os estudantes foram capazes de identificar problemas de modelagem após serem expostos a antipadrões. Além disso, os resultados mostraram que a taxa de identificação melhorou significativamente quando os antipadrões foram apresentados os estudantes.

Silva *et al.* (2017) realizaram um estudo experimental com o objetivo de compreender a influência do método ErrEx enquanto os estudantes estavam aprendendo diagramas UML (diagrama de atividade e diagrama de sequência). Os resultados mostraram que os estudantes modelaram diagramas de atividades mais completos usando o método ErrEx. Além disso, notou-se que os estudantes conseguiram compreender os conceitos devido ao diagrama errôneo que possuíam, e que, embora errado, ajuda no início da modelagem, e a identificação desses defeitos evitou a repetição de tais erros, além de ajudar na compreensão de alguns conceitos.

Por fim, Shmallo e Shrot (2020) discutiram sobre a aplicação de erros no design de um diagrama de classe UML, com o objetivo de categorizar os erros dos estudantes e avaliar se a abordagem pode levar a diagramas mais precisos. A pesquisa foi realizada com estudantes de engenharia, e 20 erros foram observados na concepção do diagrama antes da atividade. Após a atividade, os estudantes cometeram menos erros e avaliaram positivamente a abordagem. A pesquisa sugere que essa abordagem pode ser aplicada em diferentes áreas e tópicos de ensino, aumentando a motivação dos estudantes.

Apesar de haver alguns estudos relatando a eficácia do ErrEx na melhoria da aprendizagem dos estudantes em diversas áreas de ensino, incluindo a Educação em Computação, ainda há uma carência de pesquisas sobre o uso do ErrEx em sala de aula. Ainda mais, há poucos estudos sobre a melhor forma de apresentar exemplos errôneos para maximizar a compreensão dos estudantes e, assim, melhorar a sua aprendizagem. Portanto, este trabalho tem como objetivo investigar mais sobre a eficácia do uso de exemplos errôneos a partir da percepção dos estudantes. Assim, será possível compreender também a maneira mais eficiente de apresentar esses exemplos aos estudantes para que eles possam aproveitá-los ao máximo.

3. Metodologia da Pesquisa

Para entender as percepções de estudantes sobre o uso do ErrEx, foi conduzido um estudo experimental, em que foi utilizado o ErrEx para apoiar durante a modelagem do diagrama. A compreensão da percepção dos estudantes ajuda a identificar quais fatores os impedem e/ou os ajudam a alcançar as metas de aprendizado. Além disso, isso pode ajudar os professores a entender como esse método pode ser empregado no contexto de ensino de modelagem com diagramas UML.

3.1. Execução do estudo

O estudo foi conduzido com 47 estudantes do curso de graduação. Inicialmente, os estudantes foram divididos em equipes definidas por conveniência, em que eles decidiram quais estudantes fariam parte do seu grupo. Ao todo, foram formados dez grupos: sete grupos de cinco estudantes e três grupos de quatro estudantes. O experimento ocorreu em duas sessões. Cada sessão teve duração de duas aulas (duração de 40min por aula).

Na primeira sessão, os estudantes assistiram a uma aula teórica sobre diagrama de casos de uso, que enfatizou os conceitos necessários para que os estudantes entendessem os objetivos e a notação do diagrama de casos de uso. Além disso, foram realizados alguns exercícios com o objetivo de analisar se os estudantes tinham entendido o que fora apresentado na aula teórica. Na segunda sessão, o seguinte procedimento foi realizado: (a) treinamento com os estudantes sobre como utilizar o ErrEx; (b) entrega de um cenário com um problema descrito; (c) entrega de exemplos de diagramas de casos de uso, com defeitos sintáticos e semânticos para os estudantes possam identificar os problemas.

