




Evidences on the Educational Support offered by a Conversational Agent in Software Testing Lessons

Leo Natan Paschoal  AND Vânia de Oliveira Neves AND Silvana Morita Melo AND Tayana Uchôa Conte 
AND and Simone do Rocio Senger de Souza 

Resumo—Software testing is an activity that contributes to software quality. Nevertheless, little attention is still paid to software testing in the formal education. Consequently, the software industry is made up of a limited number of professionals who have an understanding of criteria that systematize and improve the quality of the test. In an attempt to remedy these problems and promote software testing, professors in the field have established Massive Open Online Course as educational initiatives. However, the main limitation of this type of initiative is the lack of supervision of students by the professor. To minimize the problem, a conversational agent called TOB-STT was defined. A study carried out introduces the conversational agent TOB-STT, however, it does not analyze the effectiveness of the support of this teaching support mechanism. Therefore, in this article, we present a controlled experiment that was intended to analyze the effectiveness of the support offered by TOB-STT. Based on the results, it was possible to recognize that the support of the conversational agent, in its current version, offered to software test students was unexpressive.

Index Terms—Chatbot, Software Engineering Educacion, Software testing.

I. INTRODUÇÃO

Teste de software é uma atividade de desenvolvimento de software que visa promover a qualidade do software, a partir do reconhecimento dos defeitos que persistem no software sob teste [1]. O custo da presença de um defeito pode ser muito alto para uma organização, porque defeitos oportunizam efeitos não desejados tanto para as organizações que desenvolvem quanto para as que mantêm o software, dado que podem possibilitar falhas de segurança, perda de dados e informações, danos ao ambiente, prejuízos financeiros, dentre outros [2], [3]. Diante disso, entende-se que para evitar tais problemas, a execução dessa atividade não pode ser dispensada.

Ainda que a importância do teste de software seja reconhecida por uma parte significativa da comunidade de engenharia de software, essa atividade é comumente negligenciada pelos currículos de computação [4]. A atenção dada pelos currículos de referência da ACM [5] e IEEE-CS [6] faz com que o teste de software seja geralmente incluído como uma unidade dentre os vários tópicos de uma disciplina de engenharia de software [7]. Como consequência, abordagens mais completas de teste deixam de ser ensinadas [8]. Para complementar, devido a quantidade de conteúdos a serem abordados no âmbito da engenharia de software e o limite da carga-horária dessa disciplina, a prática de teste é colocada à parte [8]. Todavia, o aprendizado de teste de software requer mais prática do que comumente é abordado [4].

A falta de atenção direcionada ao ensino de teste faz com que os estudantes finalizem seus cursos de graduação em

computação sem saber teste de software [9]. Diante disso, as organizações de software acabam contratando profissionais sem habilidades com essa temática. De acordo com Beneditti [7] a ausência de profissionais qualificados e responsáveis pela atividade de teste no contexto da indústria de software pode ser um dos principais motivos pelos quais as organizações de software ainda não possuem um processo maduro para realizar essa atividade. Diversos estudos corroboram com essa opinião, mencionando que os profissionais são contratados pela indústria sem ter a formação universitária adequada, assim como, treinamento suficiente [10], [11], [12].

A contratação de sujeitos que não possuem um entendimento adequado sobre a prática de teste, sem as competências necessárias, contribui com o estado atual da prática de teste que é realizada por uma quantidade expressiva de organizações de software. O estado atual, exteriorizado por surveys conduzido em diferentes países nos quais os participantes são profissionais de teste da indústria, caracteriza-se majoritariamente pelos testes (ou casos de teste) sendo definidos aleatoriamente para exercitar os sistemas de software sob teste [10], [11], [12], [13]. Portanto, acredita-se que a falta de conhecimento na área de teste tem sido um fator predominante para a organizações deixarem de adotar critérios que auxiliam na seleção eficaz de conjuntos de casos de teste [7].

Para os profissionais realizarem o seu trabalho em teste de software, ou até mesmo conquistarem melhores oportunidades de emprego, os graduados acabam procurando formas alternativas para adquirir conhecimentos na área. Algumas estratégias de aprendizado que têm sido utilizadas por esses profissionais são os recursos disponíveis *online* [10]. Dentre os recursos, encontram-se os cursos estabelecidos no âmbito de plataformas MOOC (*Massive Open Online Course*) [14], [15], que são alternativas gratuitas oferecidas por instituições de ensino reconhecidas. Esses cursos podem atender algumas demandas existentes e aperfeiçoar o desenvolvimento de algumas competências dos aprendizes.

Em especial no domínio de teste de software, os cursos de MOOCs podem ser vistos como iniciativas para os profissionais e organizações melhorarem suas habilidades e conhecer técnicas e critérios de teste que auxiliam na definição de conjuntos de casos de teste [16]. Além disso, Enou [16] cita que os MOOCs de teste de software também podem ser vistos como uma oportunidade para os professores disseminarem no contexto industrial as técnicas e tecnologias definidas ou abordadas em suas pesquisas, apoiando também a inovação nas empresas de tecnologia de informação. Diante disso, pode-se assumir que os MOOC têm potencial para contribuir com

o aprendizado de teste no contexto de ensino não formal.

Apesar da potencialidade, os MOOCs têm suas limitações. Estudos recentes que abordam as barreiras existentes no âmbito de MOOCs, mencionam que a falta de interação e feedback tem sido o principal fator para promover a evasão dos alunos nesses cursos [17], [18], [19]. Isso acontece porque os cursos disponíveis nessas plataformas são acessados por muitos estudantes e raramente há suporte de professores ou tutores. Os estudos que alavancam esse tipo de discussão também mencionam sobre o uso de agentes conversacionais¹ como recurso de apoio aos MOOC. Nesse sentido, são uma alternativa para proporcionar interação, no qual os alunos matriculados nos cursos possam interagir com os agentes, solucionar suas dúvidas e receberem feedback adequado. É nesse contexto que surgiu o agente conversacional denominado TOB-STT, um mecanismo de apoio ao ensino de teste de software [21].

O TOB-STT foi apresentado como um software capaz de interagir em língua natural inglesa com estudantes de teste de software e solucionar suas dúvidas sobre assuntos associadas a conceitos, critérios e técnicas de teste de software [21]. Apesar do estudo anterior apresentar o agente conversacional e discutir sobre um estudo de viabilidade com alunos de teste de software, ele limitou-se a compreender se as bases de conhecimento do agente conversacional eram viáveis para o seu objetivo. Portanto, um experimento controlado que considera um grupo controle para comparar com os resultados de um grupo experimental ainda não havia sido realizado. Antes do agente conversacional ser considerado em uma implantação efetiva de um MOOC de teste de software, é preciso avaliar se ele é capaz de atingir satisfatoriamente o seu objetivo. Como ainda não há evidência suficiente, uma avaliação experimental surge como uma importante etapa para coletar evidências. Diante disso, neste artigo é apresentado um estudo experimental que visa identificar a eficácia do apoio oferecido pelo agente conversacional TOB-STT ao estudante de teste de software.

Para apresentar este estudo, o artigo foi organizado da seguinte forma. A Seção 2 apresenta uma descrição sobre o TOB-STT. Estudos sobre agentes conversacionais no ensino de computação, similares ao apresentado no decorrer deste estudo, são abordado na Seção 3. Na Seção 4, o planejamento do estudo experimental é reportado. Durante a Seção 5, os resultados são discutidos. As ameaças à validade do estudo são descritas ao longo da seção 6. Na Seção 7, algumas limitações do estudo são elencadas. Por fim, as considerações finais são apresentadas.

