

Um Sistema Preliminar de Apoio à Gestão de Controle de Qualidade em Jogos Digitais

Gustavo Carvalho da Costa¹, João Pablo Silva da Silva¹

¹Universidade Federal do Pampa (UNIPAMPA)
97546-550 – Av. Tiaraju, 810 – Alegrete – RS – Brazil

{gustavocosta.aluno, joaosilva}@unipampa.edu.br

Abstract. *The following paper has the objective to address the quality control field in digital games, focusing mainly on applicable techniques. The problem to be explored is based on the need from the gaming development teams to prioritize and implement testing techniques that aim for an acceptable final product quality, in view of requirements raised. To validate this study, it was proposed a survey with quality control field professionals, where 10 scenarios with testing phase features were evaluated. Finally, as a solution, a preliminary application was developed which offers support to developers proving a description of the best testing techniques to be used.*

Resumo. *Este artigo possui como objetivo abordar a área de controle de qualidade em jogos digitais, possuindo como maior foco as técnicas aplicáveis. O problema a ser explorado se baseia na necessidade das equipes de desenvolvimento de jogos em priorizar e implementar técnicas de teste que busquem uma qualidade aceitável do produto final, tendo em vista os requisitos levantados. Para a validação do estudo, foi proposta uma pesquisa com profissionais da área de controle de qualidade, onde houve a avaliação de 10 cenários com características em relação a etapa de testes. Por fim, como solução, foi desenvolvido uma aplicação preliminar que oferece suporte aos desenvolvedores provendo a descrição das melhores técnicas de teste à serem utilizadas.*

1. Introdução

Na indústria de jogos digitais, especificamente em empresas nas quais desenvolvem e publicam jogos com grande sucesso comercial, a qualidade final em seus produtos é de extrema importância. Apesar disso, observa-se a quantidade de falhas encontradas em diversos jogos digitais após o lançamento oficial, ficando claro que em alguns projetos há falta e necessidade de uma etapa de controle de qualidade e suas abordagens que forneça uma melhor e maior eficácia para que erros graves não estejam presentes na versão final do produto. Neste contexto, essas empresas tiveram que realizar trabalho extra e lidar com custos fora do orçamento, cronograma extrapolado, pós-produção para atender a demanda e necessidade de seus jogadores, insatisfação de seus funcionários, fazendo com que houvesse uma sensação geral de “whiplash” (efeito adverso às expectativas) nas suas respectivas comunidades e internamente no ambiente de trabalho.

Logo, busca-se uma maior clareza em torno desse processo tão fundamental, uma vez que diversos títulos lançados ultimamente apresentaram um resultado indesejável.

Como se trata de uma etapa importante do desenvolvimento e é dada sequência na pós-produção, deve-se levar em conta a sua necessidade dentro do projeto e seu impacto na qualidade do jogo, observando as técnicas de teste que melhor se adaptam ao contexto do produto, as limitações de recursos, a satisfação de usuários, entre outros pontos.

Como Objetivo Geral deste artigo, o intuito é apresentar o software preliminar de recomendação na gestão do controle de qualidade em jogos digitais, para que seja aplicável e adaptável em projetos que estejam desenvolvendo jogos digitais. O software de recomendação tem como fundamento disponibilizar as técnicas de teste adaptadas ao contexto da aplicação, descrevendo as características de cada uma, tendo em vista as entradas nas quais o usuário informa durante a execução da aplicação.

Como metodologia aplicada neste estudo, definiu-se inicialmente o protocolo para mapeamento sistemático na literatura, contendo as questões de pesquisa e *strings* de busca. Com isso, foram realizadas pesquisas nas bases de dados Google Scholar e SCOPUS para identificação de artigos acadêmicos. Uma vez selecionados, iniciou-se o processo da aplicação da técnica de *snowballing*, com o intuito de identificar novos estudos. Após a seleção dos estudos mais relevantes, investigou-se o material coletado com o objetivo de gerar uma lista com as práticas de controle de qualidade, além dos aspectos de qualidade. Após isso, o problema de estudo foi analisado e planejado, no qual se buscou desenvolver um protótipo de software para apoio à gestão de controle de qualidade em jogos digitais. Por fim, foi realizada uma pesquisa com profissionais da área de teste de software focada em jogos digitais para que houvesse a validação dos dados utilizados na solução preliminar, que consiste em uma aplicação baseada em árvores de decisão. Por fim, os resultados da pesquisa foram analisados e conclusões foram discutidas a cerca de todo o trabalho realizado até então, tendo em vista o software preliminar desenvolvido.