Nesta última etapa, utilizou-se a seguinte estratégia para apresentar os exemplos aos estudantes. Iniciamos apresentando exemplos com erros simples para demonstrar os principais fundamentos da modelagem de software. Em seguida, foi feita uma nova modelagem corrigindo os erros encontrados e de forma a explicar também os diferentes elementos e relacionamentos representados no modelo. Em seguida, foi apresentado um novo diagrama incorreto, só que agora com um nível mais alto de complexidade, e novamente foi feita uma reflexão sobre o problema com os estudantes e desenvolveu-se um novo modelo correto. À medida que os estudantes se familiarizavam com os fundamentos da modelagem de software, era introduzido exemplos errôneos mais complexos que envolviam vários componentes interativos ou técnicas de modelagem mais avançadas. Esse processo foi repetido mais duas vezes durante a aula. Em seguida, os estudantes foram incentivados a trabalhar com dois exemplos errôneos em grupos, ajudando-os a desenvolver suas próprias habilidades de resolução de problemas e obter uma compreensão mais profunda do conceito e sobre os exemplos errôneos vistos em sala.

Após resolverem os problemas, o professor proveu uma orientação e *feedback* enquanto os estudantes trabalhavam com os exemplos e os incentivava a fazer perguntas se precisarem de ajuda. Isso ajudou a aumentar a confiança dos estudantes e a mantê-los motivados para aprender.

3.2. Coleção de dados

Após realizarem as atividades de modelagem, todos os estudantes foram convidados para participarem de uma sessão de Grupo Focal [Kontio et al. 2004] que foi conduzido na próxima aula. No primeiro passo, para incentivar os estudantes a apresentarem suas opiniões negativas, usou-se uma dinâmica de *Lovers e Haters* [Silva et al. 2019]. No início da sessão, os papéis de cada equipe (*Lovers* ou *Haters*) foram definidos de forma aleatória. Depois disso, cada estudante usou notas adesivas coloridas para escrever as dificuldades, facilidades, pontos positivos, negativos ou dúvidas que eles tinham ao criar o diagrama empregando o ErrEx. Em seguida, cada estudante de cada equipe colocou suas anotações em um quadro e as explicou. O moderador pediu aos outros estudantes que comentassem se concordavam ou não com a opinião do colega. A sessão durou uma hora e todas as informações desta sessão foram gravadas em vídeo pelo moderador com

o consentimento dos participantes presentes.

3.3. Análise dos dados

Analisou-se qualitativamente a transcrição das respostas seguindo os procedimentos de codificação. O objetivo da análise qualitativa foi codificar, categorizar e sintetizar dados, a fim de identificar as dificuldades e os benefícios percebidos pelos estudantes após o uso da estratégia ErrEx. A análise foi conduzida por um pesquisador e discutida com os outros pesquisadores em sucessivas reuniões.

4. Resultados

A partir da análise qualitativa, foram identificados benefícios e dificuldades sobre o uso do ErrEx. A Tabela 1 apresenta as categorias e as evidências (principais citações) relacionadas à categoria de benefícios identificados.

Tabela 1. Benefícios percebidos pelos estudantes usando o ErrEx.

ErrEx ajudou a identificar os próprios erros durante a modelagem “O ErrEx permitiu que eu observasse minhas próprias falhas na hora da modelagem.” – P02 “(Com o ErrEx), pude identificar os possíveis erros que havia cometido no meu diagrama.” – P09
ErrEx auxiliou na compreensão dos conceitos sobre o diagrama de Casos de Uso “Usando a estratégia pude entender melhor os conceitos de <i>include</i> e <i>extend</i> (...) sempre ficava na dúvida entre esses dois e muitas vezes me confundia na hora da modelagem” – P15 “(O ErrEx) me ajudou a retornar e compreender muitos dos conceitos do diagrama de casos de uso e ainda entendi que o que achei que era certo, na verdade, não era” – P16
ErrEx auxiliou a tirar dúvidas sobre a modelagem do diagrama de Casos de Uso “Teve um momento que a gente não sabia como fazer, aí a gente olhou um exemplo negativo e vimos como não tinha que fazer, nos baseamos nisso e continuamos a modelagem.” – P44 “(…) o ErrEx é interessante, me baseava bastante nos exemplos quando estava em dúvida.” – P28
Identificar os erros evitou que estes fossem repetidos durante a nova modelagem. “Visualizando o diagrama que está errado é fácil de evitar cometer o mesmo erro na nova modelagem” – P07 “(…) levamos um bom tempo analisando os erros do diagrama e tentamos evitá-los no nosso diagrama.” – P34
ErrEx melhorou o trabalho em equipe “O ErrEx também permitiu trabalhar em equipe e ouvir opiniões distintas sobre a modelagem” – P14 “(…) em grupo discutimos muitos pontos enquanto realizávamos a atividade” – P31

