

Problem-Based Learning to Align Theory and Practice in Software Testing Teaching

Jean Felipe P. Cheiran
Universidade Federal do Pampa
Av. Tiarajú, 810
Alegrete, RS 97546-550
jeancheiran@unipampa.edu.br

Ewerson Luiz de S. Carvalho
Universidade Federal do Pampa
Av. Tiarajú, 810
Alegrete, RS 97546-550
ewersoncarvalho@unipampa.edu.br

Elder de M. Rodrigues
Universidade Federal do Pampa
Av. Tiarajú, 810
Alegrete, RS 97546-550
elderrodrigues@unipampa.edu.br

João Pablo S. da Silva
Universidade Federal do Pampa
Av. Tiarajú, 810
Alegrete, RS 97546-550
joaosilva@unipampa.edu.br

ABSTRACT

Traditional approaches to teach software testing lack to align theory with practice in higher education. Problem Based Learning (PBL) is an alternative that allows learning through individual practical experiences. The objective of this paper is to report experiences to teach software testing by using PBL in Software Engineering undergraduate course of Federal University of Pampa. The report covers two editions of Problem Solving course that involved 51 students and focused on students creating and running test cases across software components developed by professors. Data collection had been made by questionnaires with open and closed questions, and data was examined by statistics and Content Analysis technique. As results, we notice evidences of knowledge and perceptions maturing by students about software testing and software quality. We also verify it's possible to mix PBL with other teaching methodologies such as gamification and to amplify their benefits.

CCS CONCEPTS

•Software and its engineering → Software testing and debugging; Collaboration in software development;

KEYWORDS

Problem Solving, Problem Based Learning, Software Testing, Gamification

ACM Reference format:

Jean Felipe P. Cheiran, Elder de M. Rodrigues, Ewerson Luiz de S. Carvalho, and João Pablo S. da Silva. 2017. Problem-Based Learning to Align Theory and Practice in Software Testing Teaching. In *Proceedings of SBES'17, Fortaleza, CE, Brazil, September 20–22, 2017*, 10 pages. DOI: 10.1145/3131151.3131181

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SBES'17, Fortaleza, CE, Brazil

© 2017 ACM. 978-1-4503-5326-7/17/09...\$15.00

DOI: 10.1145/3131151.3131181

1 INTRODUÇÃO

Proporcionar processos de ensino e aprendizagem consistentes que alinhem a teoria e a prática de Engenharia de Software, nos quais os alunos vivenciem experiências similares àquelas de sua futura atuação profissional, é um desafio que persiste e que é continuamente retomado [33]. Buscando superar esse desafio, ações de planejamento de currículos e componentes curriculares [31], articulação de novas metodologias de ensino [6, 10] e inclusão de elementos pedagógicos inovadores [19, 30] são conduzidas nas diferentes áreas da computação.

Teste de software é uma das áreas que apresenta desafios para o ensino, dado principalmente a dificuldade e consequente ineficiência em ensinar teste de software por meio de aulas teóricas e expositivas [32]. Teste de software é o processo de execução de um programa, ou parte dele, com a intenção de encontrar erros [21]. Motivado pela profusão de linguagens de programação, ambientes de desenvolvimento e ferramentas de teste disponíveis, é necessário que o docente faça uso de ambientes práticos que contextualizam o problema e ao mesmo tempo motivem os estudantes a se envolverem nas atividades de ensino de teste de software. Além disso, o ensino de teste de software também deve incorporar abordagens que ampliem a autonomia dos estudantes [9] e que aumentem sua motivação nas atividades acadêmicas [8, 14].

A Abordagem Baseada em Problemas (ABP), do inglês *Problem-Based Learning* (PBL), é uma abordagem de aprendizagem ativa que, a partir da resolução de problemas, possibilita aos alunos vivenciar experiências que retratam a realidade do contexto profissional no ambiente acadêmico. Originada nos cursos da área da saúde [2], a ABP vem sendo incorporada com sucesso em cursos e disciplinas da área de computação [3, 6, 9, 11, 23, 24, 26].

A integração de ABP a outras estratégias pedagógicas ainda é pouco explorada, mostrando-se um terreno fértil de experimentação educacional. Considerando, por exemplo, questões relacionadas à cultura digital das novas gerações de estudantes, às diferenças geracionais entre professores e alunos que podem gerar dificuldades, e ao elemento de **diversão** que faz parte do cotidiano dos jovens estudantes e que poderia ser um facilitador nos processos de aprendizagem [19], pode-se ensaiar a introdução de elementos lúdicos que não interferem diretamente na abordagem pedagógica

que está por trás do desenho curricular. Nesse caso, a gamificação se mostra uma abordagem de apoio potencial.

O objetivo desse trabalho é relatar a execução e os resultados obtidos (impressões de estudantes e professores) em duas edições de um componente curricular de Resolução de Problemas (RP) da Universidade Federal do Pampa (Unipampa) que se fundamenta em ABP e com foco em teste de software. O primeiro relato descreve uma execução tradicional e aponta indícios do amadurecimento dos alunos no contexto do componente curricular. O segundo relato apresenta a integração de ABP e elementos de gamificação, compilando percepções dos estudantes, benefícios dessa associação e problemas enfrentados.

Cheiran e Rodrigues [5] apresentam resultados preliminares obtidos no segundo relato supracitado. Neste trabalho, além de incluir e discutir os resultados do primeiro relato, os resultados do segundo relato são discutidos em profundidade.

Esse texto se encontra organizado como segue. A Seção 2 apresenta a contextualização do trabalho, incluindo uma discussão sobre os componentes curriculares de Resolução de Problemas no curso de Engenharia de Software da Unipampa, e discute um conjunto de trabalhos relacionados ao nosso tema. A Seção 3 descreve a metodologia utilizada para avaliação dos experimentos realizados. A Seção 4 apresenta o relato de como a ABP foi aplicada nos componentes curriculares retratados e a integração com gamificação em uma das edições. A Seção 5 se dedica à discussão dos resultados obtidos em cada uma das edições tratadas. Finalizando, a Seção 6 apresenta uma síntese de resultados, reflexões sobre as experiências, limitações desse relato e indicações para trabalhos futuros.

2 CONTEXTUALIZAÇÃO

O Curso de Engenharia de Software da Universidade Federal do Pampa (Unipampa), desde sua abertura em 2010, adota a abordagem de ensino-aprendizagem construtivista Aprendizado Baseado em Problemas (ABP) [31]. O conceito chave de ABP é utilizar problemas reais como motivador para o processo de aprendizagem, criando uma base de conhecimento integrada, interdisciplinar, interativa e orientada a prática [1]. Em ABP, a aprendizagem é uma iniciativa dos estudantes, feita sob orientação de um tutor. Para isso, os currículos que adotam a abordagem têm um número reduzido de conteúdo expositivo e um tempo maior para autoestudo [29]. Além disso, os estudantes são organizados em pequenos grupos para trabalhar de forma colaborativa.

A Figura 1 apresenta os principais elementos envolvidos na ABP. O problema a ser resolvido é o gatilho para o processo de aprendizagem. O conhecimento representa o que é necessário saber para encontrar uma solução adequada para o problema. O aluno é autônomo e responsável pela construção dos conhecimentos necessários para resolver o problema. O tutor atua como facilitador no processo de aprendizagem [31].