O seguinte artigo está organizado da seguinte forma: a Seção 2 aborda brevemente os trabalhos relacionados deste artigo. A Seção 3 trata de abordar o contexto conceitual da pesquisa. A Seção 4 apresenta como a pesquisa para validação das informações do software foi planejada, seu foco e análise dos resultados. A Seção 5 descreve o funcionamento do software preliminar de recomendação de técnicas de teste. Por fim, a Seção 6 apresenta as conclusões do estudo e trabalhos futuros.

2. Trabalhos Relacionados

Dos estudos selecionados, [Syed 2019], [Lahti et al. 2015], [Burnstein 2003] e [Schultz and Bryant 2016] tratam de definir o que é a etapa de controle de qualidade e sua relevância ao processo de desenvolvimento de jogos digitais. [Vargas et al. 2014] apresenta um mapeamento sistemático sobre qualidade em jogos sérios. [Lahti et al. 2015] apresenta e discute em detalhes as técnicas de controle de qualidade no desenvolvimento de jogos em jogos da Polônia. [Novak 2011] trata de abordar o ciclo de vida geral no desenvolvimento de jogos, técnicas de teste na etapa de controle de qualidade, papéis envolvidos no desenvolvimento de jogos digitais e a visão teórica dos possíveis gêneros de jogos.

Com os resultados encontrados, percebe-se a falta de um entendimento e aplicação de técnicas de teste na etapa de controle de qualidade, onde muitas vezes são ignoradas ou utilizadas por muito pouco tempo pelos desenvolvedores durante o processo de desenvolvimento, além da falta de discernimento sobre quais técnicas e processos utilizar. Além

disso, os trabalhos relacionados não apresentam e não buscam solucionar o problema de adaptação de técnicas de teste voltadas à determinado contexto, apenas realizando a discussão sobre a área como um todo.

3. Desenvolvimento

3.1. Controle de Qualidade

O Controle de Qualidade (CQ) se baseia no conceito de minimizar o número e impacto de *bugs*, falhas e outros comportamentos não desejáveis antes que o produto seja totalmente finalizado e disponibilizado aos usuários finais. CQ garante que cada produto de trabalho atenda aos requisitos impostos por meio de revisões, inspeções e especialmente, testes através do processo de desenvolvimento [Pressman 2005].

Tendo em vista a descrição de CQ, procedimentos, práticas e recursos podem ser estipulados para definir a sua utilização de forma mais ampla, o que incluem: políticas e padrões; programas de treinamento; programas de medição; processo de planejamento de testes; uso efetivo de técnicas e ferramentas de teste; monitoramento do processo e controle de sistema; sistema de gestão de configuração; prevenção de erros e detecção de erros [Burnstein 2003].

A etapa de Controle de Qualidade em jogos digitais é diferente para cada projeto e empresa, onde o processo em si geralmente está relacionado com as necessidade do usuário [Ruuska 2015]. Além de detectar e prevenir erros, os times de CQ proveem feedback inicial sobre o projeto, arte, interface do usuário, mecânicas internas, necessitando de profissionais capacitados e altamente técnicos [MCV 2019]. Apesar disso, existe uma baixa utilização de aspectos como reusabilidade de código, ferramentas entre projetos, artefatos de automatização, onde também é perceptível uma fragilidade às mudanças frequentes [Murphy-Hill et al. 2014].

3.2. Práticas Aplicáveis de Controle de Qualidade em Jogos Digitais

As técnicas de Controle de Qualidade variam de acordo com o sistema que está sendo testado e seu contexto. Técnicas bem conhecidas como teste de verificação, integração, regressão, *smoke*, *stress*, unidade e experiência do usuário tendem a ser as mais utilizadas na indústria [Bourque and Fairley 2014]. As principais técnicas de Controle de Qualidade encontradas na literatura são as seguintes: Teste Funcional, Teste Unitário, Teste de Integração, Teste de Progressão, Teste *Smoke*, Teste Exploratório, Teste *Cleanroom*, Teste A/B, Teste com Grupos Focais, *Playtesting*, Teste de Performance, Teste de Aceitação, Teste de Regressão e Teste Automatizado.

