

UNIVERSIDADE FEDERAL DE MINAS GERAIS DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Revisão de Literatura
Cinzenta:
Testes Automatizados de
Jogos Digitais
Pesquisa Tecnológica

Aluno: Leandro Diniz Silva
Orientador: André Cavalcante Hora

Belo Horizonte, 2025

Resumo

Dentro da engenharia de *software* existem diversas técnicas e padrões que podem ser implementados em um projeto para melhorar sua qualidade e robustez, como testes de *software* automatizados. Mas este recurso não é amplamente utilizado em todas os tipos de programas e o desenvolvimento de jogos é exatamente um deles. Essa falta de aderência é causada por diversos motivos, como a complexidade das interfaces dos jogos e a aleatoriedade presente nas interações do usuário. Este trabalho investiga essas questões por meio de referências na literatura cinzenta, revisando técnicas e discussões disponibilizadas em diferentes formatos. A pesquisa busca compreender o impacto de testes automatizados no desenvolvimento de jogos, bem como seus desafios técnicos e culturais.

1 Introdução

As ferramentas de teste de *software* automatizadas têm aplicabilidade em uma ampla gama de sistemas digitais, contando com suporte de uma variedade de bibliotecas e ferramentas que auxiliam na produção e visualização das verificações feitas no programa. Jogos eletrônicos, mesmo que produzidos em diversas dimensões, além da sua linguagem de programação, também são *softwares* e podem conter técnicas de testes automatizados.

De acordo com Arora, a indústria de jogos tem um alcance global sem precedentes, tendo um valor aproximado de \$184.4 bilhões de dólares [2]. E mesmo uma indústria desse tamanho encontra problemas significativos, que resultam na entrega de produtos em um estado que aparenta estar inacabado e repleto de *bugs* indesejados [10].

Em meio a tal contexto, foi primeiramente realizado um projeto em que diversos repositórios *open-source* foram explorados com o objetivo de encontrar quais ferramentas, bibliotecas, *frameworks* e o nível de adesão de técnicas relacionadas a testes automatizados no desenvolvimento de jogos.

Essa pesquisa encontrou que no ambiente *open-source* explorado não existe uma grande adoção de testes automatizados, onde até mesmo os projetos que continham algum tipo de automatização não a utilizavam em todo o sistema, escolhendo somente algumas partes do projeto que causavam problemas constantes para seus usuários e não para aumentar a confiabilidade e robustez do código escrito.

A partir dessa análise inicial, a pesquisa atual foi estruturada com o objetivo de expandir os resultados obtidos, ampliando o conhecimento de técnicas de testes automatizados no desenvolvimento de jogos. Além disso, buscam também compreender as dificuldades encontradas e as soluções desenvolvidas para elas. Realizar uma revisão de literatura cinzenta sobre esse tema pode nos mostrar *insights* valiosos sobre como indústria de jogos lida com esse paradigma.

O restante do trabalho está organizado como segue. Na Seção 2 é apresentado o Referencial Teórico. Na Seção 3 é descrita a Metodologia de Desenvolvimento empregada para a realização do trabalho. Na Seção 4 é descrito os Resultados encontrados e na última Seção 5 são feitas as Considerações Finais.

2 Referencial

Para se desenvolver esse projeto foi necessário estudar e pesquisar por trabalhos relacionados a esse tema, com o objetivo de entender como os desenvolvedores de jogos trabalham com o desenvolvimento de testes automatizados. Outro ponto de muita importância foi a pesquisa de como estruturar um projeto que utiliza de literatura cinzenta, especialmente quando aplicada a engenharia de software. Este referencial teórico busca explorar as práticas adotadas na criação de testes automatizados em jogos e como a literatura cinzenta pode ser utilizada de forma eficaz a pesquisas.

2.1 Literatura Cinzenta

De acordo com Wenger [15], uma comunidade de prática é composta por pessoas que se reúnem para compartilhar conhecimentos e aprenderem juntas, com o objetivo de aprimorar suas habilidades em um interesse comum. Com o crescimento da internet, essas comunidades passaram a se expandir para redes sociais e fóruns, como o *StackOverflow* e o *GitHub* [13]. Isso demonstra que muitas dessas discussões podem ser encontradas e analisadas, desde que sejam organizadas com uma estrutura adequada.

Essas fontes podem ser classificadas como literatura cinzenta que de acordo com Adams et al. [1] ela é composta por artefatos que não são produtos de processos de revisão por pares que caracterizam publicações acadêmicas. Esses textos podem ser classificados em níveis de acordo com sua confiabilidade [1], um exemplo dessa especificação pode ser visto na figura 1.