O Curso de Engenharia de Software da Unipampa adota ABP no currículo através de um conjunto de seis componentes curriculares denominados Resolução de Problemas (RPs). Esses componentes integram, de modo interdisciplinar e transversal, diferentes conteúdos na abordagem de uma situação-problema que se aproxima da realidade profissional que os egressos irão encontrar. Conforme apresentado na Figura 2, as RPs cobrem os seguintes eixos temáticos:

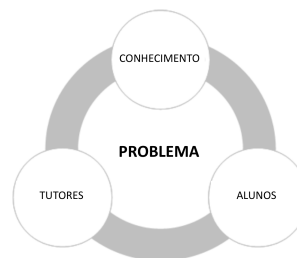


Figura 1: Elementos envolvidos em ABP [31].

Construção de Software (1º e 2º semestres), Modelagem e Projeto de Software (3º semestre), Análise e Validação de Software (4º semestre) e Processo, Evolução, Qualidade e Gerenciamento de Software (5º e 6º semestre). Os dois últimos semestres são reservados para o trabalho de conclusão de curso e estágio obrigatório [31].

Semestre 1	Semestre 2	Semestre 3	Semestre 4
Construção		Modelagem e Projeto	Análise e Verificação
Semestre 5	Semestre 6	Semestre 7	Semestre 8
Processo, Evolução, Qualidade e Gerenciamento		Estágio Obrigatório e Trabalho de Conclusão de Curso	

Figura 2: Eixos de conhecimento de cada semestre [31].

2.1 Resolução de Problemas IV

O componente curricular Resolução de Problemas IV (RP IV) é um dos seis componentes do Curso de Engenharia de Software que adota ABP. Ele busca prover aos alunos o aperfeiçoamento em Análise de Software, explorando métodos e técnicas de engenharia de requisitos e modelagem de software; e em Verificação e Validação (V&V) de software. Ao término de cada edição de RP IV, espera-se que os estudantes [31]:

- Conheçam métodos e técnicas de levantamento de requisitos;
- Saibam descrever requisitos de software dentro de um contexto;
- Realizem V&V nos produtos de trabalho relacionados.

Em RP IV, os estudantes são organizados em grupos de trabalho. Cada grupo é responsável pelo desenvolvimento ou manutenção de um sistema de software. Cada membro do grupo realiza atividades de engenharia de requisitos e de V&V necessárias para atender ao escopo do projeto. Além do software executável, os grupos entregam os artefatos de engenharia relacionados, por exemplo, especificações, planos e relatórios de inspeção, códigos-fonte, planos de teste, casos de teste, *scripts* de teste e relatórios de teste.

RP IV é organizado em três momentos distintos: abertura, execução, e fechamento. Na **abertura**, geralmente a primeira aula, os professores apresentam o problema a ser resolvido e estabelecem o escopo geral de trabalho, enquanto os alunos se organizam em grupos. Na **execução**, nas demais aulas do semestre, os estudantes

trabalham para resolver o problema. A execução, por sua vez, é organizada em marcos¹. No início de cada marco são definidos pelos professores o escopo de trabalho e os artefatos a serem entregues naquele marco. No **fechamento**, normalmente a última aula, os professores e os alunos fazem uma reflexão sobre as lições aprendidas durante o semestre letivo.

Os estudantes são responsáveis por planejar, monitorar e realizar o trabalho requerido e desenvolver as competências e habilidades necessárias para resolver o problema. Os professores atuam como facilitadores do processo de ensino-aprendizagem, atuando junto aos grupos sempre que houver demanda. Quando necessário, os professores fazem intervenções junto à turma para discutir algum tema que contribua para os objetivos de aprendizagem.

A edição de 2016 de RP IV introduziu práticas de gamificação (*i.e. gamification*) para incentivar o engajamento dos discentes para resolução de problemas. Gamificação, muitas vezes chamada ludificação, pode ser definida como “uso de elementos de *design* de jogos em contextos fora de jogos” [7], incluindo geralmente metáforas comuns nos enredos dos jogos. Conforme a proposta de Martins e Giraffa [19], foram utilizados elementos de gamificação tais como enredo, desafios, cooperação e pontuação, sendo adicionado também um elemento de competição não previsto pelo modelo. A descrição detalhada é apresentada na seção 4.2.

O processo de avaliação em RP IV leva em consideração avaliações individuais e em grupo. A **avaliação individual** analisa o trabalho realizado por cada estudante para resolver o problema. Para tanto, cada estudante deve periodicamente reportar aos professores o trabalho realizado. Os professores avaliam se o trabalho é condizente com o esperado e, ainda, se houve progresso na direção da resolução do problema. Com base nesta avaliação, os professores atribuem um conceito ao aluno. A **avaliação em grupo** considera a capacidade de entrega do grupo de trabalho como um todo. Para isso, os grupos fazem a entrega e a apresentação dos produtos de trabalho gerados. A avaliação leva em consideração o atendimento ao escopo, a qualidade dos produtos de trabalho entregues e a organização da apresentação em si. A partir disso, os professores atribuem aos estudantes do grupo uma nota.

2.2 Trabalhos Relacionados

A aplicação da metodologia ABP no processo de ensino e aprendizado em Engenharia de Software já foi relatada em outros trabalhos [9, 11, 23, 24, 26], sendo que alguns têm explorado a aplicação de ABP no ensino de Qualidade e Teste de Software [9, 11, 24].

Santos et al. [26] apresentam a aplicação da metodologia ABP no Mestrado Profissional em Engenharia de Software (MPES) do C.E.S.A.R [4], com o objetivo de aprimorar as habilidades dos estudantes em resolver problemas reais no contexto de fábricas de software. De acordo com os autores, o uso da ABP tem apresentado resultados positivos desde a primeira turma na qual foi aplicada, tais como um crescimento no desempenho das fábricas e uma redução da diferença entre os rendimentos dos estudantes. Como os autores não encontraram métricas da aplicação de ABP, em um contexto similar, não foi viável a comparação com outras propostas. Por

este motivo, os resultados apresentados consideram as diferentes turmas de alunos onde a metodologia foi aplicada.

Richardson e Delaney [23] apresentam um estudo de caso da aplicação da ABP no ensino de técnicas de diagramação em uma disciplina da engenharia de software. No contexto do estudo de caso, um dos autores atuou como um facilitador/guia, enquanto o outro atuou apenas como observador. Este se concentrou somente na análise dos fatores de aprendizado e em como os problemas eram implementados em sala de aula. Além disso, o observador buscava identificar quais são os pontos fracos e fortes da aplicação de ABP no contexto do estudo de caso. Como resultado é levantada uma série de questões, como por exemplo o tamanho dos grupos de alunos, que devem ser pequenos, e o papel do professor tutor na condução da disciplina.

Por sua vez, Figuerêdo et al. [9] apresentam uma abordagem, usando a metodologia ABP, para o treinamento real de engenheiros de teste de software. Com o objetivo de avaliar a abordagem proposta foi conduzido um estudo empírico com dez estudantes de graduação, divididos em dois grupos de cinco alunos. Os dois grupos de estudantes tinham por objetivo testar uma ferramenta CASE de suporte ao teste funcional, utilizando teste exploratório, com o objetivo de encontrar *bugs*. Os autores destacam que o teste exploratório estimula as características de criatividade e investigação, desafiando o testador a desenvolver e projetar casos de testes melhores e mais eficazes. Neste trabalho foram avaliados os conhecimentos dos participantes antes e depois da execução do estudo, notas e a quantidade de *bugs* reportados. Os resultados indicam que a abordagem possibilita o engajamento dos estudantes em situações reais, possibilitando a aquisição de experiência em cenários reais de teste de software.