II. TOB-STT: UMA VISÃO GERAL

TOB-STT é um agente conversacional baseado em regras que possui bases de conhecimento sobre conteúdos de teste de software, escritos em uma linguagem denominada AIML (*Artificial Intelligence Markup Language*²). Ele foi definido

para subsidiar as atividades do domínio de teste de software que estão condicionadas ao modo de ensino à distância. Nessa perspectiva, ele possui uma interface similar a um bate-papo, que será usada pelo aluno para exteriorizar dúvidas e opiniões. Nessa mesma interface, o aluno receberá o feedback do agente conversacional para suas incertezas e convicções. Nessa perspectiva, esse mecanismo de apoio ao ensino faz uso de língua natural para se comunicar e realiza tratamentos linguísticos nas requisições que são enviadas pelo aluno para produzir um feedback apropriado para o aluno.

O agente foi definido para suportar os três diferentes estágios de uma conversa. Portanto, o aluno poderá se apresentar, cumprimentar o agente (fase de início da conversação), interagir sobre o conteúdo de teste de software (desenvolvimento da conversação) e se despedir do agente (encerramento da conversação). Em relação à interação sobre o domínio ao qual ele foi estabelecido, ele é capaz de reconhecer especificamente quatro tipos de intenção. Cada uma dessas intenção é apresentada na Tabela I, juntamente com sua definição e alguns exemplos de diálogos que são aceitos pelo agente conversacional. Nessa perspectiva, após compreender o que o aluno quer saber, o agente emite uma mensagem direcionada àquela requisição.

Informações técnicas associadas aos algoritmos e tecnologias usadas no estabelecimento do TOB-SST foram reportadas por Paschoal et al. [21]. Vale ressaltar que a versão atual do TOB-STT está disponível para uso no seguinte endereço: <<https://bit.ly/3kgExR3>>. Como ele foi concebido em formato de software livre, buscando estimular a comunidade a contribuir com o seu desenvolvimento, seus códigos-fontes estão disponíveis para uso e modificação pela comunidade de engenharia de software, em: <<http://bit.ly/2ZkWQdJ>>. Adicionalmente, acredita-se que a distribuição em licença livre/aberta flexibiliza a evolução regular do material de ensino e facilita o alcance à educação. Considerando essas perspectivas, assume-se que o agente conversacional está alinhado ao movimento dos recursos educacionais abertos da UNESCO (*United Nations Educational, Scientific and Cultural Organization*)³.

Para contemplar os diferentes modos de acesso ao conteúdo educacional e paradigmas educacionais (*i.e.*, aprendizagem móvel e aprendizagem ubíqua), o TOB-STT foi estabelecido para se adequar à diferentes interfaces de dispositivos (*e.g.*, *desktop*, *tablets*, *smartphones*). A interface do agente conversacional é apresentada na Figura 1. A Figura 1a, ainda, ilustra um exemplo de diálogo entre o TOB-SST e um aluno. Nesse exemplo, traduzido do inglês, o aluno começa a interação perguntando para o TOB-STT o que é um “bug”. Como resposta, o agente apresenta uma explicação considerando uma definição existente no SWEBOK (*Software Engineering Body of Knowledge*). Além disso, o TOB-STT tenta deixar claro que um “bug” tem a mesma definição de um defeito. Na sequência, o aluno solicita ao agente para que ele faça uma diferenciação entre defeito e erro, finalizando sua interação pedindo ao agente que ele explique como aplicar o critério de teste conhecido como particionamento em classes de equivalência.

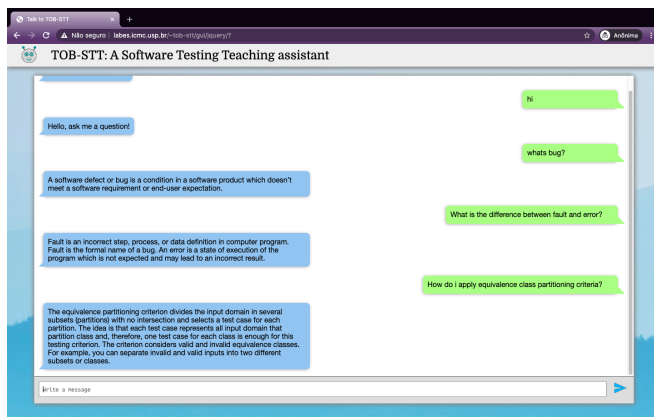
¹Os agentes conversacionais são sistemas que interagem com seus usuários na língua natural humana, através de processamento de linguagem natural [20].

²Uma descrição completa dessa linguagem é apresentada por Wallace [22].

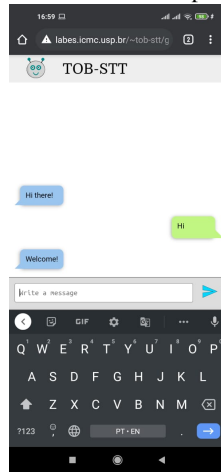
³Mais informações disponíveis em: <<https://bit.ly/3iwhqS8>>.

Tabela I: Conjunto de intenções reconhecidas pelo TOB-STT

Intenção	Descrição da intenção	Exemplos
Definir	Essa intenção se caracteriza por permitir ao TOB-STT reconhecer perguntas ou comentários que estejam relacionados à determinação de algum termo, conceito ou jargão da área de teste de software	O que é teste de software? Você pode descrever os critérios do teste funcional?
Confrontar	Essa intenção possibilita ao TOB-STT identificar perguntas ou comentários que possuem uma natureza associada ao efeito de diferenciar dois ou mais termos, conceito ou jargões da área de teste de software	Qual a diferença entre teste funcional e teste estrutural? Faça um paralelo entre defeitos, erros e falhas
Aplicar	Essa intenção permite ao TOB-STT detectar perguntas ou comentários que têm relação ao uso e demonstração de algum critério de teste de software	Como aplicar o critério análise de valor limite? Explique como eu devo usar o critério particionamento em classes de equivalência
Exemplificar	Essa intenção oportuniza ao TOB-STT distinguir perguntas ou comentário que solicitam exemplos reais de uso das técnicas e critérios de teste de software	Apresente exemplos de uso do critério análise do valor limite Cite exemplos de aplicações do critério particionamento em classes de equivalência



(a) TOB-STT sendo acessado por um desktop



(b) TOB-STT sendo acessado por um smartphone

Figura 1: Interface do TOB-STT

O TOB-STT é semelhante a outros agente conversacionais apresentados na literatura. Para contextualizar o mecanismo mencionado no decorrer desta seção frente ao que existe de estudos que envolvem esta temática, uma seção foi organizada e é apresentada a seguir.

III. AGENTES CONVERSACIONAIS NO ENSINO DE COMPUTAÇÃO

A exploração de agente conversacional no domínio de computação, mais especificamente no contexto de práticas educacionais em computação não é algo novo. Estudos anteriores revelam esforços de pesquisadores ao propor soluções educacionais baseadas em agentes conversacionais para problemas de ensino de computação, na educação formal e no treinamento de profissionais da área. Nesse sentido, existem agentes conversacionais estabelecidos para diferentes contextos, ambientes e disciplinas da Computação. No decorrer desta seção, alguns trabalhos que se assemelham ao presente estudo serão abordados de modo a caracterizar o estudo em relação às pesquisas e produtos existentes.

Em Katchapakirin e Anutariya [23], a necessidade do estabelecimento de um agente conversacional surgiu ao passo que a Tailândia adotou conteúdos de computação na educação básica e no ensino médio. Em meio a adoção, gestores reconheceram que não possuíam recursos humanos qualificados na área de computação para atuar como professores e ensinar determinados conteúdos aos alunos. Em particular, a escassez de profissionais concentrou-se na temática da linguagem de programação visual baseada em blocos Scratch⁴. Assim, para mitigar o problema foi desenvolvido o ScratchThAI, um agente conversacional que apoia os alunos por meio de conversas textuais. Os auxílios são feitos visando estimular o desenvolvimento de habilidades de pensamento computacional.