Teste Funcional, no nível de sistema, é utilizado para garantir que o comportamento do sistema é compatível com a especificação de requisitos [Burnstein 2003]. Em jogos digitais, testes funcionais servem para que o testador identifique qualquer problema com a jogabilidade que possa afetar a experiência do jogador, o que inclui performance, estabilidade, *features*, entre outros aspectos. [Syed 2019].

Teste Unitário é realizado pelos desenvolvedores para garantir que o comportamento desejado de uma seção específica da aplicação seja atendido, no qual possui ótimo desempenho para identificar problemas o mais cedo possível no ciclo de vida de desenvolvimento [Negrão 2020].

Teste de Integração expõe os defeitos das interações entre as unidades integradas realizadas previamente nos testes unitários [Negrão 2020]. Cada módulo testado anteriormente é adicionado um de cada vez, até o momento em que todos estão integrados. Isso evita que grandes falhas ocorram quando várias unidades são testadas ao mesmo tempo em conjunto e auxilia o testador a manter um ambiente de teste relativamente pequeno para análise [Burnstein 2003].

Teste de Progressão baseia-se no contexto de testar determinadas pequenas funcionalidades ou aspectos do jogo do início ao fim (missões, mapas, puzzles, etc.) [Syed 2019]. O objetivo é completar cenários de jogo em que haja qualquer tipo progresso, evitando encontrar erros que atrapalhem o mesmo, impossibilitando a continuidade da progressão do jogo ou que façam o jogo parar de funcionar repentinamente [Ramadan and Hendradjaya 2014].

Teste *Smoke* tem o objetivo de testar e revelar erros em um sistema já finalizado, verificando as funcionalidades mais básicas [Mozgovoy and Pyshkin 2018]. Dependendo do jogo, algumas rotinas podem ser automatizadas para testar funcionalidades básicas, mas que sejam executadas muitas vezes, como criar uma nova conta ou completar o tutorial [Syed 2019].

Teste Exploratório possui o foco de permitir aos testadores que utilizem a aplicação que está sendo testada da forma que quiserem, explorando o jogo sem nenhuma restrição, produzindo ao final casos de teste e documentação de teste [Lahti et al. 2015].

Teste *Cleanroom* possui como fundamento produzir testes no qual irão emular o comportamento típico de como os usuários jogam [Schultz and Bryant 2016]. O time de desenvolvimento investiga os diferentes tipos e modos de jogabilidade que seus jogadores possuem e capturam essas informações para encontrar erros onde uma abordagem típica de testes de funcionalidades não seria capaz de verificar [Lahti et al. 2015].

Teste A/B possui uma abordagem que gira em torno da divisão de usuários de um software em dois grupos com versões diferentes do sistema, no qual os usuários não podem testar ambas versões e ao final da realização dos testes, os resultados são comparados e avaliados entre as versões, de acordo com o feedback fornecido [Lahti et al. 2015].

Teste com Grupos Focais se baseiam no recrutamento de potenciais usuários para testar o jogo em uma sessão organizada pela equipe de desenvolvimento ou pelo time de CQ [Lahti et al. 2015]. Como vantagem, participantes podem mencionar e notar imperfeições que são esquecidas pelo time de desenvolvimento [Lahti et al. 2015].

Playtesting, considerada a técnica mais importante que um testador irá utilizar, é a atividade que a equipe de testadores e de projeto irão praticar durante toda a elaboração de sistema com o objetivo de ganhar discernimento em quais aspectos o jogo está ou não alcançando os objetivos de experiência do usuário [Fullerton 2014]. Métricas como eficiência, performance, dificuldade, níveis de aborrecimento e empolgação são analisadas pelos testadores em um documento detalhado, promovendo as informações mais importantes para que hajam melhorias no futuro, geralmente aplicando ferramentas de análise de dados para identificar o comportamento dos jogadores [Negrão 2020] [de Oliveira Neto et al. 2019].

Teste de Performance é utilizado para verificar se o software atende aos requisitos

de performance, onde a equipe de testadores deve verificar e otimizar a alocação de recursos do sistema [de Oliveira Neto et al. 2019]. Em jogos digitais, parâmetros importantes devem ser cuidadosamente levados em consideração, como tempo de resposta entre servidores e jogadores, tempo de transações, cobertura de rede, escalabilidade do sistema, restrições de memória e processadores, consumo de bateria e performance gráfica [Hamilton 2022].