Richardson et al. [24] relatam como aplicaram ABP com o objetivo de ensinar aquelas habilidades práticas e teóricas, sobre qualidade de software, que são requeridas dos estudantes quando entram no mercado de trabalho. Apesar dos resultados apresentados demonstrarem claramente que a adoção de metodologia tem resultados positivos tanto para os alunos quando para os professores, a ampla adoção da ABP no ensino de engenharia de software é dependente do desenvolvimento de problemas e conceitos, bem como da utilidade dos resultados dentro da indústria.

Embora diversos trabalhos apresentem a aplicação e a avaliação de ABP em Engenharia de Software, a falta de um *framework* que permita replicar o mesmo processo de avaliação torna os resultados dependentes do contexto e dos instrumentos utilizados. A falta de um processo bem definido inviabiliza ainda a replicação e a comparação de estudos entre diferentes instituições, não permitindo assim extrapolar os resultados em diferentes contextos.

3 MATERIAIS E MÉTODOS

Este trabalho possui caráter exploratório e qualitativo. Como estudo exploratório, ele determina tendências de fenômenos relativamente desconhecidos [25, 34]. Como a compreensão da eficácia de ABP na Engenharia de Software exige analisar a perspectiva dos participantes, o enfoque qualitativo permite aprofundamento de seus pontos de vista, opiniões e experiências [25]. Este estudo foi estruturado em forma de relato de experiência, visando expressar a ABP

¹Cada marco representa a entrega dos resultados para um conjunto de tarefas específicas. Ele é precedido por um período pré-estabelecido de semanas de trabalho no estilo *time boxing* [22].

como uma alternativa para o ensino de assuntos de engenharia de software [34].

O relato apresentado cobre duas edições do componente curricular RP IV do curso de Engenharia de Software da Unipampa. As atividades de desenvolvimento e acompanhamento foram executadas em laboratórios de informática, embora o trabalho extraclasse em outros ambientes também tenha sido necessário. A edição de 2014 contou com participação de 26 alunos. Os dados capturados nessa edição continham o que os alunos entendiam por teste de software e qual sua importância. Para isso, os questionários realizados foram disponibilizados aos alunos através de atividades criadas no ambiente Moodle. Sendo assim, os dados não foram colhidos de forma anônima. Enquanto isso, a edição de 2016 contou com participação de 25 alunos. Os dados dessa edição continham uma avaliação de satisfação dos alunos com relação ao componente curricular.

Cabe destacar que, devido à rotatividade dos professores que conduzem disciplinas de RP, não foram realizados experimentos dessa natureza em algumas edições e, justamente por não haver medições ou estratégias pedagógicas diferenciadas, a edição de 2015 não foi incluída no artigo.

Para coletar dados, um questionário foi elaborado por meio da ferramenta *Google Forms*, este respondido de forma anônima. Questionários com perguntas fechadas e abertas foram utilizados como instrumento para coleta sistemática de dados [18]. A edição de 2014 conteve uma pergunta aberta e foi aplicada duas vezes no componente curricular: no primeiro e no último dia de aula. A edição de 2016 conteve dez perguntas fechadas de múltipla escolha com uso da escala Likert² [17] e duas perguntas abertas. Esse questionário foi aplicado apenas uma vez no final do respectivo marco do componente curricular. As respostas obtidas em cada item foram analisadas do ponto de vista de sua mediana e sua moda [15]. Assim, pode-se identificar a tendência central da postura dos participantes em relação à afirmação realizada.

As respostas para as perguntas abertas foram descritas e interpretadas pela técnica de análise de conteúdo. O processo aplicado consiste em cinco etapas distintas [20]:

- (1) **Preparação:** inclui a leitura superficial dos dados e a organização do trabalho como identificação dos itens originais de dados por um código único para facilitar rastreabilidade;
- (2) **Unitarização**³: consiste na leitura detalhada dos itens originais de dados, na definição da granularidade dos itens de análise (ou unidades de análise), na extração e no isolamento dessas unidades, e na preservação de seu contexto pela rastreabilidade;
- (3) **Categorização:** considerando os critérios de adequação, exaustividade, homogeneidade, exclusão mútua e consistência, as unidades de análise são classificadas em categorias;
- (4) **Descrição:** resume-se a adicionar um descritor para cada categoria (que reflete o conjunto de significados de suas

unidades) e analisar a frequência e a porcentagem de itens no conjunto de categorias criado;

- (5) **Interpretação:** consiste na avaliação dos materiais de análise para a produção de teorias que emergem das informações e das categorias em um processo de reflexão iterativo.

4 RELATOS DE EXPERIÊNCIA EM ENSINO

Esta seção apresenta relatos das experiências em ensino realizadas em duas edições do componente curricular Resolução de Problemas IV. O primeiro deles diz respeito ao sistema de micro-ondas utilizado como foco na edição 2014. Detalhes desta edição são apresentados na subseção 4.1. Enquanto isso, a subseção 4.2 apresenta detalhes sobre a edição 2016 da mesma disciplina, na qual foi empregado um sistema para controle de uma máquina de bebidas como foco do trabalho.

4.1 Relato: Micro-ondas

Na edição 2014, o componente curricular de RP IV foi dividido em quatro marcos. A Figura 3 apresenta os tópicos abordados em cada marco dessa edição e sua duração.

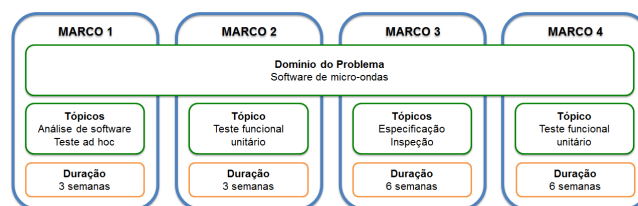


Figura 3: Marcos RP IV edição 2014.

Conforme pode ser observado na Figura 4, o software de micro-ondas foi estabelecido como domínio do problema para todo o componente curricular. Trata-se de uma simulação de software embarcado, implementada em linguagem Java. Na disciplina, diferentes módulos (display, teclado, simulador gráfico, unidade lógica, etc.) foram disponibilizados pelos professores juntamente com seu código-fonte.

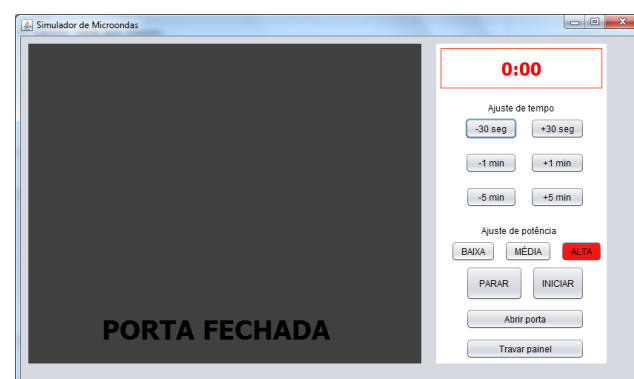


Figura 4: Simulador gráfico do software de micro-ondas de RP IV edição 2014.

² A escala Likert [17] adotada se baseia em um esquema clássico de 5 pontos de concordância: discordo totalmente, discordo, indiferente/indeciso, concordo, e concordo totalmente.

³ Termo adotado por Moraes [20] para identificar a etapa de separação em unidades.

O marco 1 abordou a análise do software disponibilizado (incluindo executáveis e demais artefatos) e testes *ad hoc*. O marco 2 se concentrou em teste funcional de unidade sistemático e na correção de erros. O marco 3 envolveu elicitação e especificação de novos requisitos, inspeção de código e documentação. O marco 4 se concentrou em implementação dos novos requisitos, teste funcional de unidade, correção de erros e validação com usuários. No âmbito desse artigo, os marcos 2 e 4 se destacam por abordar teste de software. Para isso, os alunos formaram grupos de quatro componentes e nenhuma restrição foi aplicada na formação desses grupos. Essa edição do componente curricular contou com a participação de quatro professores que avaliaram grupos diferentes em cada marco.