Com base nas categorias de feedback identificadas no estudo, algumas discussões emergiram sobre o uso do ErrEx para o ensino e a aprendizagem de modelagem de software. Por exemplo, a categoria “**ErrEx ajudou a identificar os próprios erros durante a modelagem**” sugere que os ErrEx podem ser eficazes para promover a autorreflexão dos estudantes e o desenvolvimento da metacognição. A partir do momento em que os estudantes são capazes de identificar seus próprios erros e falhas, eles podem começar a trabalhar para corrigi-los e melhorar sua compreensão do assunto. Além disso, a categoria “**ErrEx auxiliou a tirar dúvidas sobre a modelagem do diagrama de Casos de Uso**” sugere que os exemplos errôneos podem ser úteis para ajudar os estudantes a superar dificuldades específicas na aprendizagem da modelagem de software. Os exemplos fornecidos pelo ErrEx parecem ter sido especialmente valiosos para os estudantes que estavam enfrentando dúvidas ou incertezas sobre como proceder em determinadas situações.

Outra discussão que emergiu com base nas categorias é sobre a eficácia do ErrEx em promover a aprendizagem colaborativa e a melhoria do trabalho em equipe. A categoria “**ErrEx melhorou o trabalho em equipe**” sugere que os exemplos podem ser úteis para ajudar os estudantes a colaborar de forma mais eficaz e discutir diferentes pontos

de vista e abordagens para a modelagem de software. Isso pode ser particularmente valioso em um ambiente de aprendizagem onde os estudantes estão trabalhando em grupos para concluir projetos complexos. Além disso, a categoria “**ErrEx auxiliou na compreensão dos conceitos sobre o diagrama de Casos de Uso**” sugere que a ferramenta pode ser eficaz para ajudar os estudantes a consolidar sua compreensão de conceitos-chave na modelagem de software. Ao entender melhor os conceitos e princípios por trás da modelagem de software, os estudantes podem ser capazes de aplicar seu conhecimento de forma mais eficaz e criar modelos mais precisos e completos.

Os códigos relacionados às categorias mostram indícios de que os estudantes tiveram uma boa percepção sobre o ErrEx. Contudo, foram identificadas algumas dificuldades percebidas pelos estudantes, em que estes enfatizam que o ErrEx não ajudou na aprendizagem durante a modelagem do diagrama de casos de uso. A Tabela 2 apresenta as categorias e subcategorias identificadas sobre as dificuldades, além de apresentar as evidências (principais citações) relacionadas às subcategorias.

Tabela 2. Dificuldades percebidas pelos estudantes usando o ErrEx.

<p>Os exemplos errados causaram dúvidas durante a modelagem “Os exemplos errados não facilitaram a modelagem, porque os exemplos só me causavam dúvidas, eu já tinha dúvida e os exemplos não me auxiliavam.” – P12 “(...) não achei que foi muito útil, porque a gente começou a discutir sobre a modelagem e surgiram muitas dúvidas durante a prática.” – P20</p>
<p>ErrEx não auxiliou a modelar o cenário no diagrama “Na minha opinião, é bem melhor resolver o problema primeiro sem olhar os erros. Assim, consigo pensar melhor em como modelar o cenário no diagrama.” – P25 “Ajudou a identificar erros no meu diagrama, mas não me ajudou muito a modelar um novo diagrama.” – P38</p>
<p>Necessidade de um conhecimento prévio por parte dos estudantes “Como não vim na aula anterior, fiquei um pouco perdida sobre como fazer a atividade (...) acho que a gente precisa de uma boa base de conhecimento sobre o assunto, antes de usar ele na modelagem.” – P33</p>