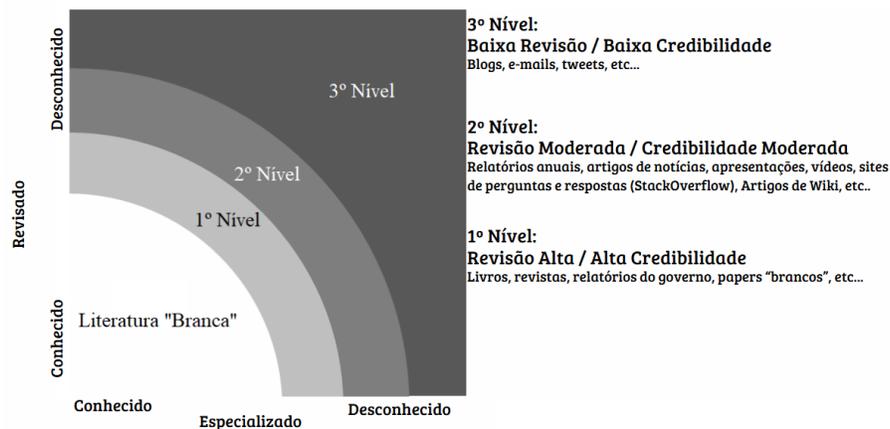


Figura 1: “Sombras” das literaturas cinzentas (Original de [1])

No entanto, seu uso também apresenta desafios, como a necessidade de validar a credibilidade das fontes e garantir que os dados coletados sejam relevantes

e precisos. Para superar esses problemas, existem diretrizes específicas que orientam o uso desse tipo de material em pesquisas. Um exemplo é a abordagem desenvolvida por Garousi et al. [3], criada especificamente para projetos na área de engenharia de software, oferecendo métodos estruturados para identificar, selecionar e avaliar fontes de literatura cinzenta de forma criteriosa e eficiente.

2.2 Diferenças no Desenvolvimento de Jogos

De acordo com Whittaker [16] o principal objetivo de um testador é simular as interações entre o usuário e o programa, no entanto, em jogos essas interações não são tão diretas quanto em um *software* comum, pois um jogo tem como principal objetivo entreter e divertir seu consumidor [8][11] e não necessariamente entregar um serviço. Isso cria desafios únicos para o desenvolvimento de testes automatizados.

No mundo de desenvolvimento de jogos, a maior parte dos testes são feitos por um time de profissionais de controle de qualidade [11], também chamado de *QA*, esses especialistas acessam o software por um tipo de *black box* e simulam as interações dos usuários para tentar encontrar problemas. Esse tipo de testes tem suas vantagens e é considerado muito importante [6], como conseguir testar o "fator diversão" de um jogo. Mas existem desvantagens como o tipo de trabalho repetitivo para os funcionários e a probabilidade de erros humanos [7].

Uma outra particularidade encontrada nos *video games* é o alto nível de aleatoriedade [9][10]. Esta aleatoriedade, que é essencial para certas características dos jogos, pode ser causada por diversos fatores, como o uso de *multithreading*, a inteligência artificial dos personagens e as ações não determinísticas incorporada em certos sistemas do jogo [8].

Existe uma dinamicidade inerente nas interfaces de jogos, além do seu grande nível acoplação com o código [9], o que cria diversos obstáculos ao se tentar automatizar testes de integração. Alguns desenvolvedores tentam usar técnicas de *template matching*, mas pela constante mudança de telas e a quantidade de detalhes pode-se gerar testes não determinísticos, ou seja testes *flakys* [7].

2.3 Testes Automatizados

Para facilitar a criação de testes automatizados, alguns pesquisadores criaram suas próprias ferramentas especializadas. Um desses recursos encontrados [5] utiliza de uma máquina de estados que divide o sistema em duas partes, em domínio, todas as interações da tela que não acontecem por interação dos jogadores, como o início do jogo ou a introdução de um novo obstáculo na tela, e em avatar, onde são encontradas todas as modificações que acontecem no mundo causadas por ações de um *player*, por exemplo o personagem controlado pular e destruir algum bloco.

Outro meio de testar é utilizar de ferramentas já existentes nas *engines* de criação de jogos. A Unity oferece um *framework* de testes chamado *Unity Test Framework* (UTF) e para facilitar a automação dos testes consegue dividir os mesmos em dois tipos *Edit Mode* e *Play Mode* [9]. O primeiro modo são

testes que podem ser realizados sem a necessidade de iniciar o jogo, e o segundo somente quando o *software* estiver aberto.

Criar e utilizar essas ferramentas pode ser considerado como um custo de desenvolvimento, mas esses custos normalmente são negligenciados pois os programadores de jogos sabem que existe uma grande chance de seu código ser jogado fora caso alguma mudança no design aconteça [8]. Além disso, existe o problema da duplicação de código, onde pode existir diversas funções que tem o mesmo objetivo mas que são escritas com um nível de otimização diferente para cada caso, com o objetivo de melhorar a performance do jogo.

3 Metodologia

A principal abordagem utilizada nessa pesquisa foi a a revisão de literatura cinzenta, onde as referências e textos utilizados serão encontrados em locais como blogs, *issues* de repositórios no *GitHub* e discussões em fóruns *online* complementados por literaturas formais. Esse tipo de abordagem foi escolhida para tentar explorar como esse conteúdo é compartilhado por profissionais da área e como essas discussões refletem os desafios enfrentados.