Conflitos podem ocorrer devido a “necessidades altamente divergentes de duas partes, um desentendimento que ocorre entre duas pessoas, ou como resultado de um fator intangível” [16]. Entretanto aqui, eles são provocados propositalmente pelos professores que solicitam funcionalidades e características conflitantes para a versão evoluída do software de micro-ondas. Conforme *International Institute of Business Analysis* (IIBA) [13], dentre as atividades aplicadas para resolução de conflitos estão:

- (i) facilitar e mediar a comunicação entre as partes;
- (ii) identificar os verdadeiros interesses de cada parte;
- (iii) identificar vantagens e desvantagens técnicas relevantes sobre as decisões possíveis; e
- (iv) mediar um acordo que resolva as diferenças da melhor forma possível.

A avaliação individual é realizada por um professor em apresentações no final de cada marco. Embora os grupos apresentassem em conjunto os resultados alcançados até aquele marco, os professores questionavam individualmente as contribuições e a participação de cada componente no trabalho realizado. Além disso, os professores avaliaram a qualidade geral (*i.e.* consistência, corretismo e completude) do conjunto de técnicas e artefatos esperado para aquela etapa do componente curricular.

A avaliação dos grupos ocorre apenas no final do componente curricular, após o quarto marco. Essa avaliação considerou a integração e a sumarização dos resultados de cada marco anterior na forma de software funcional e de relatórios.

As principais tecnologias adotadas nessa edição do componente curricular foram a linguagem de programação Java e o *framework* de testes de unidade JUnit 4. Os artefatos disponibilizados pelos professores foram os códigos-fonte Java dos componentes e uma modelagem simplificada do sistema, usando diagramas de casos de uso e diagramas de classes em *Unified Modeling Language* (UML). Os artefatos entregues pelos grupos nos marcos 2 e 4 foram os códigos-fonte em Java, os códigos de teste JUnit, e o relatório de testes e inspeções (sem um formato pré-determinado).

4.2 Relato: Máquina de bebidas

Na edição 2016 do componente curricular, a estratégia de divisão e avaliação por marcos foi novamente utilizada. A Figura 5 apresenta os tópicos, os contextos dos problemas e a duração de cada marco.

O marco 1 teve como cenário a execução de testes sobre o software de uma máquina automática de bebidas e, tangencialmente, análise de software. Os marcos 2 e 3 tiveram como cenário a execução de testes e inspeções sobre o software de gerência de

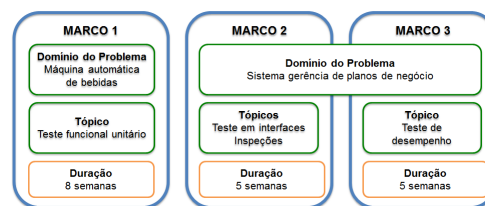


Figura 5: Marcos de RP IV edição 2016 [6].

planos de negócios desenvolvido pela empresa júnior Ideiah da Unipampa.

Devido ao marco 1 ser o único a integrar estratégias de gamificação e devido a não ter sido realizado planejamento e coleta de dados nos marcos 2 e 3, apenas o marco 1 é discutido neste trabalho. No contexto da máquina automática de bebidas, os módulos foram desenvolvidos e pré-compilados pelos professores e entregues aos alunos, simulando a produção por empresas terceirizadas. Dentre os módulos constam: *dispenser* de bebidas, entrada de dinheiro, *dispenser* de dinheiro, *display*, etc.

Essa edição contou com a participação de dois professores. Nos dois primeiros marcos cada professor avaliou um conjunto diferente de grupos, enquanto que no último marco, os dois professores avaliaram todos os grupos.

As tarefas do primeiro marco se resumem a desenvolver o núcleo lógico da máquina (ou módulo controlador) e a testar os módulos disponibilizados. Para isso, os alunos formaram grupos de três elementos.

A avaliação individual é realizada semanalmente por um professor. Ela inclui a avaliação do trabalho realizado naquela semana e definição das atividades a serem realizadas na próxima semana. A promessa de atividades a serem realizadas fundamenta a próxima avaliação e sempre passa pela ponderação do professor. Esse é responsável por indicar se a quantidade e a profundidade das atividades prometidas está adequada, aquém ou além do esperado. A média aritmética simples das avaliações semanais representa o resultado da avaliação individual final no componente curricular.

Conforme mencionado na seção 2.1, foram adicionados elementos de gamificação [19] para o primeiro marco do componente curricular. Estes incluem: enredo, desafios, recursos, objetivos, cooperação, competição, desempenho e pontuação.

Estes elementos foram escolhidos considerando as limitações impostas pelo componente curricular (tempo de acompanhamento em sala de aula, esquemas de avaliação da universidade, etc.). O enredo proposto indicou que os alunos pertencem a uma empresa de desenvolvimento de aplicativos embarcados que terceiriza o desenvolvimento de algumas partes do sistema. Seus desafios incluíam desenvolver o núcleo lógico e testar os componentes terceirizados. Para isso, contaram com os seguintes recursos: os professores de RP IV, os professores dos componentes curriculares de suporte, os livros básicos e complementares adotados, e o *framework* JUnit. Nesse marco, os objetivos específicos foram:

- encontrar falhas nos componentes entregues pelos professores que foram propositalmente contaminados com *bugs*;

- encontrar outras falhas (não intencionais) nos componentes entregues pelos professores;
- desenvolver o módulo com a lógica para a máquina de bebidas;
- passar em todos os testes de integração realizados pelos professores;
- passar em todos os testes de integração executados por outro grupo;
- criar testes de integração que encontre erros no projeto de outro grupo.

O trabalho em grupo exige cooperação, divisão de tarefas e autogerência para atingir os objetivos específicos. Um tipo especial de teste de software denominado **teste do amigo vampiro** foi empregado para imprimir caráter de competição ao marco. A criação desse elemento, cujo nome foi proposto em analogia aos mortos-vivos que se alimentam de sangue dos outros, está contextualizada no enredo criado, seguindo a metáfora de um jogo. Tal teste permitia que, no evento final do marco, cada grupo executasse testes de integração sobre o projeto de outro grupo. Caso encontrasse erros, o primeiro grupo (criador dos testes) se apropriava de parte da nota final do segundo grupo (criador do código testado). O processo cíclico desses testes é ilustrado na Figura 6.

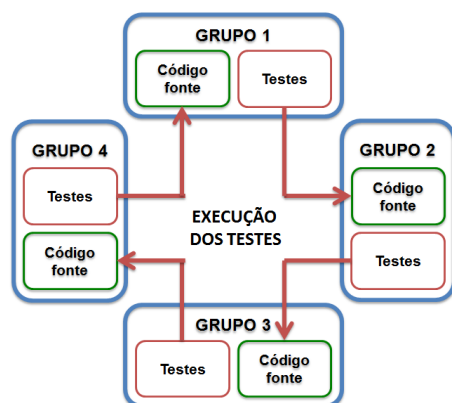


Figura 6: Ilustração dos testes do amigo vampiro de RP IV edição 2016.