As citações mencionadas indicam algumas limitações do uso do ErrEx na modelagem de diagramas de casos de uso. Primeiramente, os exemplos errados fornecidos pelo ErrEx causaram dúvidas e confusão aos estudantes durante a modelagem. Isso pode ser um problema para estudantes que ainda estão aprendendo os conceitos básicos de modelagem de casos de uso, pois exemplos errados podem prejudicar o entendimento correto desses conceitos. Além disso, alguns estudantes relataram que o ErrEx não foi útil para ajudá-los a modelar um novo diagrama, sugerindo que a ferramenta pode não ser tão eficaz em termos de orientação prática na criação de novos diagramas.

Outra limitação destacada pelas citações é a necessidade de um conhecimento prévio dos conceitos de modelagem de casos de uso para aproveitar plenamente o ErrEx. Alguns estudantes relataram que, sem uma boa base de conhecimento sobre o assunto, o ErrEx não foi útil para eles durante a atividade de modelagem. Isso sugere que a ferramenta pode ser mais adequada para estudantes que já têm algum conhecimento prévio sobre modelagem de casos de uso. Portanto, embora o ErrEx possa ser uma ferramenta útil para ajudar os estudantes a identificar e corrigir erros em diagramas de casos de uso, ele pode não ser suficiente para fornecer orientação prática para estudantes iniciantes na modelagem de casos de uso.

Além disso, os estudantes propuseram algumas sugestões sobre como aplicar o ErrEx de forma mais eficiente para o aprendizado (ver Tabela 3).

Tabela 3. Sugestões propostas pelos estudantes.

Sessão de dúvidas durante a aplicação do ErrEx “Talvez se tivesse, entre cada sessão um tempo para tirar dúvidas, nem que sejam as dúvidas mais críticas que realmente estão impedindo a gente de continuar, aí talvez isso ajudaria” – P41
Prática de modelagem professor-estudante utilizando o ErrEx “Se o professor pegasse um problema e fosse montando o diagrama para a gente ver como funciona seria muito melhor e agregaria mais para a gente entender o ErrEx.” – P08

As citações dos estudantes abordam pontos importantes relacionados à aplicação do ErrEx em sala de aula. A primeira categoria evidencia a importância da sessão de dúvidas durante a utilização de ErrEx. O estudante destaca que, se houvesse um tempo específico para a resolução das dúvidas mais críticas, isso poderia ajudar a turma a avançar na atividade. Isso indica a necessidade de que o professor esteja disponível e preparado para responder às dúvidas dos estudantes, tornando o processo mais produtivo e facilitando a compreensão dos conceitos envolvidos.

A segunda categoria aponta para a necessidade de práticas mais interativas e colaborativas entre professor e estudantes durante a utilização do ErrEx. A sugestão é que o professor demonstre como utilizar o método, realizando a modelagem de um problema para a turma, facilitando o entendimento dos estudantes. Essa abordagem pode ser importante, principalmente para estudantes com menos experiência ou dificuldades em visualizar a aplicação dos conceitos. A prática colaborativa pode incentivar a participação ativa dos estudantes e promover um ambiente de aprendizado mais engajador e efetivo.

5. Lições Aprendidas

Com base nas citações, fica evidente que a utilização de exemplos errôneos durante as aulas de modelagem pode gerar dúvidas e confusões nos estudantes. Dessa forma, uma dica valiosa aos docentes que desejam utilizar esse tipo de estratégia é ter cuidado ao selecionar os exemplos, buscando sempre utilizá-los de forma estratégica e consciente.