De modo similar ao estudo de Katchapakirin e Anutariya, o trabalho de Ocaña et al. [24] aborda o agente conversacional para apoiar alunos que não estão estudando Computação no ensino superior, mas estão submetidos a iniciativas associadas ao aprendizado de programação em níveis básico de ensino. Em particular, em Ocaña et al. [24] o agente conversacional é tratado como um companheiro de aprendizagem em um ambiente projetado para crianças aprenderem programação. Nesse estudo, o agente conversacional se encarregar de pedir aos alunos que escrevam seus programas e, caso seja necessário, façam a depuração. Ele foi projetado para possibilitar que os alunos pratiquem conceitos de programação com programas

⁴Mais informações disponíveis em: <<http://bit.ly/3qd28oJ>>.

reais e com feedback imediato. Conforme os autores do estudo, o feedback imediato é usado para apoiar o aluno a depurar o programa.

Também é possível localizar estudos que mencionam agentes conversacionais sendo construídos para o ensino de redes de computadores. Em Herpich et al. [25], é apresentado ELAI, um agente conversacional disponível em um mundo virtual que simula um laboratório dedicado ao ensino de redes de computadores. O agente é representado através de um NPC (*Non-player character*) e tem o papel de solucionar as dúvidas dos alunos sobre os conteúdos presentes nesse laboratório.

No estudo de Leonhardt et al. [26] é apresentado um agente conversacional como ferramenta para o treinamento do gerenciamento de redes de computadores. A ideia é que o agente conversacional sirva de apoio à profissionais com pouca experiência. Segundo os autores, os profissionais com pouca experiência tendem a ter menor compreensão sobre detalhes de protocolos de gerenciamento e o agente conversacional surge para ensinar esse profissional a como proceder para encontrar uma dada informação. Além disso, o agente foi definido para explicar conceitos do domínio.

No âmbito do ensino de engenharia de software, Paschoal et al. [27] apresentou o protótipo de um agente conversacional que visa apoiar alunos a desenvolver habilidades associadas a extração de requisitos de software, por meio da técnica de entrevista. No estudo, o agente conversacional assume o papel de um *stakeholder* e o aluno interage com ele para tentar extrair os requisitos e melhorar suas habilidades. Ainda, o agente conversacional possui bases de conhecimento sobre conceitos associados ao domínio que podem ser úteis para solucionar as dúvidas dos estudantes quando eles estão envolvidos em atividades educacionais.

Em Mikic-Fonte et al. [18], é abordado o desenvolvimento de um agente conversacional que fica encarregado de responder as dúvidas dos alunos da disciplina de arquiteturas de computadores. A intenção dos autores é permitir que os alunos desenvolvam o hábito de localizar informações de forma autônoma, sem a interação direta do professor. Além disso, segundo os autores, o agente conversacional libera o professor de responder às perguntas que geralmente são feitas todo semestre/ano. No contexto dessa disciplina, a maioria das dúvidas dos estudantes tendem a ser sobre o uso de emuladores de linguagens de montagem (e.g., “como posso instalar o emulador no meu sistema Linux?”). Com a finalização do desenvolvimento, os autores descrevem que esperam usar o agente em um curso de uma plataforma MOOC que é mantida pela universidade.

Outro estudo que menciona sobre a possibilidade de uso de um agente conversacional em curso MOOC de computação é o de Aguirre et al. [19]. O trabalho em questão reporta o planejamento de um agente conversacional para solucionar problemas que os alunos enfrentam ao aprender a linguagem Java e o paradigma de orientação a objetos. O agente conversacional é descrito como um recurso adicional ao MOOC, que faz uso da linguagem natural falada. Os autores do estudo descrevem que a função principal do agente é sugerir módulos do MOOC que o aluno precisa prestar mais atenção, assim como, prover explicações sobre conceitos dos conteúdos

abordados no curso.

IV. ESTUDO EXPERIMENTAL

O estudo experimental abordado neste artigo foi planejado e executado considerando o processo experimental recomendado por Wohlin et al. [28]. Desse modo, ele seguiu as tarefas de identificação e definição do escopo, planejamento, operação e execução. Cada uma dessas tarefas será reportada ao longo desta seção.

A. Identificação e definição do escopo

O experimento foi proposto com a intenção de avaliar se o agente conversacional provê um apoio educacional eficaz ao atender alunos que precisam de ajuda. Assim, o efeito que se busca observar está associado a disponibilidade e auxílio do agente conversacional no contexto de atividades educacionais sobre teste de software. Para tanto, analisou-se a eficácia dos alunos ao realizar uma atividade sobre conceitos de teste de software com a ajuda do agente conversacional. Essa análise é baseada na seguinte questão de pesquisa: o agente conversacional TOB-STT consegue apoiar alunos de teste de software a solucionar as suas dúvidas sobre o conteúdo quando estão fazendo suas atividades?. A partir da questão que norteia o estudo experimental, hipótese para serem investigadas foram geradas:

- **Hipótese nula:** Não existe diferença entre a eficácia dos alunos ao realizarem uma atividade educacional apoiada pelo TOB-STT com a eficácia dos alunos ao realizarem uma atividade educacional sem apoio do TOB-STT.
- **Hipótese alternativa:** Existe diferença entre a eficácia dos alunos ao realizarem uma atividade educacional apoiada pelo TOB-STT com a eficácia dos alunos ao realizarem uma atividade educacional sem apoio do TOB-STT.

A partir da definição da questão de pesquisa, as metas do experimento foram especificadas em cinco partes, como um paradigma semelhante ao GQM (Goal-Question-Metric) [29]: analisar o agente conversacional TOB-STT, com o propósito de verificar a sua efetividade quando usado para apoiar alunos, com respeito a eficácia do aluno ao realizar atividades educacionais, do ponto de vista dos pesquisadores, no contexto de alunos de graduação em Computação que estavam estudando teste de software.

O estudo experimental foi conduzido em duas instituições de ensino, são elas: Universidade Federal de Grande Dourados (UFGD) e Universidade Federal Fluminense (UFF). Ele foi realizado no contexto da disciplina de teste de software, de ambas universidades. No período que o experimento foi conduzido, na UFGD a disciplina dispunha de 21 alunos matriculados e na UFF ela possuía 17 alunos.

B. Planejamento

Após a definição inicial do experimento, estabeleceu-se as variáveis consideradas no estudo e os instrumentos necessários para que o mesmo fosse executado. Esta seção apresenta uma discussão sobre esse planejamento.

Seleção das variáveis

No método científico de experimentação, variáveis são usadas como um mecanismo para mensurar a relação entre causa e efeito. Conforme Wholin et al. [28], essas variáveis são denominadas de variáveis independentes e variáveis dependentes. As variáveis independentes caracterizam-se por apresentar a causa que afeta o resultado de um processo experimental. Por outro lado, as variáveis dependentes retratam o efeito produzido pelas variáveis independentes. A partir desse entendimento, neste estudo experimental foram consideradas as seguintes variáveis independentes:

Conhecimento dos participantes: refere-se ao conhecimento preliminar dos estudantes sobre o conteúdo de teste de software. No contexto deste experimento, os alunos de graduação já haviam participado de aulas sobre conceitos e terminologias de teste de software antes da realização do experimento. As aulas haviam sido ministradas há aproximadamente oito semanas antes da realização do experimento.

Atividade educacional: consiste na atividade sobre teste de software que é realizada pelos alunos.

Mecanismo de apoio ao ensino (fator): representa a iniciativa que guia a configuração dos tratamentos considerados no experimento. Neste experimento são considerados dois tratamentos, são eles:

- **Tratamento A:** os alunos têm a sua disposição o agente conversacional para solucionar suas dúvidas enquanto realizam a atividade educacional.
- **Tratamento B:** os alunos não têm a sua disposição o agente conversacional, tendo que realizar a atividade educacional sem suporte para resolução de dúvidas.

Em relação às variáveis dependentes, considerou-se:

Eficácia: retrata a eficácia dos estudantes em reconhecer os defeitos injetados em um atividade sobre conceitos, terminologias e suas definições no domínio de teste de software. A fórmula usada para mensurar a eficácia é:

$$E_{(i)} = \frac{n_{(i)}}{TOTAL}$$

em que:

- E = representa o valor atribuído para a eficácia;
 i = representa o aluno, isto é, $\{aluno_1, aluno_2, \dots, aluno_z\}$;
 n = representa o número de defeitos que foram identificados corretamente, isto é, que estavam previstos no oráculo e foram reconhecidos pelos alunos;
 total = representa o número de defeitos previstos pelo oráculo de teste.