Teste de Aceitação é um marco importante para os desenvolvedores, pois é nessa etapa que os jogadores irão determinar se o produto implementado está de acordo com seus requisitos [Burnstein 2003]. Alguns pontos devem ser atendidos previamente antes da execução dos testes de aceitação, como requisitos de negócio, código-fonte finalizado, atividades de CQ completadas, *bugs* e defeitos solucionados [Altexsoft 2021].

Teste de Regressão é realizado para verificar se a qualidade do jogo ainda permanece em um bom nível mesmo após várias mudanças terem sido feitas, adicionando funcionalidades novas ou alterações da estrutura interna [Aleem et al. 2016]. A necessidade deste tipo de teste vem do fato que antigos defeitos e erros podem ser reintroduzidos no código e gerar novos erros [Novak 2011].

Devido a complexidade de interações de usuários em jogos digitais ultimamente, testes em geral estão altamente dependentes de testadores humanos [Zheng et al. 2019]. Tendo isso em vista, com uma demanda maior por jogos mais sofisticados, testes automatizados se tornaram altamente adotados pelas empresas para reduzir os custos de produção e melhorar a qualidade interna [Bertolino 2007]. Como a automatização de testes pode ser considerada uma grande área de teste, englobando e automatizando outras técnicas já discutidas nesta seção, deve-se considerar aspectos como cronograma de desenvolvimento, criação de testes, tempo de execução para a escolha do melhor método de teste automatizado [Nilsson and Nilsson 2021].

4. Aplicação do Formulário de Pesquisa

Para a elaboração do escopo e planejamento da avaliação proposta, foi definida a escolha da característica de Usabilidade como foco central da pesquisa e busca da resolução do problema em questão. Este ponto trouxe vantagens, como foco em apenas uma linha de pesquisa, gerando uma análise mais precisa e com resultados simples, além de diminuir a complexidade do escopo, mas realizando uma análise mais precisa do produto de trabalho realizado.

Para a validação dos artefatos do estudo à serem utilizados no software preliminar, ou seja, as informações de cenários de teste manipuladas pelo sistema, foi elaborado um formulário de pesquisa no *Google Forms* para que pessoas com um perfil específico de participante colaborassem no estudo, sendo estes profissionais da área de teste de software e/ou controle de qualidade aplicada a jogos digitais, atuando tanto na indústria de jogos ou de forma acadêmica. O objetivo do participante foi de analisar as informações apresentadas em dez cenários elaborados contendo características distintas da etapa de testes e controle de qualidade durante o desenvolvimento de um jogo digital e, além disso, definir por meio de uma escala a sua opinião a respeito das técnicas de teste a serem utilizadas naquele contexto. A escala é conhecida como Escala Likert [Likert 1932]) e possui notas de 1 (discordo fortemente) a 5 (concordo fortemente). Além disso, o participante obrigatoriamente deveria expôr uma explicação do porquê da sua escolha.

Cada um dos dez cenários apresentam as mesmas características, porém, com diferentes valores associados a elas. Com isso, é possível criar diversos cenários tendo em vista associações entre as informações e realizando uma avaliação individual de cada uma. As características presentes e seus respectivos valores são:

- **Quantidade de Testadores:** Valores que variam entre menos que cinco, entre cinco e dez, entre dez e quinze e acima de vinte testadores.
- **Experiência em Testes:** Definição da experiência da equipe que realizará os testes. Pode variar entre uma predominância de testadores júniores, plenos, sêniores ou um equilíbrio entre os três.
- **Investimento Geral em Testes:** Valores monetários para investimento do cenário. Podem variar entre R\$ 1.000 e R\$ 5.000; R\$ 20.000 e R\$ 50.000, entre outras faixas de valores.
- **Área mais relevante para investimento:** O foco do teste a ser realizado. Atualizações e Correções, Certificação em Plataformas, Testes Internos e Testes com Usuários são possíveis focos.
- **Cronograma de Testes:** Estimativa do tempo disponível para testar o cenário. Pode variar entre uma e duas semanas, entre três a quatro semanas, e assim por diante.
- **Principais Gêneros:** Uma tupla de gêneros de jogos, como Ação e Aventura, Estratégia e Terror, Sobrevivência e Ação, entre outros.
- **Público Alvo:** Identificação da característica dos jogadores propensos a jogar esse estilo de jogo. Pode variar entre Jogadores Casuais, Jogadores *Hardcore* e Jogadores Sociais.
- **Plataforma:** Plataforma no qual o jogo será testado, como *PlayStation*, *XBox*, *PC* e *Nintendo Switch*.
- **Aspecto de Usabilidade a ser testado:** Aspectos como Conformidade, Operabilidade, Aprendibilidade, Eficiência e Flexibilidade.
- **Técnicas de Teste Recomendadas:** Listagem de três a quatro possíveis técnicas de teste recomendadas para avaliação.