A pontuação individual dos alunos e a pontuação dos grupos constituem a nota final do aluno, sendo que cada uma delas contribui com 50%. Destaca-se que a pontuação não é uma questão apenas de endogenia, quando algo possui valor apenas dentro do próprio jogo [28]. De fato, as pontuações nas avaliações e a abordagem do teste do amigo vampiro definem a nota final do aluno e consequentemente a sua aprovação na disciplina. Assim, a evolução do aluno no currículo depende de seu sucesso no cumprimento dos desafios apresentados no jogo. Como consequência, os professores observaram que este fator gerou um desconforto inicial em alguns alunos e, por outro lado, um maior comprometimento, causando um impacto positivo na qualidade dos artefatos de teste criados.

A avaliação dos grupos foi realizada em eventos que ocorreram no final de cada marco. No evento referente ao marco 1, os professores recuperaram os códigos-fonte e os códigos de teste dos diversos grupos para realizar as seguintes atividades:

- executar os testes (funcionais de unidade) do grupo sobre os códigos-fonte com defeito disponibilizados pelos professores;
- executar a auditoria⁴ (testes de sistema) dos professores sobre o código-fonte do grupo; e
- escolher e executar os testes do amigo vampiro (testes de integração) de outro grupo sobre os códigos-fonte do grupo sendo avaliado (ilustrado na Figura 6).

Na avaliação, encontrar os cinco erros propositalmente inseridos no código disponibilizado pelos professores gerava a nota base do grupo. Na auditoria, os professores ao encontrar erros no código produzido pelo grupo geravam descontos na nota, sem limite de descontos. Os testes do amigo vampiro de um grupo que encontrassem erros no código produzido por outro grupo se apropriaram de parte da nota, dentro de um limite máximo de transferência de nota.

As tecnologias adotadas no primeiro marco dessa edição do componente curricular foram a linguagem Java e o *framework* de testes de unidade JUnit 4. Os artefatos disponibilizados pelos professores foram os arquivos *.class* referentes aos componentes terceirizados pré-compilados em Java, e a especificação do sistema no modelo Oracle MD050. Os artefatos entregues pelos grupos no marco 1 foram os códigos-fonte em Java, os códigos de teste JUnit; a especificação e o relatório de testes com base no modelo IEEE SA 829-2008 [12].

5 RESULTADOS OBTIDOS

O componente de RP IV foi avaliado em duas perspectivas distintas nos anos de 2014 e 2016. Na primeira delas o objetivo consistiu em observar mudanças na percepção dos alunos acerca dos principais tópicos abordados. A segunda visou avaliar a introdução de novas dinâmicas no funcionamento do componente curricular para aumentar o engajamento dos alunos. Nas seções seguintes, são apresentados detalhadamente as avaliações aplicadas e seus resultados.

5.1 Resultados: Micro-ondas

A avaliação da mudança na percepção dos alunos em relação a teste de software se fundamentou em realizar uma pergunta no começo e no final do componente curricular.

A pergunta **O que é teste de software e qual é sua importância?** foi disponibilizada no ambiente Moodle, e os alunos tiveram aproximadamente 10 minutos para respondê-la durante aula. Foram registradas 18 respostas na primeira aula e 21 respostas na última aula.

A técnica de análise de conteúdo [20] foi adotada para examinar as respostas. Na preparação de informações, as respostas foram exportadas a partir do ambiente Moodle para planilhas eletrônicas locais. **Em seguida, uma leitura permitiu formar uma visão geral das respostas e confirmar que o processo de análise de conteúdo proposto era adequado.**

Na etapa de unitarização dos conteúdos, cada uma das 39 respostas registradas foi lida e analisada, resultando em 109 unidades de análise que são apresentadas classificadas nas Tabelas 1 e 2.

⁴No cenário criado para gamificação, "auditoria" é a atividade na qual os professores testam os códigos gerados pelos alunos no intuito examinar as atividades de desenvolvimento.

Na etapa de classificação, as unidades de conteúdo foram agrupadas por similaridade, formando as mesmas 11 categorias nos dois momentos: primeira aula e última aula. Como exemplo de unidades de análise similares que foram concentrados em apenas uma categoria temos: “encontrar os erros”, “encontrar bugs” e “descobrir falhas”.

Na fase de descrição, essas categorias foram nomeadas com um rótulo comum que resume seus conteúdos. As Tabelas 1 e 2 apresentam as categorias das respostas à pergunta realizada, respectivamente, na primeira aula e na última aula. Na análise foram incluídas porcentagens para refletir as proporções de alunos, dado que o número de respostas não foi o mesmo para os dois momentos.

Tabela 1: Categorias de respostas para a pergunta “O que é teste de software e qual é sua importância?” na primeira aula - RP IV edição 2014.

ID	Categoria	N	%
1	Encontrar/buscar/descobrir erros/falhas/bugs/defeitos	14	78%
2	É muito importante	7	39%
3	Mostrar o que não funciona	6	33%
4	Possui vários métodos e técnicas	6	33%
5	Testar de acordo com requisitos	3	17%
6	Aumentar confiança	3	17%
7	Aumentar qualidade	3	17%
8	Garantir funcionamento correto	2	11%
9	Garantir mínimo de erros	2	11%
10	Testar de acordo com planejamento	1	6%
11	Presente em todo o ciclo de desenvolvimento	1	6%

Tabela 2: Categorias de respostas para a pergunta “O que é teste de software e qual é sua importância?” na última aula - RP IV edição 2014

ID	Categoria	N	%
1	Aumentar qualidade	15	71%
2	É muito importante	11	52%
3	Encontrar/buscar/descobrir erros/falhas/bugs/defeitos	11	52%
4	Aumentar confiança	4	19%
5	Mostrar o que não funciona	4	19%
6	Garantir funcionamento correto	3	14%
7	Garantir mínimo de erros	3	14%
8	Possui vários métodos e técnicas	3	14%
9	Presente em todo o ciclo de desenvolvimento	3	14%
10	Testar de acordo com requisitos	3	14%
11	Testar de acordo com planejamento	1	5%

Observa-se uma alteração significativa da perspectiva dos alunos em relação à pergunta realizada. O elemento mais frequente nas respostas passou do objetivo do teste de software (categoria 1 na Tabela 1) para um benefício direto do teste de software adequadamente realizado (categoria 1 na Tabela 2). Em outra perspectiva, percebe-se que a categoria 7 da Tabela 1 substituiu as categorias 3 e 4 dentre as mais frequentes. É possível que essa mudança seja fruto do reconhecimento pelos alunos do aumento de qualidade

do produto final após o planejamento e a aplicação dos testes no componente curricular.

Declarações explícitas da importância de teste de software aumentaram proporcionalmente nas respostas da última aula (Tabela 2). E, a grande concentração de unidades de análise nas primeiras três categorias da (Tabela 2) revela também a convergência da percepção dos alunos.

Dessa forma, considera-se um visível amadurecimento dos alunos em relação a teste de software ao deixarem de vê-lo como uma coleção de técnicas para encontrar falhas e passarem a compreendê-lo como um processo importante na controle de qualidade de software. Por fim, as demais categorias não sofreram modificações significativas que justifiquem mudanças de ponto de vista dos estudantes.

5.2 Resultados: Máquina de bebidas

Para avaliar a percepção dos sobre a execução do componente curricular e o uso de gamificação na edição de 2016, foi criado um questionário. Este contém dez itens obrigatórios de múltipla escolha e dois itens de texto livre não obrigatórios. Os dez itens de múltipla escolha seguiram a escala Likert [17] que é detalhada na seção 3. Os itens que compunham o questionário estão divididos em duas partes.