Descrição da instrumentação

A condução deste experimento requisitou a elaboração e recuperação de materiais. Esses materiais assumem diferentes funcionalidades no contexto do experimento, como ajudar o professor a instruir os estudantes que participam do experimento e coletar dados que posteriormente foram utilizados na análise. O primeiro instrumento definido foi o conjunto de slides usados para ensinar aos alunos os fundamentos de teste e o teste funcional. Esse conjunto de slides e as argumentações que foram construídas para as preleções sobre

o conteúdo tiveram como referência o livro “Introdução ao teste de software” de Delamaro et al. [30].

Com base no livro texto, foram definidos slides que buscavam ajudar o professor a transmitir os fundamentos de teste que envolveu terminologias de teste (engano, defeito, erro, falha, caso de teste, oráculo, conjunto de casos de teste), níveis de teste (teste de unidade, teste de integração, teste de sistema), as principais técnicas de teste (teste funcional, teste estrutural, teste baseado em defeitos) e quais são os critérios de teste dessas técnicas. Ainda, os slides também visavam difundir aspectos teóricos e práticos sobre o teste funcional e seus principais critérios de teste (*i.e.*, particionamento em classes de equivalência e análise do valor limite).

Vale salientar que esses slides foram produzidos por professoras doutoras na área de teste de software, pela instituição de origem do pesquisador principal deste trabalho. Portanto, possuíam o mesmo *background* e referências para instruir os seus alunos sobre tais assuntos. Os slides foram usados para preparar alunos da UFGD e UFF. Nesse caso, aluno do UFGD constituíram o grupo experimental, submetidos ao tratamento A. Já os estudantes da UFF integraram o grupo controle, sujeitos ao tratamento B.

A atividade educacional foi estipulada após a descrição do material para instrução. Essa atividade consiste na análise de um argumento feito pela equipe teste sobre a importância dessa atividade continuar sendo conduzida no âmbito de uma organização de software e, em sua essência, ser priorizada no processo de desenvolvimento. O argumento é um texto direcionado a equipe que gerencia a organização, dado que a mesma solicitou para a equipe de teste um documento que explicasse o funcionamento do teste e o porquê dele ser priorizado. Assim, na atividade, os alunos assumem o papel de testadores e precisam fazer uma análise no documento, listando os defeitos reconhecidos em uma tabela. A análise visa identificar ‘defeitos’ que foram inseridos no documento. O argumento envolveu conceitos, definições de assuntos de teste e exemplos de testes sendo feitos em código-fonte.

O agente conversacional TOB-STT é um instrumento que faz parte do estudo. No planejamento, discutiu-se sobre duas possibilidades: (i) utilizar a versão disponível para acesso web organizada por Paschoal et al. [21]; (ii) instalar a versão em um servidor local da universidade em que o estudo foi realizado para disponibilizá-la aos alunos. Optou-se pela primeira alternativa, porque o estudo não envolve análise das interações feitas pelo agente. Assim, não foi necessário preparar um servidor para disponibilizar o agente. Ao mesmo tempo, acredita-se que isso seja mais conveniente para os trabalhos feitos no contexto do TOB-STT e do grupo que está desenvolvendo o mesmo, por não gerar versões executáveis duplicadas.

Para apoiar a condução da atividade e apresentar o agente conversacional aos alunos, foi utilizado um tutorial definido por Paschoal et al. [21]. O tutorial visa esclarecer as funcionalidades do TOB-STT para os usuários. Nele, há exemplos de interações e link para acesso ao agente. Adicionalmente, visando explanar aos estudantes o experimento, foi construído um Termo de Consentimento Livre e Esclarecido (TCLE), que busca comunicar ao participante do estudo sobre o uso dos

dados fornecidos e sobre a possibilidade de desistência do estudo experimental a qualquer momento da execução, sem prejuízos próprios.

Por fim, um formulário que visa coletar a percepção do usuário sobre o agente conversacional foi recuperado. O instrumento é baseado no estudo de Herpich et al. [25], organizado na pesquisa de Paschoal et al. [21]. Ele é constituído por nove assertivas, baseadas na escala Likert de cinco pontos, que variam de discordo totalmente a concordo totalmente. Essas assertivas se alternam entre assertivas negativas e assertivas positivas, de modo a controlar a tendência hipotética dos participantes concordarem com as afirmações [31]. As assertivas são listadas a seguir:

- 1) Ao interagir com o TOB-STT pela primeira vez, a experiência não foi animadora.
- 2) As respostas fornecidas pelo TOB-STT foram sobre o tópico questionado.
- 3) O TOB-STT não soube responder alguma pergunta realizada.
- 4) Ao utilizar o TOB-STT, eu consegui obter o conhecimento pretendido.
- 5) O TOB-STT não forneceu informações confiáveis em suas respostas.
- 6) O TOB-STT contribuiu para a realização da tarefa.
- 7) O TOB-STT demorou para fornecer as respostas.
- 8) O TOB-STT possui uma interface fácil de usar.
- 9) Eu fiquei insatisfeito com o TOB-STT.

C. Operação e execução

Após a definição dos instrumentos, foi feita a preparação da execução do experimento e a execução.

Preparação

A partir da definição do experimento e preparação do material, em conjunto com docentes da UFGD e UFF, o contexto da execução do experimento foi especializado. No semestre em que ocorreu o planejamento, as professoras estavam ministrando a disciplina de teste de software em suas respectivas universidades. As ementas das disciplinas eram similares⁵ e ambas eram ofertadas para alunos que estavam no último ano de graduação, possuindo a configuração de disciplina optativa. Portanto, o contexto foi selecionado pela adequação e as amostragens usadas no estudo experimental são selecionadas por conveniência. A partir disso, a quantidade de aulas necessárias para realização do estudo foi delimitada, considerando o cronograma de cada disciplina, de modo que o experimento não prejudicasse o andamento da mesma.

Como os conteúdos fazem parte da disciplina, nenhuma tática especial foi necessária. Aulas sobre fundamentos de teste e sobre o teste funcional seriam lecionadas normalmente, considerando o material desenvolvido. Uma aula seria destinada ao tópico de fundamentos e a outra ao tópico de teste funcional. Assim, restaria uma terceira aula para realização da atividade, a ser desenvolvida após oito semanas. Nesse sentido, a atividade aconteceria após os alunos terem retido

o conhecido sobre o assunto. A atividade foi planejada para ser realizada em um horário que habitualmente a aula ocorria, mas a atividade não seria obrigatória para os estudantes, e seria conduzida de modo que os resultados obtidos por eles no contexto dela não interfeririam em suas notas. A atividade seria usada para estimular a participação dos alunos em sala de aula e como uma oportunidade para os mesmos revisarem o conteúdo e observarem aspectos que ainda precisavam ser fixados.

Para a condução, as nove assertivas destinadas a coleta de feedback dos estudantes foram organizadas em um formulário no Google⁶. Assim, o formulário seria disponibilizado aos alunos que participariam do grupo experimental após a realização da atividade. A atividade educacional, por sua vez, foi organizada de modo que os alunos tivessem acesso a ela por meio de computadores nos laboratórios de informática de cada uma das instituições de ensino em que o experimento foi executado. Isso foi feito com a intenção de facilitar a execução, especialmente porque os alunos do grupo experimental teriam que realizar a atividade ao mesmo tempo que interagiriam com o TOB-STT.

Por fim, o ambiente virtual de aprendizagem Moodle foi preparado para facilitar o recebimento dos envios que os alunos teriam que fazer (*i.e.*, a lista de defeitos encontrados no argumento sobre a importância de priorizar o teste de software). Para o grupo experimental, o ambiente foi de essencial importância dado que os alunos teriam que fornecer feedback sobre as suas percepções após o uso do agente conversacional.