Com as características definidas, inicia-se o processo de criação dos cenários. Tendo em vista cada uma das informações acima, busca-se realizar a relação entre elas da melhor forma possível, gerando interações que façam correlação com cenários próximos da realidade. Três exemplos de cenários utilizados na aplicação do formulário de pesquisa estão listados abaixo.

- **Cenário 1:** Entre 5 e 15; Equilíbrio entre Júnior, Pleno e Sênior; Entre R\$ 1.000 e R\$ 5.000; Testes internos; Entre uma semana e duas semanas; Ação e Aventura; Jogadores Casuais; *PC*; Aprendibilidade; **Técnicas Recomendadas:** Teste Unitário, Teste de Regressão e Teste Exploratório.
- **Cenário 2:** Mais de 20; Predominante: Sênior; Acima de R\$ 100.000; Certificação em plataformas; Mais que três meses; *RPG* e Ação; Jogadores *Hardcore*, *PlayStation*; Eficiência; **Técnicas Recomendadas:** Teste de Regressão, Teste de Aceitação, Teste de Performance e Teste de Funcionalidade.
- **Cenário 3:** Entre 15 e 20; Predominante: Pleno; Entre R\$ 21.000 e R\$ 50.000; Atualizações e correções; Entre um mês e dois meses; Estratégia e *RPG*; Jogadores de Jogos Sociais; *Nintendo Switch*; Conformidade; **Técnicas Recomendadas:** Teste de Aceitação, Teste Exploratório e Teste Automatizado.

4.1. Análise dos Resultados da Pesquisa

Considerando as respostas fornecidas pelos três participantes, nota-se uma não conformidade entre os mesmos em concordar com as técnicas de teste sugeridas para determinados cenários. Em alguns cenários apresentados, as notas fornecidas pelos entrevistados variaram de forma extrema, onde 33% avaliaram com nota 5, 33% com nota 3 e 33% com nota 1. A Figura 1 mostra a relação das avaliações de todos os cenários apresentados. As opiniões divergem em relação às técnicas específicas mais adequadas para cada contexto, ordem de aplicação e quantidade de testadores necessários.

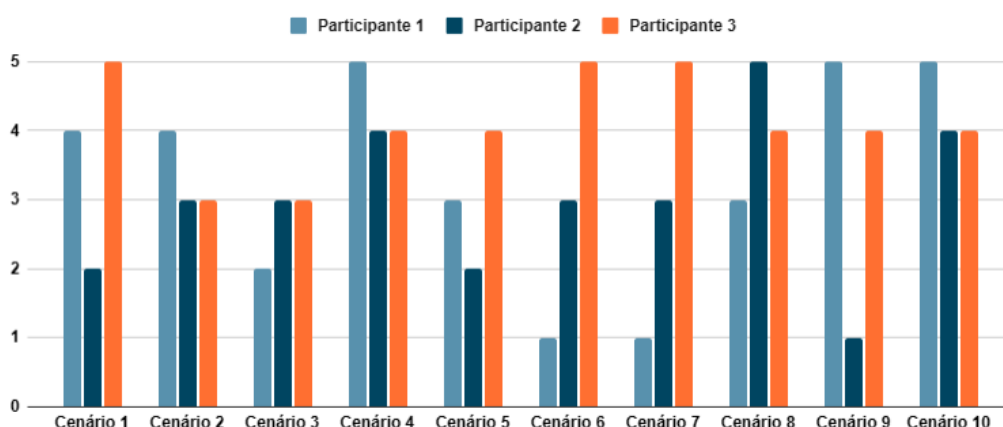


Figura 1. Visão geral das avaliações de todos os participantes

No geral, as opiniões convergem para a relevância de técnicas como testes de usabilidade, testes de aceitação, testes exploratórios e inspeções de conformidade. Essas técnicas são consideradas eficazes para garantir a conformidade com requisitos, identificar problemas de usabilidade, coletar *insights* valiosos dos usuários e melhorar a experiência geral do jogo.