Uma possibilidade é apresentar os exemplos errados de forma que os estudantes possam identificar os erros e compreender melhor os conceitos da modelagem, mas sem utilizar esses exemplos como única referência para a construção dos seus próprios diagramas. Além disso, é importante que o docente esteja disponível para esclarecer as dúvidas que possam surgir durante o processo, oferecendo apoio e orientação aos estudantes. Apesar de, em um primeiro momento, o docente ter feito a atividade com os estudantes, quando estes estavam conduzindo as atividades de modelagem com seus pares, notou-se também a necessidade da presença mais ativa do docente neste momento. Com isso, o docente nota que isso está potencialmente relacionado ao fato de que essa atividade requer que os estudantes recebam *feedback* constante durante a utilização de exemplos com erros. Segundo Cooper (2005), o *feedback* é essencial para que os estudantes entendam onde os modelos precisam ser melhorados e, principalmente, porque os modelos são considerados de alta qualidade ou não.

Outra possibilidade é utilizar exemplos errôneos de forma mais pontual, como forma de ilustrar as consequências de um erro específico na construção do diagrama. Nesse caso, é importante que o docente apresente também exemplos corretos e explique claramente a diferença entre ambos, para que os estudantes possam compreender a lógica por trás da modelagem e utilizá-la de forma efetiva em suas próprias atividades.

Nossos resultados corroboram os resultados de Cooper *et al.* (2005), Hilburn *et al.* (2011) e Silva *et al.* (2017). Nesses estudos, foram identificados vários aspectos positivos do uso de exemplos no Ensino de Engenharia de Software, como por exemplo, a redução do número de defeitos nos modelos criados pelos estudantes. Os autores perceberam que, após o uso de exemplos em aulas, os estudantes conseguiram entender melhor os conceitos de modelagem e que a identificação de erros evita que os estudantes os repitam novamente em modelagens futuras. Também observou-se que os estudantes refletiram sobre os erros e que isso melhora suas habilidades de modelagem de software. Ao empregar uma atividade com exemplos errôneos, pode-se fazer os estudantes refletirem sobre dois princípios fundamentais da aprendizagem: o fazer e a reflexão. A reflexão pode ser entendida como uma forma de pensar que envolve sair de uma situação para criar conhecimento. A reflexão é um tópico significativo para a educação em engenharia devido à importância da reflexão para a prática profissional [Schön 2017], a conexão da reflexão com a metacognição e o papel da reflexão na aprendizagem a partir da experiência [Dewey 1986, Kolb 2014]. Acredita-se que usando exemplos errôneos de forma eficaz, os professores podem ajudar seus estudantes a aprender e compreender novos conceitos e técnicas de forma mais eficaz.

Com base na experiência obtida neste trabalho, são apresentadas dicas para professores que pretendem aplicar exemplos errôneos em suas aulas de modelagem:

- Antes de apresentar os exemplos errôneos, verifique se os estudantes possuem conhecimento e compreensão sobre o assunto a ser ensinado. Isso ajuda os estudantes a construir uma base sólida, compreender melhor conceitos complexos e aprimorar suas habilidades de pensamento crítico. Por outro lado, a introdução precoce de conceitos complexos por meio dos exemplos sobrecarrega os estudantes, leva à confusão e à incompreensão, dificultando o processo de aprendizagem.
- Explique claramente para os estudantes qual é o propósito de utilizar exemplos errôneos na melhoria da aprendizagem de modelagem de software.
- Encoraje os estudantes a identificar os erros e explicar à turma ou aos colegas por que eles estão errados. Assim, ao compreenderem seus erros, os estudantes podem evitar cometer os mesmos erros no futuro, melhorando sua experiência geral de aprendizagem.
- Incentive os estudantes a fazerem perguntas e tirarem dúvidas durante a aplicação dos exemplos errôneos. Isso pode levar a uma compreensão mais profunda do assunto e a uma experiência de aprendizagem mais envolvente para os estudantes.
- Para melhorar a compreensão e a capacidade de aprendizagem, os docentes devem ajudar os estudantes a corrigir erros de modelagem que ocorrem com mais frequência durante a elaboração de diagramas.
- Reserve um tempo para discutir os erros e ajudar os estudantes a entender por que estão errados. Isso pode ajudar os estudantes a se sentirem mais confortáveis em admitir seus erros e mais motivados para aprender com eles.
- Discuta as implicações dos erros, como eles podem afetar a qualidade do software e como podem ser corrigidos.
- Incentive a prática de modelagem por meio de exemplos e/ou projetos reais. Assim, eles poderão aplicar os conhecimentos adquiridos na resolução de problemas.
- Esteja aberto a *feedbacks* e críticas dos estudantes sobre a aplicação dos exemplos errôneos, para que você possa ajustar sua abordagem para melhorar a experiência

de aprendizado.