Execução

O estudo experimental foi executado em três aulas, em cada uma das instituições de ensino. A primeira e a segunda aula constituem o treinamento necessário para a realização da atividade. Na terceira aula, a atividade educacional foi realizada. Essa seção busca descrever as atividades do experimento realizadas no decorrer dessas aulas. A descrição será feita conforme as execuções.

- *Aula 1:* No primeiro encontro, a professora de cada disciplina apresentou os fundamentos de teste de software. Nesse sentido, foram explicados os objetivos do teste, o processo de teste, terminologias, casos de teste (o que são casos de teste, parte que constituem um caso de teste, ordem de execução: casos de teste em cascata, casos de teste independentes), técnicas de teste (o que são, para que servem, testes baseados em especificação, testes baseados em implementação), critérios de teste, etapas de teste (planejamento, projeto, execução, análise), fases de teste (unidade, integração, sistema). Ao passo que cada assunto era abordado, exemplos eram apresentados e dúvidas sanadas. Ao final, um exercício de fixação do conteúdo foi executado e as dúvidas que surgiram nos estudantes, a partir do exercício, foram resolvidas.
- *Aula 2:* No segundo encontro, a professora de cada disciplina explicou a técnica de teste funcional. Foi

⁵Pensei em colocar o link com as ementas, o q vcs acham?

⁶Mais informações disponíveis em: <<http://bit.ly/3nEc212>>.

mencionado que essa técnica também é conhecida como caixa-preta, que ela se baseia em especificações e documentações do software para se derivar requisitos de teste. Foram discutidos os passos para se aplicar um critério de teste funcional, apresentados os critérios de teste funcionais mais conhecidos, dando atenção especial ao particionamento de classes de equivalência e análise do valor limite. À medida que o passo-a-passo de cada critério foi ensinado, exemplos eram demonstrados. Ao final, de modo similar ao primeiro encontro, os alunos tiveram que realizar uma atividade para exercitar os critérios aprendidos. Conforme dúvidas foram surgindo, as professoras buscavam resolvê-las.

- *Aula 3 (grupo experimental)*: No terceiro encontro, a professora da disciplina de teste de software da UFGD realizou a aula em um laboratório de informática. O laboratório possuía um computador para cada aluno matriculado na disciplina, com acesso à Internet. A professora disponibilizou no Moodle a atividade educacional, o *link* para acesso ao agente conversacional e o *link* para acesso ao formulário que contém as assertivas sobre a percepção de uso. Ela apresentou a atividade e disponibilizou o TCLE. Nesse momento, os alunos foram informados que a atividade seria individual, que não seria considerada avaliativa, apenas para fixação de conteúdo. Os alunos foram informados que deveriam fazer a leitura da atividade e interagir com o agente conversacional, considerando o TOB-STT como um tutor – especialista do domínio. Os alunos também foram notificados que não deveriam usar outro material para consulta, de modo a usar somente o TOB-STT. A partir disso, os alunos teriam 60 minutos para realizar a atividade. Ao final, eles teriam que enviar a tabela com os defeitos identificados e oferecer feedback por meio do formulário disponível.
- *Aula 3 (grupo controle)*: No terceiro encontro, a professora da disciplina de teste de software da UFF realizou a atividade em laboratório, de modo equivalente ao que foi realizado pela professora da UFGD. Assim, cada aluno tinha um computador, com acesso ao ambiente no qual havia sido disponibilizado a atividade educacional e uma opção para envio da tabela com os defeitos identificados. Os alunos foram avisados pela professora que a atividade era individual e eles não poderiam utilizar qualquer mecanismo de apoio. Portanto, consulta a materiais didáticos eram vetados. Os alunos teriam um tempo equivalente a 60 minutos para realizar a atividade e enviar para a professora.

Após a execução do experimento, os dados coletados foram analisados. A análise é apresentada na próxima seção.

V. RESULTADOS

Essa seção busca descrever as análises realizadas nos dados obtidos. Inicialmente, será feita uma descrição sobre as decisões definidas e tomadas durante as análises. Na sequência é apresentada a análise sobre a eficácia do apoio do agente conversacional. Por fim, é apresentada uma discussão sobre as percepções dos alunos que interagiram com o TOB-STT.

A. Sobre as análises

Os defeitos identificados e listados pelos alunos do grupo controle e do grupo experimental foram analisados, considerando um oráculo preliminarmente definido. Com base na análise, considerou-se a participação de 21 estudantes para o grupo experimental e 17 para o grupo controle. A partir disso, a eficácia foi calculada para cada entrega feita por participante do estudo. Assim, diferentes análises nos dados coletados foram feitas. Em particular, utilizou-se análise descritiva e inferência estatística. No contexto da estatística descrita, o estudo se concentra em observar as medidas de posição (*i.e.*, mínimo, máximo, média e mediana) e a medida de dispersão (*i.e.*, o desvio-padrão amostral).

Ainda, na análise descritiva, utilizou-se *box-plots* para reconhecer *outliers*⁷ em cada variável do estudo. Vale salientar que como a amostra é relativamente pequena (*i.e.*, inferior a 30 por grupo amostral), em todas as análises os *outliers* foram considerados.

Por fim, no contexto da estatística inferencial, foi utilizado o teste Shapiro-Wilk para compreender a normalidade dos dados. Como o experimento considera um fator e dois tratamentos, no planejamento da análise do experimento considerou-se o uso do teste t de student para dados que seguem uma distribuição normal e o teste de Mann-Whitney para dados que não seguem uma distribuição normal.

B. Eficácia

Em relação à eficácia, constatou-se que tanto o grupo experimental quanto o grupo controle apresentaram um desempenho não esperado pelos pesquisadores. Isso ocorre porque eram esperados valores de eficácia superiores a 50%, o que indicaria que os alunos identificariam ao menos 50% dos defeitos injetados na atividade educacional. No entanto, conforme é possível observar na Tabela II, a média de defeitos reconhecidos foi de 27%. Esses valores são equivalentes em ambos os grupos.

Ao direcionar a atenção para a mediana, constatou-se que os grupos conseguiram obter uma mediana igual a 30%. Isso significa que 50% dos valores de eficácia são inferiores a 30%. O desvio-padrão amostral revela que a variação do número de defeitos identificados do grupo controle foi equivalente ao grupo experimental. O valor mínimo, que indica o menor valor de eficácia de cada grupo, revela que cada grupo teve alunos que conseguiram identificar corretamente somente um defeito inserido no argumento. Finalmente, ao considerar o valor máximo, constata-se que o grupo experimental teve ao menos um indivíduo participante do estudo que obteve eficácia superior ao grupo controle.

Os *box-plots* construído e representados pela Figura 2 apoiam a análise gráfica da eficácia dos estudantes ao identificar os defeitos, em cada grupo. Nota-se, por meio da Figura 2, que os alunos que não utilizaram o agente conversacional para apoiar a identificação de defeitos obtiveram uma maior variabilidade que os alunos que tiveram apoio do TOB-STT. Isso pode indicar que o agente conversacional contribuiu de alguma forma para que a variabilidade da eficácia na

⁷*Outliers* representam valores atípicos que são causados por variações nos grupos amostrais.

Tabela II: Estatística descritiva dos valores de eficácia

	Grupo	
	Controle	Experimental
Média	27%	27%
Mediana	30%	30%
Desvio-padrão	0,13	0,13
Mínimo	10%	10%
Máximo	50%	60%

identificação de defeitos tenha sofrido uma menor oscilação em relação ao grupo que não teve apoio. Ainda, é possível reconhecer que o grupo experimental teve alunos que foram *outliers* na identificação de defeitos, obtendo 60% de eficácia e 50% de eficácia.