Os cenários 1, 4, 8 e 10 receberam as notas mais altas, indicando que foram considerados pelos entrevistados como os cenários mais adequados em termos de técnicas de avaliação propostas, apresentando concordância geral dos três participantes com as técnicas de teste recomendadas. Já os cenários 3, 5, 6 e 7 receberam as notas mais baixas, indicando que foram considerados pelos entrevistados como os cenários menos adequados, onde mudanças são recomendadas.

Por fim, houveram casos onde os participantes propuseram novas sugestões de técnicas de teste relevantes, como por exemplo: percurso cognitivo, inspeções de conformidade, avaliações de usabilidade e técnicas formais de tempo de reação. Com esses dados analisados após a realização da pesquisa, as informações foram inseridas no sistema de recomendação, uma vez que foram validadas, para servir de apoio na execução do software.

5. Desenvolvimento do Software Preliminar

Como solução, foi realizada a elaboração de um software utilizando-se do conceito de árvores de decisão, uma vez que o escopo do problema pode ser resolvido com algoritmos de classificação, sendo visualmente fácil de entender a sua estrutura e conceitos

fundamentais. O objetivo é auxiliar a equipe de desenvolvimento a compreender quais técnicas de teste são mais adequadas de acordo com o seu contexto, tendo em vista características como cronograma, quantidade de testadores, área de teste mais relevante, entre outros aspectos.

O sistema possui as seguintes entradas, baseadas no formulário de pesquisa previamente realizado, fornecidas pelo usuário: Quantidade de testadores; Nível de experiência da equipe; Valor a ser investido no cenário de teste; Área mais relevante para investimento; Estimativa de duração das tarefas de teste; Principais gêneros que se enquadram no jogo; Público alvo; Plataforma no qual o jogo será publicado e, por fim, o aspecto de Usabilidade a ser testado no cenário.

A análise das entradas pelo sistema se constitui na utilização de um algoritmo de classificação baseado em árvores de decisão. Cada entrada apresentada acima é considerada uma *feature* que está armazenada dentro de um *dataset*, ou seja, os dados de treinamento para a geração da possível árvore de classificação. Os dados presentes no *dataset*, neste caso, os possíveis cenários e suas informações, são provenientes da validação da pesquisa com os profissionais da área de teste de software e/ou controle de qualidade.

Para cada interação, o algoritmo considera a partição do conjunto de dados de treinamento usando a saída de uma função dos atributos de entrada. A seleção da função mais apropriada é feita de acordo com algumas medidas de divisão. Ou seja, o ganho de informação é usado para decidir em qual *feature* haja a divisão em cada passo ao construir a árvore. Para cada nó da árvore, o valor da informação mede quanta informação uma *feature* fornece sobre uma classe (neste caso, o resultado final da execução do software, ou seja, a classe de Técnicas Recomendadas). A divisão com o maior ganho de informação será selecionada como a primeira partição e o processo irá continuar até que todos os nós estejam puros, ou seja, em uma amostragem do *dataset*, toda a informação pertence a mesma classe, ou até que o ganho de informação seja igual a zero.

Por fim, a saída da solução preliminar proposta é a classe da listagem das técnicas de teste mais adaptadas para o contexto fornecido pelo usuário, onde uma descrição de cada técnica é exibida para que haja entendimento da vantagem de sua aplicação em um cenário real.

6. Conclusão

Considerando os resultados da aplicação do formulário com profissionais das áreas de teste de software e/ou controle de qualidade em jogos digitais, percebe-se que não há uma concordância geral sobre quais técnicas de teste ou de inspeções devem ser utilizadas dado um cenário específico de desenvolvimento. Isto ocorre por uma série de razões, justamente por causa do cenário atual da indústria de jogos ser amplo, onde cada projeto possui inúmeras características distintas entre si, no qual a realidade interna varia entre diferentes empresas. Além disso, a experiência e conhecimento técnico de cada profissional na área colabora na variação da estimativa sobre o que é melhor para um determinado cenário e o que não é agradável ou possível de se utilizar.

Muitas das respostas coletadas se basearam em experiências prévias, sejam aplicadas em casos reais ou meramente por dominarem um conhecimento específico, o que fornece uma noção do que realmente pode ser aplicado em outros cenários que compartilham de determinadas características. Tendo isso em vista, é difícil analisar um cenário

de testes em um jogo digital, onde não existe uma solução definitiva que solucione com altíssima acurácia o problema sobre quais técnicas de teste se devem utilizar, apenas fornecendo uma avaliação que mais se assemelha à realidade do projeto.