Entende-se que cada situação de ensino é diferente, então os docentes devem adaptar as dicas, ou até criar e definir novas dicas, conforme necessário para atender às necessidades e objetivos de aprendizagem específicos dos estudantes e das turmas.

6. Considerações finais e Trabalhos futuros

Este artigo apresentou um relato de experiência, sob a perspectiva dos estudantes, que coletou as percepções dos estudantes para compreender como o uso do ErrEx pode apoiar no processo de aprendizagem durante a modelagem de software com diagramas da UML, mais especificamente diagrama de casos de uso. Para isso, foram coletados dados qualitativos por meio de sessões de Grupo Focal realizadas com os estudantes.

A partir dos resultados, percebeu-se que os estudantes entenderam melhor os conceitos de modelagem e que os exemplos errados evitam que estudantes os repitam novamente em futuras modelagens. Quando se combina estratégias focadas nas estratégias de “fazer” (*doing*) e “reflexão” (*reflection*), os estudantes tendem a alcançar objetivos de aprendizado com mais eficácia durante a aprendizagem. No entanto, os estudantes também relataram que o ErrEx não auxiliou a modelar um novo diagrama. O ErrEx não deve ser utilizado para ensinar o passo a passo da modelagem do diagrama, mas para enfatizar os principais erros que os estudantes cometem na modelagem, como enfatizado por P11: “(...) *trata-se de uma outra maneira de encarar o problema e apesar de não ajudar muito na modelagem e criação do diagrama, este método ajuda a verificar se o diagrama está bem certo ao mostrar os erros, ou seja, o que não fazer.*”

Os benefícios e dificuldades identificados podem ajudar os professores na seleção dessa estratégia como forma de apoiar os estudantes a alcançar um determinado objetivo de aprendizagem. Por exemplo, se o professor desejar que os estudantes alcancem as primeiras habilidades sobre o entendimento de um objetivo de aprendizagem específico, o ErrEx pode ser empregado. Ressalta-se ainda que com base nos resultados alcançados foram sugeridas algumas dicas úteis sobre como usar exemplos errôneos com seus estudantes. Por fim, acredita-se que o uso de exemplos errados em sala de aula pode ser uma ferramenta útil para ajudar os estudantes a aprender mais sobre modelagem de software e aumentar seu envolvimento e sua motivação no conteúdo. Como trabalhos futuros, pretende-se realizar novos estudos com o objetivo de comparar o ErrEx com outras estratégias de ensino (por exemplo, *Project-Based Learning*) para entender como esses métodos se comportam no processo de ensino de outros diagramas UML.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Williamson Silva agradece pelo apoio financeiro da FAPERGS (Projeto ARD/ARC – processo 22/2551-0000606-0) e FAPEMIG (Projeto APQ-00743-22).

Referências

Adams, D. M., McLaren, B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., and Van Velsen, M. (2014). Using erroneous examples to improve mathematics