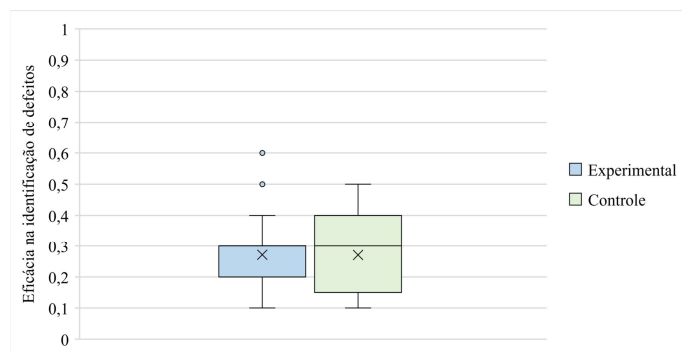


Figura 2: Comparação dos valores de eficácia na identificação de defeitos

Finalmente, para conferir se a eficácia dos alunos ao realizarem a atividade educacional sobre teste de software com apoio do TOB-STT é diferente da eficácia dos alunos que realizam a mesma atividade educacional sem apoio do agente, testes de inferência foram executados. Primeiramente, a normalidade dos dados foi analisada, considerando o nível de confiança de 95%. Com base no teste Shapiro-Wilk, observou-se que: (i) os dados de eficácia do grupo experimental não seguem uma distribuição normal, dado que *p-value* resultou em 0,0000028; (ii) os valores de eficácia do grupo controle também não seguem uma distribuição normal, uma vez que *p-value* foi igual a 0,0000093. Diante disso, o teste de Mann-Whitney foi realizado.

A partir do teste de Mann-Whitney, obteve-se o *p-value* de 0.484352. Nesse sentido, não é possível rejeitar a H_0 . Portanto, o estudo indica que os alunos que não tiveram o apoio do agente conversacional conseguiram obter a mesma eficácia na realização da atividade educacional que os alunos que tiveram apoio do agente. Isso assinala que o suporte do agente conversacional não foi eficaz como se esperava. Vale salientar que a eficácia do suporte educacional oferecido pode ter relação com diferentes aspectos. Acredita-se que a percepção dos alunos sobre o agente pode oferecer indícios que ajudam a compreender por que o agente conversacional não conseguiu potencializar a identificação dos defeitos por parte dos alunos.

C. Feedback dos participantes

Uma análise exploratória foi feita nos dados coletados sobre as percepções dos alunos que utilizaram o TOB-STT para apoiar na atividade educacional. A análise é ilustrada pela Figura 3. Nesse conjunto de dados, nota-se que a assertiva que os alunos mais concordam é a assertiva sobre a interface ser fácil de usar. A assertiva que os alunos relevaram uma maior discordância tem relação com o tempo que o TOB-STT leva para fornecer suas respostas. Nesse caso, como a assertiva era negativa, entende-se que os alunos ficaram satisfeitos com o desempenho do agente ao oferecer respostas.

A partir da análise geral, visando apoiar a descoberta de um melhor entendimento sobre o resultado obtido na análise de eficácia, as assertivas foram organizadas em categorias. Foram definidas sete categorias – a saber: adequação da resposta, autoeficácia, desempenho, eficácia, experiência, qualidade da interface, satisfação. A categoria que agrupou mais de uma assertiva é a adequação de resposta. Ela aborda aspectos sobre a percepção do aluno a respeito da qualidade da interação promovida pelo TOB-STT. Com base na Figura 3, é possível perceber que a maioria dos estudantes manifestou que: (i) as respostas fornecidas pelo TOB-STT estavam de acordo com o assunto da pergunta; (ii) as informações disponibilizadas pelo TOB-STT são confiáveis; (iii) o TOB-STT não conseguiu responder algumas perguntas.

Como houve uma quantidade expressiva de indicações sobre o TOB-STT ter apresentado dificuldades em responder algumas perguntas feitas pelos estudantes, acredita-se que esse problema pode ter uma certa influência no suporte educacional. Nesse sentido, os alunos podem ter ficado com dúvidas específicas sobre algum conceito de teste de software e o agente não conseguiu ajudá-los. Outra interpretação possível é que os alunos tenham feito perguntas que não são relativas ao domínio e, portanto, o agente não soube responder. Entretanto, como os resultados de eficácia foram aquém do esperado, a primeira suspeita não pode ser descartada.

Uma assertiva do formulário diz respeito aos alunos presumirem que possuem as capacidades necessárias para atingir um determinado objetivo (*i.e.*, autoeficácia). Nesse contexto, ao serem interrogados se conseguiram obter o conhecimento que eram esperados a partir do uso do TOB-STT, pôde-se observar que as respostas foram distribuídas entre a escala Likert. A opção que atraiu a atenção maior por parte dos estudantes foi a indecisão. Nesse sentido, a maioria dos estudantes não foi capaz de assumir um lado da escala.

Em um determinada assertiva, os alunos tiveram que informar se acreditavam que o TOB-STT conseguiu contribuir para a realização da atividade. Conforme é possível observar na Figura 3, a maioria das respostas corrobora com a hipótese levantada no início do estudo, que consiste em que o agente conversacional consegue oferecer apoio na realização da atividade. Com base nesse resultado, entende-se que o apoio adicional promoveu um sentimento de confiança nos alunos (*i.e.*, os alunos tiveram a convicção de que o agente conseguiu ajudá-los). Sabe-se, todavia, que o suporte não se deu como esperado.

O instrumento também ajudou a entender como foi a ex-

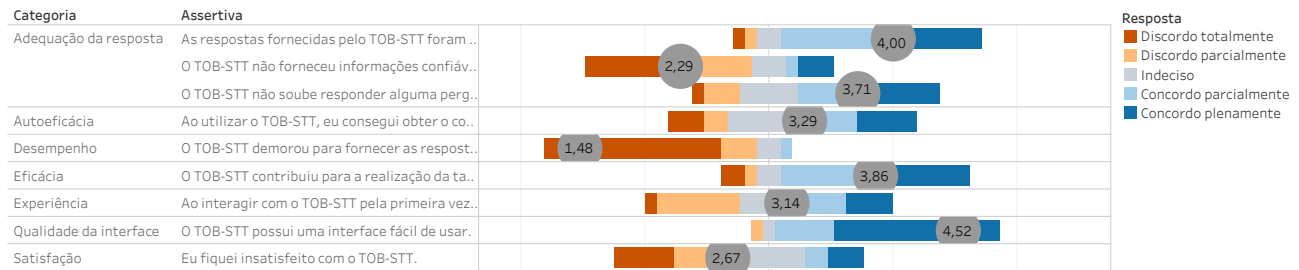


Figura 3: Percepções dos estudantes sobre o TOB-STT

periência dos alunos ao interagir com o agente conversacional durante a atividade. Em particular, os alunos indicaram se a experiência não foi animadora. A Figura 3 revela que não houve uma opção de escala Likert que teve a atenção maior por parte dos alunos. A assertiva que teve mais votos foi a de discordo parcialmente. Como a análise descritiva apresentada ao longo dessa seção tem considerado a tendência de distribuição das indicações feitas pelos alunos, acredita-se que alguns alunos ficaram animados com a experiência de utilizar o agente, enquanto que outros não.

Finalmente, a última assertiva abordou a satisfação dos alunos com o TOB-STT. De modo similar a assertiva relacionada com a experiência, os dados indicam que alguns alunos ficaram satisfeitos e outros não muito satisfeitos.

A partir dos resultados encontrados, podê-se coletar algumas evidências que contribuem para o entendimento a respeito da análise de eficácia do suporte educacional. Os resultados que mais prendem a atenção dizem respeito ao fato de que o agente não conseguiu ajudar alguns alunos, no sentido de oferecer a resposta para algumas dúvidas. Mesmo assim, os alunos acreditam que o agente conseguiu ajudá-los na conclusão da atividade. Como os resultados de eficácia, apresentado na seção V-B, indicam que o suporte oferecido pelo agente conversacional não promoveu uma maior eficácia na identificação de defeitos por alunos de teste de software, a sensação de amparo se manifestou na sessão experimental.