A aplicação desenvolvida possui o intuito de auxiliar a equipe de desenvolvimento a entender quais técnicas de teste são mais adequadas ao seu contexto. A validação das informações apresentadas ao usuário no relatório proveniente da execução do software foram validadas por meio das respostas coletadas do formulário com profissionais. Com essas informações, o desenvolvedor tem a possibilidade de ver a descrição das técnicas de teste e aplicá-las em seu projeto, fazendo assim que os recursos possam ser alocados de melhor forma e o fluxo de teste seja otimizado como um todo. Apesar do software de recomendação estar na etapa de protótipo, o mesmo atende ao objetivo inicial e condiz com o que foi planejado.

Como trabalhos futuros, busca-se melhorias gerais no software e em suas funcionalidades. Um ponto a ser explorado gira em torno da questão do sistema ajudar a documentar e rastrear o processo de teste, permitindo que a equipe de desenvolvimento acompanhe e avalie o progresso em relação as técnicas de teste recomendadas. Outro ponto a ser analisado é a realização de entrevistas com mais profissionais da área de controle de qualidade em jogos, com o objetivo de obter perspectivas diferentes e maior validação ao estudo. Além disso, um *dataset* maior e com mais casos relevantes para treinamento da árvore de decisão deve ser explorado, uma vez que isso pode abranger mais casos e cenários contendo diferentes características do desenvolvimento de jogos digitais.

Referências

- Alem, S., Capretz, L. F., and Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4(1):1–30.
- Altexsoft (2021). How to conduct user acceptance testing: Process stages, deliverables, and end-user testing place in quality assurance. Acesso em: jan. 2023.
- Bertolino, A. (2007). Software testing research: Achievements, challenges, dreams. In *Future of Software Engineering (FOSE'07)*, pages 85–103. IEEE.
- Bourque, P. and Fairley, R. E. (2014). Swebok, version 3.0: Guide to the software engineering body of knowledge. *Piscataway (NJ, USA): IEEE*.
- Burnstein, I. (2003). Practical software testing: A process-oriented approach. In *Springer Science & Business Media*.
- de Oliveira Neto, J. N., Viana, D., Sá, E., Rivero, L., Lopes, R. F., and Silva, F. (2019). Is there time for software testing in the indie games development? a survey with practitioners of the game industry. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 37–46.
- Fullerton, T. (2014). *Game design workshop: a playcentric approach to creating innovative games*. CRC press.
- Hamilton, T. (2022). Game testing: Types how to test mobile/desktop apps. Acesso em: jan. 2023.
- Lahti, M. et al. (2015). Game testing in finnish game companies. Master's thesis.

- Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.
- MCV (2019). Quality control: Understanding your qa team can only improve your game. Acesso em: dez. 2022.
- Mozgovoy, M. and Pyshkin, E. (2018). A comprehensive approach to quality assurance in a mobile game project. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*, pages 1–8.
- Murphy-Hill, E., Zimmermann, T., and Nagappan, N. (2014). Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? In *Proceedings of the 36th International Conference on Software Engineering*, pages 1–11.
- Negrão, P. M. (2020). *Automated Playtesting In Videogames*. PhD thesis.
- Nilsson, F. and Nilsson, J. (2021). Comparing automated testing approaches for fps games.
- Novak, J. (2011). *Game development essentials: an introduction*. Cengage Learning.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
- Ramadan, R. and Hendradjaya, B. (2014). Development of game testing method for measuring game quality. In *2014 International Conference on Data and Software Engineering (ICODSE)*, pages 1–6. IEEE.
- Ruuska, E. (2015). Quality assurance testing on video games: The importance and impact of a misunderstood industry.
- Schultz, C. P. and Bryant, R. D. (2016). *Game testing: All in one*. Mercury Learning and Information.
- Syed, M. (2019). Quality control in games development.
- Vargas, J. A., García-Mundo, L., Genero, M., and Piattini, M. (2014). A systematic mapping study on serious game quality. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, pages 1–10.
- Zheng, Y., Xie, X., Su, T., Ma, L., Hao, J., Meng, Z., Liu, Y., Shen, R., Chen, Y., and Fan, C. (2019). Wuji: Automatic online combat game testing using evolutionary deep reinforcement learning. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 772–784. IEEE.