A primeira parte diz respeito apenas ao componente curricular Resolução de Problemas e é composta pelas seguintes afirmações:

- 1) A disciplina possibilitou aplicação prática dos conceitos teóricos abordados no curso.
- 2) A disciplina possibilitou aprender, de forma autônoma e independente, métodos e técnicas além daqueles previstos na ementa.
- 3) A disciplina possibilitou trabalhar colaborativamente em grupo com objetivo de entregar o projeto no prazo.
- 4) A disciplina possibilitou trabalhar colaborativamente em grupo com objetivo de entregar o projeto com qualidade.
- 5) Os cenários criados pelos professores representam possíveis cenários reais.

A segunda parte dos itens diz respeito aos procedimentos adotados no marco da máquina de bebidas. Ela é composta pelos seguintes itens:

- 6) Colaborei mais com meu grupo no problema da máquina de bebidas que no problema do sistema de gerência de planos de negócio.
- 7) Me senti mais seguro no problema da máquina de bebidas por saber a quantidade exata de erros que precisava encontrar.
- 8) Me senti mais motivado no problema da máquina de bebidas pela auditoria.
- 9) Me senti mais motivado no problema da máquina de bebidas pelos testes do amigo vampiro.
- 10) A regra do amigo vampiro no problema da máquina de bebidas causou problemas de relacionamento com outros grupos.
- 11) Diga o que mais gostou no problema da máquina de bebidas. (opcional)
- 12) Diga o que menos gostou no problema da máquina de bebidas. (opcional)

O sumário das respostas para os itens de múltipla escolha do questionário pode ser observado nas Figuras 7 e 8. Nesses gráficos, o eixo vertical se refere à quantidade de participantes que escolheram uma determinada resposta para cada item. O eixo horizontal representa cada uma das afirmações.

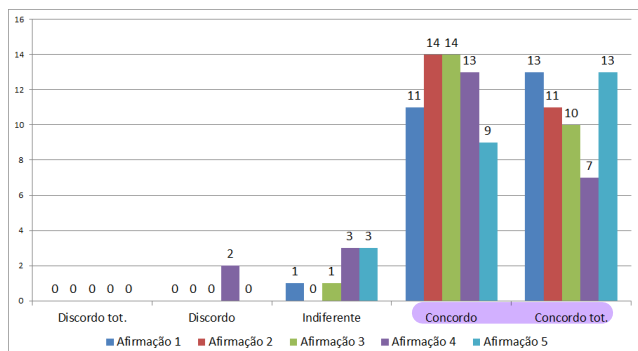


Figura 7: Resultados da parte 1 do questionário de RP IV edição 2016 [5].

A mediana e a moda das respostas indicam, conforme a Figura 7, que a turma da edição 2016 concorda significativamente que o componente curricular de RP IV atende objetivos de ABP. Isto inclui permitir a prática de conceitos teóricos, aprender métodos e técnicas não previstos na ementa da disciplina, trabalhar colaborativamente com foco no prazo, e trabalhar colaborativamente com foco na qualidade da entrega. Adicionalmente, há uma tendência da turma concordar totalmente que os problemas propostos pelos professores representam possíveis problemas reais que encontrariam na indústria de software.

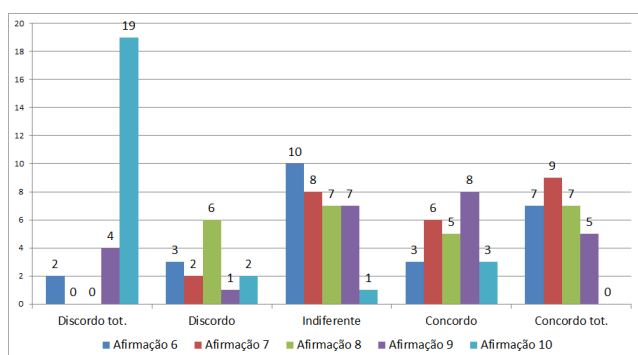


Figura 8: Resultados da parte 2 do questionário de RP IV edição 2016 [5].

Nota-se na Figura 8 que a turma da edição 2016, no geral, é indiferente ou indecisa em relação a colaborar mais como grupo e se sentir mais motivada pela auditoria no problema da máquina de bebidas. Assim, uma possibilidade é que os elementos de cooperação e desafio introduzidos por meio de gamificação não contribuíram significativamente para mobilizar os alunos.

Observa-se também que há uma tendência na turma em concordar (pela mediana) ou concordar totalmente (pela moda) que se sentiram mais seguros sabendo a quantidade exata de erros. Além disso, houve maior motivação devido aos testes do amigo vampiro. Dessa forma, objetivos específicos claros e competição entre os grupos, como elementos de gamificação, favoreceram o engajamento dos alunos para realizar as tarefas. Adicionalmente, há uma tendência expressiva na turma em discordar completamente que os testes do amigo vampiro tenham causado problemas entre os grupos, ou seja, essa abordagem não implica em problemas de relacionamento, apresentando uma abordagem segura para competição.

Os depoimentos obtidos nos itens 11 e 12 do questionário foram também investigados por meio da análise de conteúdo [20]. As 17 respostas para o item 11 e as 16 respostas para o item 12 foram tabuladas em planilhas. A avaliação preliminar dos textos permitiu identificar que o processo de análise de conteúdo seria adequado. Na etapa de unitarização, foram identificados 30 elementos para o item 11 e 18 elementos para o item 12. A etapa de classificação resultou em 18 categorias distintas para elementos do item 11 e 10 categorias para elementos do item 12. A heterogeneidade de perspectiva dos participantes nas respostas às perguntas abertas resultou em uma maior quantidade de categorias. Os resultados da etapa de descrição com os rótulos das categorias se encontram nas Tabelas 3 e 4. Nessas tabelas são apresentados os valores absolutos (N) e relativos (%).

Tabela 3: Categorias para o item 11 “Diga o que mais gostou no problema da máquina de bebidas.” - RP IV edição 2016 [5].

ID	Categoria	N	%
1	Testar códigos dos colegas roubando nota	6	35%
2	Planejar e criar testes de unidade	3	18%
3	Usar JUnit para testes	3	18%
4	Dinâmica da disciplina e abordagem de avaliação	2	12%
5	Programar em Java parte da solução	2	12%
6	Trabalhar com técnicas e tecnologias já conhecidas	2	12%
7	Avaliação justa	1	6%
8	Avaliações semanais organizadas	1	6%
9	Desafios difíceis e factíveis	1	6%
10	Escolha do problema	1	6%
11	Escopo bem definido	1	6%
12	Esforço consolidado em conhecimento	1	6%
13	Incentivos pela descoberta de erros	1	6%
14	Nota	1	6%
15	Planejar e criar testes funcionais	1	6%
16	Procurar erros do sabotador	1	6%
17	Sistema verossímil com problemas do mercado	1	6%
18	Testes na auditoria	1	6%

Em concordância com a análise da Figura 8, os dados da Tabela 3 indicam que a dinâmica de competição por meio de testes cruzados entre os grupos e o roubo de notas (categorias 1 e 4) foi um fator positivo para grande parte dos estudantes.

É possível observar também que as categorias 2, 3, 5 e 6 elencadas na Tabela 3 refletem uma alteração em relação ao funcionamento tradicional de ABP na Unipampa. Enquanto tradicionalmente é

exigida autonomia para aprendizado de novos conteúdos e técnicas desde o começo dos componentes curriculares de RP, o primeiro marco desse RP exigiu habilidades e conhecimentos já adquiridos em semestres anteriores, permitindo que os estudantes pudessem se concentrar mais em resolver o problema e desenvolvessem suas habilidades em profundidade nos tópicos abordados. Essa abordagem facilitou a integração com as estratégias de gamificação.