VI. AMEAÇAS À VALIDADE

Tendo em vista que o estudo apresentado no decorrer deste artigo utilizou o processo experimental, é preciso esclarecer que ele não está livre de ameaças à validade. Assim, durante o planejamento e condução, algumas iniciativas foram feitas, visando mitigá-las. As ameaças e ações feita para mitiga-las são apresentadas a seguir:

- **Confiabilidade das medidas:** no estudo experimental, a eficácia foi utilizada como métrica para conferir se o aluno que tem acesso ao agente conversacional TOB-STT consegue identificar os defeitos com mais precisão que o aluno que não tem acesso. Essa métrica tem sido usada em estudos experimentais sobre ensino de teste de software, como Souza [32] e Paschoal et al.[33], ao passo que considera as respostas emitidas pelo aluno em relação ao que era esperado em um oráculo. Embora esse cuidado tenha sido tomado, é possível que conclusões errôneas sejam tomadas. Para mitigar qualquer ameaças

nesse contexto, além do estudo usar métrica adotada em outros trabalhos da temática, o oráculo de teste foi definido por uma pesquisadora e professora da disciplina de teste de software com mais de 20 anos de atuação. A comparação entre os defeitos identificados pelos alunos com o que era esperado no oráculo, por sua vez, foi feita por uma única pessoa que tem atuado em pesquisas sobre ensino de teste de software. Ainda, para conseguir novas evidências sobre a eficácia do apoio oferecido pelo TOB-STT, um formulário específico foi usado. Esse formulário já havia sido utilizado no contexto de outras pesquisas que buscavam coletar as percepções dos alunos sobre agentes conversacionais na educação. Portanto, evitou-se formulário mal formulado, que poderiam prejudicar a viabilidade da pesquisa.

- **Viés de um único método:** a eficácia do apoio educacional do agente conversacional foi analisada a partir de uma única atividade educacional. Idealmente, acredita-se que a eficácia seja observada ao longo da realização de um conjunto maior de atividades. Isso ajudaria a ter um contexto mais próximo do ambiente de aprendizado. Os alunos podem ir melhor em uma atividade que em outra, por causa de suas afinidades, preferências, etc. No entanto, mais aulas das disciplinas seriam necessárias para envolver um conjunto de atividades, o que comprometeria o cronograma das disciplinas. Além disso, em estudos experimentais da temática, a realização de uma única atividade tem sido adotada [33], justamente para não prejudicar o ensino de outros assuntos e o calendário da disciplina.
- **Generalização dos resultados:** espera-se que os resultados obtidos em um estudo que usa o processo experimental sejam generalizáveis. Nesse sentido, a escolha dos participantes do experimento pode inviabilizar a generalização, porque são eles que fornecem os dados usados durante a análise. No contexto desse experimento, estudantes de teste de software foram selecionados. Além disso, os participantes eram de duas instituições de ensino, o que deixa a amostragem heterogênea. Assim, a ameaça foi mitigada a partir da seleção do contexto acadêmico.
- **Treinamento obtido pelo aluno:** o conhecimento preliminar dos estudantes que participaram do experimento pode comprometer os resultados obtidos. Portanto, o modo como os alunos foram ensinados/treinados precisou ser planejado e abordado com cuidado. Os alunos do grupo

controle e do grupo experimental receberam treinamento sobre os mesmos assuntos, com slides baseados em uma referência única, por professores especialistas no assunto, colegas de grupo de pesquisa, com suas dissertações e teses na área, e estudos feitos em colaboração sobre os assuntos que fazem parte do experimento.

- Risco de um erro acontecer na análise: em alguns estudos experimentais, é possível o teste não indicar o relacionamento entre causa e efeito, mas esse relacionamento pode existir. Nesse sentido, é preciso um cuidado específico com os tratamentos e os dados analisados. No contexto desse experimento, essa ameaça ficou em evidência. Para mitigá-la, uma replicação do experimento foi realizada, dessa vez com alunos que estavam estudando a disciplina de Teste e Inspeção de Software no Instituto de Ciências Matemáticas e Computacionais na Universidade de São Paulo. Os alunos em questão, selecionados por conveniência, também estavam no último ano de seus cursos de graduação, receberam os treinamentos e após oito semanas realizar a atividade educacional com apoio do TOB-STT. A partir disso, os dados foram coletados e analisados. Ao final, o teste de Mann-Whitney revelou novamente que não há diferença estatística em termos de eficácia dos alunos que realizaram a atividade com apoio do agente (alunos da USP) e sem apoio do agente conversacional (alunos da UFF), dado que $p\text{-value}$ foi equivalente a 0,959729. Portanto, com a nova análise pode-se mitigar a ameaça assumindo que a hipótese aceita na seção V-B é uma hipótese verdadeira.

VII. LIMITAÇÕES

Embora o estudo tenha sido realizado de modo sistematizado, com planejamento e mitigação de ameaças que poderiam comprometer a sua validade, ele possui algumas retribuições. Em particular, como o grupo experimental foi constituído por alunos de uma mesma universidade, é possível que os alunos tivessem uma eficácia inferior se não tivessem uma ajuda do agente conversacional. Como o estudo não coletou dados adicionais que poderiam ser utilizados para conferir essa suposição, uma questão de pesquisa fica em aberta “os alunos que interagiram com o TOB-STT teriam uma eficácia inferior a eficácia obtida por eles mesmos se não tivessem o apoio do TOB-STT?”. Vale salientar que uma replicação da sessão experimental foi realizada no ICMC-USP e ajuda a dar um entendimento que o agente não conseguiu oferecer aumento na eficácia dos resultados, todavia, o conhecimento preliminar do aluno não foi investigado. Então, novas teorias podem surgir a partir deste estudo.

Uma maneira de abordar essas limitações, poderia ser trabalhada na divisão de grupos. Em vez da divisão de alunos para formação dos grupos (controle e experimental) ser baseada na instituição de origem dos alunos, cada disciplina poderia ter sido dividida em dois grupos, randomicamente, no qual um grupo realizaria a atividade com o agente conversacional e outra sem o agente conversacional. No entanto, essa divisão requisitaria de espaços diferentes (dois laboratórios de informática em cada universidade) e

ao menos um outro professor para apoiar a execução, dado que enquanto um grupo estaria participando da atividade sem apoio do agente conversacional, o outro grupo estaria em outro laboratório participando da atividade com apoio do agente. Uma segunda alternativa que poderia oferecer um entendimento maior sobre a eficácia, seria por meio de uma configuração de experimentação pareada [28]. Nesse sentido, cada participante do experimento iria experienciar ambos os tratamentos. Dado esse tipo de configuração, para o contexto desse experimento, os pesquisadores teriam que lidar com uma ameaça relacionada com o fato de o aluno obter informação no primeiro experimento e usar no segundo tratamento. Por exemplo, conhecer a atividade educacional.

VIII. CONCLUSÕES

O TOB-STT é um agente conversacional que foi estabelecido com a intenção de apoiar momentos em que os alunos precisam de suporte de um especialista e não tem acesso ao professor. Destinado para cenários como MOOCs de teste de software, ele foi descrito inicialmente em Paschoal et al. [21], em um estudo que aborda a avaliação de seu conhecimento sobre teste de software. Como a eficácia do seu apoio não havia sido estudada, este artigo se propôs a compreender melhor o seu impacto quando usado como mecanismo de apoio ao ensino em um cenário no qual o aluno não tem apoio do professor. Assim, um experimento controlado foi planejado e executado, no qual alguns estudantes de teste de software realizaram uma atividade educacional com apoio do mecanismo enquanto que outros alunos realizaram a mesma atividade sem apoio do TOB-STT. A partir das sessões experimentais, concluiu-se que o suporte oferecido pelo agente não teve um impacto significativo na realização da atividade educacional de teste de software. A configuração experimental usada neste estudo pode ser modificada para se ter um entendimento a respeito de um aspecto que foi discutido a partir dos dados analisados: os alunos que interagiram com o agente conversacional poderiam ter tido uma eficácia inferior se não tivessem apoio do TOB-STT. Como trabalho futuro, pretende-se investigar essa hipótese. Ainda, o trabalho promove algumas questões que precisam ser refletidas e estudadas: (i) há atividades educacionais específicas que o agente conversacional TOB-STT consegue oferecer um apoio significativo?; (ii) há atividades educacionais, não conhecidas pela comunidade, que professores de teste de software precisam de suporte e o agente conversacional pode suprir essa necessidade?. Os resultados também indicaram que o agente conversacional não está conseguindo responder algumas questões feitas pelos alunos. Assim, acredita-se que o modelo usado para representar o conhecimento do agente conversacional também precisa ser estudado e melhorado.