- learning with a web-based tutoring system. *Computers in Human Behavior*, 36:401–411.
- Alistair, C. (2001). Writing effective use cases. *Michigan: Addison-Wesley*.
- Anda, B., Hansen, K., and Sand, G. (2009). An investigation of use case quality in a large safety-critical software development project. *Information and Software Technology*, 51(12):1699–1711.
- Anda, B. and Sjøberg, D. I. (2002). Towards an inspection technique for use case models. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 127–134. ACM.
- Balaban, M., Maraee, A., Sturm, A., and Jelnov, P. (2015). A pattern-based approach for improving model quality. *Software & Systems Modeling*, 14(4):1527–1555.
- Bolloju, N., Schneider, C., and Vogel, D. (2011). Asymmetrical effects of using positive and negative examples on object modeling. In *Information Systems Development*, pages 85–96. Springer.
- Cooper, K., Liddle, S., and Dascalu, S. (2005). Experiences using defect checklists in software engineering education. In *Proceedings of the ISCA 18th International Conference on Computer Applications in Industry and Engineering (CAINE 2005)*, pages 402–409.
- Curry, L. A. (2004). The effects of self-explanations of correct and incorrect solutions on algebra problem-solving performance. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.
- Dewey, J. (1986). Experience and education. In *The Educational Forum*, volume 50, pages 241–252. Taylor & Francis.
- Große, C. S. and Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and instruction*, 17(6):612–634.
- Hilburn, T. B., Towhidnejad, M., and Salamah, S. (2011). Read before you write. In *Proceedings of the 24th Conference on Software Engineering Education and Training (CSEE&T)*, pages 371–380. IEEE.
- Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.
- Kontio, J., Lehtola, L., and Bragge, J. (2004). Using the focus group method in software engineering: obtaining practitioner and user experiences. In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.*, pages 271–280. IEEE.
- Ma, Z. (2017). An approach to improve the quality of object-oriented models from novice modelers through project practice. *Frontiers of Computer Science*, 11(3):485–498.
- McLaren, B. M., van Gog, T., Ganoë, C., Karabinos, M., and Yaron, D. (2016). The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior*, 55:87–99.

- Misbhauddin, M. and Alshayeb, M. (2015). Extending the uml use case metamodel with behavioral information to facilitate model analysis and interchange. *Software & Systems Modeling*, 14(2):813–838.
- Nascimento, E., Silva, W., Conte, T., Steinmacher, I., Massollar, J., and Travassos, G. H. (2016). Is a picture worth a thousand words?: A comparative analysis of using textual and graphical approaches to specify use cases. In *Proceedings of the 30th Brazilian Symposium on Software Engineering*, pages 93–102. ACM.
- Nascimento, E. S., Silva, W. A. F., de França, B. B. N., Gadelha, B., and Conte, T. U. (2017). A model on the difficulties to specify use cases. In *Proceedings of the XX Ibero-American Conference on Software Engineering (CIBSE)*, pages 263–276.
- Oliveira, M., Ribeiro, L., Cota, É., Duarte, L. M., Nunes, I., and Reis, F. (2015). Use case analysis based on formal methods: an empirical study. In *International Workshop on Algebraic Development Techniques*, pages 110–130. Springer.
- Schön, D. A. (2017). The reflective practitioner: How professionals think in action. *Routledge*.
- Shmallo, R. and Shrot, T. (2020). Constructive use of errors in teaching the uml class diagram in an is engineering course. *Journal of Information Systems Education*, 31(4):282.
- Silva, W., Steinmacher, I., and Conte, T. (2019). Students' and instructors' perceptions of five different active learning strategies used to teach software modeling. *IEEE Access*, 7:184063–184077.
- Silva, W. A. F., Steinmacher, I. F., and Conte, T. U. (2017). Is it better to learn from problems or erroneous examples? In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, pages 222–231. IEEE.
- Skudder, B. and Luxton-Reilly, A. (2014). Worked examples in computer science. In *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*, pages 59–64.
- Tiwari, S. and Gupta, A. (2013). A controlled experiment to assess the effectiveness of eight use case templates. In *Proceeding of the 20th Asia-Pacific Software Engineering Conference (APSEC), 2013*, volume 1, pages 207–214. IEEE.
- Tiwari, S. and Gupta, A. (2015). A systematic literature review of use case specifications research. *Information and Software Technology*, 67:128–158.
- Tsovaltzi, D., Melis, E., McLaren, B. M., Meyer, A.-K., Dietrich, M., and Gogvadze, G. (2010). Learning from erroneous examples: when and how do students benefit from them? In *European conference on technology enhanced learning*, pages 357–373. Springer.
- Van Gog, T. and Rummel, N. (2018). Example-based learning. In *International Handbook of the Learning Sciences*. Routledge.