Dentre as categorias na Tabela 3 que contiveram apenas uma unidade de conteúdo, destacam-se as categorias 9, 11 e 13 por estarem relacionadas à dinâmica de gamificação utilizada: a missão, o enredo e os objetivos específicos devem estar claros para que os desafios possam ser superados e para que a recompensa na forma de pontos seja alcançável (veja a seção 4.2 para mais detalhes sobre os elementos de gamificação usados).

Tabela 4: Categorias para o item 12 “Diga o que menos gostou no problema da máquina de bebidas.” - RP IV edição 2016 [5].

ID	Categoria	N	%
1	Descontos muito rigorosos nos testes dos professores	6	38%
2	Acompanhamentos semanais pouco rigorosos	4	25%
3	Atualizações no escopo	1	6%
4	Compreender e produzir documentação	1	6%
5	Nota	1	6%
6	Pouco tempo para fazer todas as atividades	1	6%
7	Problema inicialmente confuso	1	6%
8	Problemas no JUnit na apresentação	1	6%
9	Programar em Java	1	6%
10	Testar códigos dos colegas roubando nota	1	6%

É visível na Tabela 4 que grande parte dos participantes teve problemas com penalidades rigorosas (categoria 1). Essas penalidades ocorriam quando o código produzido pelos estudantes para a lógica da parte de controle da máquina de bebidas não passava nos testes funcionais de unidade feitos pelos professores (que representavam a auditoria do sistema). Esse mesmo problema ocorre no projeto tradicional de jogos que elenca a punição como um importante gerador de valor endógeno e um fator de aumento de desafio, mas que deve ser equilibrada para não causar frustração e manter o participante na zona de “fluxo” [27].

Em relação ao rigor dos acompanhamentos individuais semanais (categoria 2), houve uma percepção inversa à categoria 1 que pode ser observada na Tabela 4. Nesse caso, as avaliações semanais descritas na subseção 4.2 tiveram como funções principais: ajudar a superar obstáculos, esclarecer dúvidas e manter os estudantes num ritmo sustentável de trabalho. A primeira e a segunda funções estão relacionadas ao elemento **Recursos** [19] de gamificação, em especial **Help**, e foram cumpridas. A terceira está relacionada ao elemento **Desempenho** [19] e foi responsável por alertar os estudantes quando estivessem se esforçando pouco ou produzindo pouco por meio de avaliações individuais. Em referência à categoria 2, é inconclusivo se houve falta de exigência durante as avaliações individuais por parte dos professores, levando a produtos de baixa

qualidade na apresentação final, ou se houve problemas da auto-gestão do grupo, fazendo com que os próprios estudantes avaliassem de forma errônea o quão próximos estavam dos objetivos específicos da missão.

Das 8 categorias restantes na Tabela 4 que contiveram apenas um item isolado, destacam-se a categoria 7 pela introdução de regras relacionadas a dinâmicas de gamificação e a categoria 3 que, na realidade, trata-se de um equívoco do estudante, pois houve apenas atualizações de detalhamento nos requisitos e o escopo do trabalho permaneceu o mesmo durante todo o marco.

6 CONSIDERAÇÕES FINAIS

Neste trabalho é apresentado um relato da integração de Aprendizagem Baseada em Problemas (ABP), gamificação e ensino de teste de software. O relato cobre as edições de 2014 e 2016 do componente curricular Resolução de Problemas IV (RP IV), do curso de Engenharia de Software na Unipampa. Em RP IV são abordados os temas de Engenharia de Requisitos e Verificação e Validação (V&V) de software, embora o presente trabalho tenha como foco a área de teste de software.

Com base nos resultados apresentados no relato, pode-se afirmar que não houve incompatibilidade entre ABP e gamificação, e que a integração proporciona benefícios à aprendizagem. Na edição 2014 foi observado amadurecimento da perspectiva dos alunos nos conhecimentos na área de teste de software por meio de ABP. Na edição 2016, os resultados demonstram que os alunos reconhecem que o componente curricular ofereceu um espaço colaborativo de prática e aprendizagem autônoma. Além disso, no ponto de vista dos alunos, a definição de objetivos nítidos em ABP e as dinâmicas de competição da gamificação favorecem a formação de um ambiente incentivador ao aprendizado.

Os professores que conduziram RP IV perceberam nas edições descritas neste relato que o uso de gamificação pode gerar uma competição acirrada. Entretanto, cabe ressaltar que os alunos jogaram com *fair play*⁵ e eles mesmos concordaram que não aconteceram situações de conflito entre os grupos. Outra percepção dos docentes, que vai ao encontro a avaliação realizada pelos alunos, relaciona-se aos descontos de nota realizados pelos docentes. Na visão dos discentes, os professores foram excessivamente rigorosos na execução das auditorias nos artefatos desenvolvidos. Uma forma de mitigar esse problema seria convidar outro professor para atuar como revisor do processo de auditoria.

Considerando que o componente curricular de RP IV ocorre em paralelo com o componente de V&V, não é possível medir exatamente a influência do componente de V&V nos resultados obtidos em RP IV. Além disso, a coleta de dados foi realizada apenas em duas edições de RP IV. Consequentemente, os resultados não são generalizáveis, embora apresentem tendências significativas.

Outro ponto a ser considerado é a ausência da comparação do desempenho dos alunos entre edições do componente curricular. É possível realizar uma comparação baseada nas notas obtidas pelos alunos em cada critério avaliado. Em geral, os componentes de RP consideram critérios semelhantes para avaliação. Contudo, os

⁵Termo originalmente relacionado à ética no meio esportivo que define que jogadores devem se portar de maneira justa e leal com o adversário.

instrumentos de avaliação ainda não são padronizados para comparar o desempenho. Além disso, é necessário considerar que cada edição de RP IV pode ser ministrada por um grupo diferente de professores, causando significativo viés na comparação por notas.

Com o objetivo de mitigar os pontos negativos levantados no relato, como trabalhos futuros se planeja definir um processo e artefatos para a condução do ensino de teste de software utilizando ABP. Dessa forma, pode-se comparar os resultados da aplicação de ABP no ensino de teste de software entre as diferentes turmas de RP IV. A definição de um processo possibilitaria ainda que os estudos conduzidos na Unipampa sejam replicados em outros contextos e outras instituições. Assim, pode-se obter um maior volume de dados sobre a abordagem, o que permitirá realizar avaliações como maior exatidão.

Com base nos pontos positivos e negativos levantados neste trabalho, ABP se apresenta não apenas como uma abordagem exclusiva para ser adotada no processo de ensino-aprendizagem, mas se mostra adequada à integração de outras estratégias pedagógicas. A gamificação, por exemplo, propiciou um ambiente de engajamento diferenciado para os alunos, como se pode observar nos resultados obtidos. Na perspectiva dos componentes curriculares RP da Unipampa, inúmeras podem ser as experiências que incorporam estratégias cujos propósitos sejam complementares e convergentes.

AGRADECIMENTOS

Os autores gostariam de agradecer aos professores Cristiano Tolfo e Sam Devincenzi pelo apoio nos componentes curriculares de Resolução de Problemas e pelas contribuições neste artigo.