AGRADECIMENTOS

Os autores gostariam de agradecer a CAPES - Código de Financiamento 001, a FAPESP (Processos 2018/26636-2 e 2019/06937-0) e ao CNPq (Processos 311494/2017-0 e 312922/2018-3) pelo apoio financeiro.

REFERÊNCIAS

- [1] M. E. Delamaro, J. C. Maldonado, and M. Jino, "Conceitos básicos," in *Introdução ao Teste de Software*, M. E. Delamaro, J. C. Maldonado, and M. Jino, Eds. Rio de Janeiro, Brasil: Elsevier, 2016, ch. 1, pp. 1–8.
- [2] G. Tan, "A collection of well-known software failures," 2016. [Online]. Available: <http://www.cse.psu.edu/~gxt29/bug/softwarebug.html>
- [3] M. Zhivich and R. K. Cunningham, "The real cost of software errors," *IEEE Security Privacy*, vol. 7, no. 2, pp. 87–90, 2009.
- [4] G. Fraser, A. Gambi, and J. M. Rojas, "A preliminary report on gamifying a software testing course with the code defenders testing game," in *European Conference of Software Engineering Education*, 2018, pp. 50–54.
- [5] ACM, "Curricula recommendations," 12 2016, acesso em: 10 de janeiro de 2020.
- [6] IEEE, "Computing curriculum efforts," 12 2013, acesso em: 10 de janeiro de 2020.
- [7] F. B. V. Benitti, "A methodology to define learning objects granularity: a case study in software testing," *Informatics in Education*, vol. 17, no. 1, pp. 1–20, 2018.
- [8] L. N. Paschoal and S. R. S. Souza, "A survey on software testing education in brazil," in *Brazilian Symposium on Software Quality*, 2018, pp. 334–343.
- [9] O. A. L. Lemos, F. F. Silveira, F. C. Ferrari, and A. Garcia, "The impact of software testing education on code reliability: An empirical assessment," *Journal of Systems and Software*, 2017.
- [10] V. Garousi and J. Zhi, "A survey of software testing practices in canada," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1354–1376, 2013.
- [11] S. Ng, T. Murnane, K. Reed, D. Grant, and T. Y. Chen, "A preliminary survey on software testing practices in australia," in *Australian Software Engineering Conference*. IEEE, 2004, pp. 116–125.
- [12] S. M. Melo, V. X. Moreira, L. N. Paschoal, and S. R. S. Souza, "Testing education: A survey on a global scale," in *Brazilian Symposium on Software Engineering*, 2020, pp. 554–563.
- [13] A. C. Dias-Neto, S. Matalonga, M. Solari, G. Robiolo, and G. H. Travassos, "Toward the characterization of software testing practices in south america: looking at brazil and uruguay," *Software Quality Journal*, vol. 25, no. 4, pp. 1145–1183, 2017.
- [14] A. G. O. Fassbinder, M. Fassbinder, E. F. Barbosa, and G. D. Magoulas, "Massive open online courses in software engineering education," in *IEEE Frontiers in Education Conference*, 2017, pp. 1–9.
- [15] J. M. Prates, R. E. Garcia, and J. C. Maldonado, "Moocs on the context of software engineering teaching and training: Trends and challenges," in *IEEE Frontiers in Education Conference*, 2018, pp. 1–9.
- [16] E. P. Enouï, "Teaching software testing to industrial practitioners using distance and web-based learning," in *International Workshop on Frontiers in Software Engineering Education*. Springer, 2019, pp. 73–87.
- [17] A. Mittal, L. Vigentini, M. Djatmiko, G. Prusty, Y. Sharma, and M. E. King, "Mooc-o-bot: using cognitive technologies to extend knowledge support in moocs," in *IEEE International Conference on Teaching, Assessment, and Learning for Engineering*, 2018, pp. 69–76.
- [18] F. A. Mikic-Fonte, M. Llamas-Nistal, and M. Caeiro-Rodríguez, "Using a chatterbot as a faq assistant in a course about computers architecture," in *IEEE Frontiers in Education Conference*, 2018, pp. 1–4.
- [19] C. C. Aguirre, C. D. Kloos, C. Alario-Hoyos, and P. J. Muñoz-Merino, "Supporting a mooc through a conversational agent. design of a first prototype," in *International Symposium on Computers in Education*, 2018, pp. 1–6.
- [20] J. L. Z. Montenegro, C. A. Costa, and R. R. Righi, "Survey of conversational agents in health," *Expert Systems with Applications*, vol. 129, pp. 56–67, 2019.
- [21] L. N. Paschoal, L. F. Turci, T. U. Conte, and S. R. Souza, "Towards a conversational agent to support the software testing education," in *Brazilian Symposium on Software Engineering*, 2019, pp. 57–66.
- [22] R. S. Wallace, "The anatomy of alice," in *Parsing the Turing Test*. Springer, 2009, pp. 181–210.
- [23] K. Katchapakirin and C. Anutariya, "An architectural design of scratchthai: A conversational agent for computational thinking development using scratch," in *International Conference on Advances in Information Technology*, 2018, pp. 1–7.
- [24] J. M. Ocaña, E. K. Morales-Urrutia, D. Pérez-Marín, and S. Tamayo-Moreno, "How to create a pedagogic conversational agent for teaching computer science," in *Advanced Online Education and Training Technologies*. IGI Global, 2019, pp. 114–134.
- [25] F. Herpich, F. B. Nunes, G. B. Voss, and R. D. Medina, "Three-dimensional virtual environment and npc: a perspective about intelligent agents ubiquitous," in *Handbook of Research on 3-D Virtual Environments and Hypermedia for Ubiquitous Learning*. IGI Global, 2016, pp. 510–536.
- [26] M. D. Leonhardt, L. Tarouco, R. M. Vicari, E. R. Santos, and M. S. da Silva, "Using chatbots for network management training through problem-based oriented education," in *IEEE International Conference on Advanced Learning Technologies*, 2007, pp. 845–847.
- [27] L. N. Paschoal, M. M. Oliveira, and P. M. M. Chicon, "A chatterbot sensitive to student's context to help on software engineering education," in *XLIV Latin American Computer Conference*. IEEE, 2018, pp. 839–848.
- [28] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.
- [29] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," in *Encyclopedia of Software Engineering*, 1st ed., J. J. Marciniak, Ed. Hoboken, New Jersey: John Wiley & Sons, 1994, vol. 2, pp. 528–232.
- [30] M. E. Delamaro, J. C. Maldonado, and M. JINO, *Introdução ao teste de software*, 2nd ed. Elsevier, 2016, vol. 1.
- [31] J. R. Lewis, "The system usability scale: past, present, and future," *International Journal of Human-Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018.
- [32] D. M. Souza, "Subsídios à integração de ferramentas de avaliação automática e sistemas de gerenciamento de aprendizagem," Ph.D. dissertation, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2017.
- [33] L. N. Paschoal, M. M. Oliveira, S. M. Melo, E. F. Barbosa, and S. R. Souza, "Evaluating the impact of software testing education through the flipped classroom model in deriving test requirements," in *Brazilian Symposium on Software Engineering*, 2020, pp. 570–579.