REFERÊNCIAS

- [1] D. E. Allen, R. S. Donham, and S. A. Bernhardt. 2011. Problem-based learning. *New Directions for Teaching and Learning* 2011, 128 (2011), 21–29. DOI: <http://dx.doi.org/10.1002/tl.465>
- [2] Howard Barrows. 1996. Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning* 1996, 68 (1996), 3–12.
- [3] Marcia Cera, Mateus Dal Forno, and Vanessa Gindri Vieira. 2012. A Proposal to Teach Software Engineering based on Problem Solving. *Brazilian Journal of Computers in Education* 20, 03 (2012), 116–132.
- [4] C.E.S.A.R. 2017. Mestrado Profissional em Engenharia de Software. (2017). Retrieved July 20, 2017 from <http://www.cesar.edu.br/course/mestrado-profissional-em-engenharia-de-software>
- [5] Jean Felipe Patikowski Cheiran and Elder de Macedo Rodrigues. 2017. Engenharia de Requisitos e V&V: Aprendizagem Baseada em Problemas Associada à Gamificação. In *Aprendizagem Baseada em Problemas na Engenharia de Software: Relatos de Experiência*, Cristiano Tolfo (Ed.). Ediurcamp, Bagé, Chapter 4, 71–91.
- [6] Aline Vieira de Mello, Amanda Meincke Melo, Andréa Sabedra Bordin, Cristiano Tolfo, Elder de Macedo Rodrigues, Ewerson Luiz de Souza Carvalho, Fábio Natanuel Kepler, Gilleanes Thorwald Araujo Guedes, Jean Felipe Patikowski Cheiran, and Jo ao Pablo Silva da Silva. 2017. *Aprendizagem Baseada em Problemas na Engenharia de Software: Relatos de Experiência*. Ediurcamp, Bagé.
- [7] Sebastian Deterding, Rilla Khaled, Lennard Nacke, and Dan Dixon. 2011. Gamification: Toward a Definition. In *ACM CHI Conference on Human Factors in Computing Systems 2011*. ACM, Vancouver, 1–4.
- [8] Virginia Farias, Carla Moreira, Emanuel Coutinho, and Ismayle S. Santos. 2012. iTest Learning: Um Jogo para o Ensino do Planejamento de Testes de Software. In *V Fórum de Educação em Engenharia de Software*. ACM, Natal, 8.
- [9] Figuerêdo, C.O.; Santos, S.C.; Borba, P.H.; Alexandre, G.H. 2011. Using PBL to develop software test engineers. In *14th IASTED International Conference on Computers and Advanced Technology in Education*. Acta, Cambridge, 7.
- [10] Luiz Paulo Franz, Jo ao Pablo S. da Silva, and Jean Felipe P. Cheiran. 2014. O uso de Coding Dojo no aprendizado colaborativo de programação de computadores. *Revista Novas Tecnologias na Educação* 12, 2 (2014), 9.
- [11] L. F. S. Hoffmann, L. E. G. Vasconcelos, E. Lamas, A. M. Cunha, and L. A. V. Dias. 2014. Applying Acceptance Test Driven Development to a Problem Based Learning Academic Real-Time System. In *11th International Conference on Information Technology: New Generations*. IEEE, Las Vegas, 3–8. DOI: <http://dx.doi.org/10.1109/ITNG.2014.63>
- [12] IEEE. 2008. Standard for Software and System Test Documentation. (2008). Retrieved July 20, 2017 from <https://standards.ieee.org/findstds/standard/829-2008.html>
- [13] IIBA. 2011. *Um guia para o Corpo de Conhecimento de Análise de Negócios* (2 ed.). International Institute of Business Analysis, Toronto.
- [14] F. F. Jorge, C. I. M. Bezerra, E. F. Coutinho, J. M. Monteiro, and R. M. C. Andrade. 2015. A Evolução do Jogo iTestLearning para o Ensino das Atividades de Execução de Testes de Software. In *XX Conferência Internacional sobre Informática na Educação*. Universidad de Chile, Santiago, 295–305.
- [15] R. Larson and B. Faber. 2003. *Elementary Statistics* (2 ed.). Prentice Hall, Boston.
- [16] R. Lewicki, D.Saunders, and J. Minton. 1997. *Essentials of Negotiation* (2 ed.). Irwin, Chicago.
- [17] Rensis Likert. 1932. A technique for the measurement of attitudes. In *Archives of Psychology*. APA, New York, 1–55.
- [18] M. A. Marconi and E. M. Lakatos. 2010. *Fundamentos de metodologia científica*. Atlas, São Paulo.
- [19] Cristina Martins and Lucia Maria Martins Giraffa. 1999. Gamificação nas práticas pedagógicas em tempos de cibercultura: proposta de elementos de jogos digitais em atividades gamificadas. In *XI Seminário de Jogos Eletrônicos, Educação e Comunicação*. PUCRS, Porto Alegre, 11–19.
- [20] R. Moraes. 1999. Análise de Conteúdo. *Revista Educação* 22, 37 (1999), 7–32.
- [21] Glenford J. Myers, Corey Sandler, and Tom Badgett. 2011. *The art of software testing*. John Wiley & Sons, Hoboken.
- [22] Roger S. Pressman. 2010. *Software Engineering: A Practitioner's Approach* (7 ed.). McGraw-Hill Education, New York.
- [23] Ita Richardson and Yvonne Delaney. 2009. Problem Based Learning in the Software Engineering Classroom. In *22nd Conference on Software Engineering Education and Training*. IEEE, Washington, 174–181.
- [24] I. Richardson, L. Reid, S. B. Seidman, B. Pattinson, and Y. Delaney. 2011. Educating software engineers of the future: Software quality research through problem-based learning. In *24th IEEE-CS Conference on Software Engineering Education and Training*. IEEE, Honolulu, 91–100.
- [25] R. H. Sampieri, C. F. Collado, and M. P. B. Lucio. 2013. *Metodologia de Pesquisa*. Penso, Porto Alegre.
- [26] S. C. Santos, M. C. M. Batista, A. P. Cavalcanti, J. O. Albuquerque, and S. Meira. 2008. Usando pbl na qualificação de profissionais em engenharia de software. In *I Fórum de Educação em Engenharia de Software*. ACM, Campinas, 9.
- [27] J. Schell. 2008. *The art of game design: a book of lenses* (1 ed.). Morgan Kaufmann, San Francisco.
- [28] Jesse Schell. 2011. *A Arte do Game Design: o livro original* (1 ed.). Elsevier, Rio de Janeiro.
- [29] H. G. Schmidt, J. I. Rotgans, and E. H. J. Yew. 2011. The process of problem-based learning: what works and why. *Medical Education* 45, 8 (2011), 792–806. DOI: <http://dx.doi.org/10.1111/j.1365-2923.2011.04035.x>
- [30] Marcello Thiry, Alessandra Zoucas, and Rafael Queiroz Gonçalves. 2010. Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo. In *Anais do XXVII Simpósio Brasileiro de Informática na Educação*. SBC, João Pessoa, 10.
- [31] Unipampa. 2015. Projeto Político-Pedagógico do Curso de Engenharia de Software. (2015). Retrieved July 20, 2017 from <http://cursos.unipampa.edu.br/cursos/engenhariadesoftware/?pagina.fixa=ppc>
- [32] Pedro Valle, Ellen Francine Barbosa, and José Maldonado. 2015. Um mapeamento sistemático sobre ensino de teste de software. In *Simpósio Brasileiro de Informática na Educação*. SBC, Maceió, 71.
- [33] Cristiane G. von Wangenheim and Dioni A. da Silva. 2009. Qual Conhecimento de Engenharia de Software é Importante para um Profissional de Software?. In *II Fórum de Educação em Engenharia de Software*. ACM, Fortaleza, 8.
- [34] R. S. Wazlawick. 2014. *Metodologia de Pesquisa para Ciência da Computação* (2 ed.). Elsevier, Rio de Janeiro.