3.1 Objetivos

Para estudar esse tipo de literatura não acadêmica, é fundamental estabelecer objetivos alinhadas a essa abordagem. O principal propósito é compreender por que há uma baixa adoção de técnicas de testes automatizados na indústria de desenvolvimento de jogos e identificar as dificuldades enfrentadas nesse processo que contribuem para essa resistência.

Outro ponto a ser estudado também é o porque um programador vê a necessidade, ou falta dela, de testar o código que ele esta desenvolvendo para um jogo quando se tem uma equipe de garantia de qualidade (*QA*) que deveria testar essas interações manualmente para o software que esta sendo produzido.

3.2 Questões de Pesquisa

O uso de questões de pesquisa é útil em estudos que utilizam literatura cinzenta, pois elas servem como um guia estruturado para explorar essas fontes não convencionais e garantir a relevância e a coerência dos dados coletados. Nesse contexto, questões de pesquisa bem definidas ajudam a filtrar informações úteis e direcionar a pesquisa para informações alinhadas aos objetivos do projeto, evitando desvios em um vasto e heterogêneo volume de fontes.

Além disso, ao trabalhar com literatura cinzenta, questões de pesquisa permitem uma análise mais crítica e estruturada do conteúdo, promovendo um papel de bússola intelectual facilitando a triangulação de informações entre diferentes tipos de fontes. Levando em consideração os objetivos dessa pesquisa foram criadas questões de pesquisa para o seu direcionamento, essas são:

- **Questão - 1:** Quais desafios técnicos dificultam a implementação de testes automatizados?
- **Questão - 2:** Quais desafios culturais dificultam a implementação de testes automatizados?
- **Questão - 3:** Como a presença de equipes de garantia de qualidade (QA) influencia a decisão de programadores em adotar ou não testes automatizados no desenvolvimento de jogos?
- **Questão - 4:** Qual é o impacto da falta de testes automatizados na qualidade e confiabilidade dos jogos lançados no mercado?

3.3 Atividades

A principal ferramenta utilizada para encontrar o material a ser revisado foi o mecanismo de pesquisa do *Google*. Ao utilizar dessa ferramenta foi necessário definir os parâmetros de busca para encontrar páginas relacionadas ao tema da pesquisa. As buscas feitas podem ser vistas abaixo:

- Video game automated testing
- Automated tests on video game development
- Automated testing on the AAA video game industry

Dado que as buscas no *Google* frequentemente retornam centenas de milhares de resultados, foi estabelecida uma limitação para revisar apenas as primeiras quatro páginas, aproximadamente os primeiros 35 links. Todos os links nessas páginas foram analisados para verificar se o conteúdo apresentado estava relacionado ao tema pesquisado.

Quando as páginas encontradas eram fóruns que resultavam em debates entre seus leitores, todos os comentários eram analisados e caso recomendassem leituras externas para complementar uma ideia, esses materiais também eram considerados e estudados.

A literatura cinzenta apresenta limites na veracidade de seus fatos, pois eles não passam por uma revisão de pares como outros estudos acadêmicos. Então para selecionar dados e discussões com um certo fator de autenticidade, entre as milhares de possíveis fontes, foi utilizado das diretrizes modeladas por Garousi et al [3].

Essa metodologia se divide em 20 perguntas a serem feitas com o objetivo de se definir se um material cinzento deveria ser utilizado. Alguns desses critérios utilizados são:

- A organização da publicação é respeitável?
- É um autor individual associado a uma organização respeitável?
- O autor publicou outros trabalhos na área?

- O autor tem expertise na área?
- O texto enriquece ou acrescenta algo único para a pesquisa?

Seguindo esses critérios, todos os textos foram analisados e classificados de acordo com as questões que eles respondem. Especial atenção foi dada ao último ponto, avaliando se os textos também abordam alguma das perguntas de pesquisa.

4 Resultados

No total foram analisados cerca de 90 sites nessa pesquisa, mas seguindo a metodologia definida na seção 3 somente 10 deles foram selecionados. Isso aconteceu porque muitos dos resultados não alcançavam o padrão da diretriz utilizada, muitos deles não tinham nenhuma informação nova ou confiável, parte deles era venda de produtos e serviços relacionados a testes e outros não eram relacionados em nenhum nível ao tema. Alguns desses resultados foram também artigos formais, todos esses foram estudados e os que mais agregaram a pesquisa foram utilizadas como referencial teórico.

A Tabela 1 apresenta um resumo das conexões entre os textos analisados e as questões de pesquisa. Um aspecto recorrente identificado na maioria dos textos foi a presença de desafios técnicos na implementação dos testes automatizados. Além disso, todas as questões de pesquisa contaram com, no mínimo, quatro leituras relacionadas, o que contribuiu significativamente para uma compreensão mais aprofundada do tema.

QP	Descrição	Leituras
1	Descreveram desafios técnicos na implementação de testes automatizados.	L1, L2, L4, L5, L6, L7, L8
2	Descreveram desafios culturais encontrados no ambiente de desenvolvimento.	L1, L2, L4, L8
3	Envolveram a equipe de QA com os testes automatizados.	L4, L5, L7, L9
4	Mostraram resultados obtidos com os testes.	L3, L4, L8, L9

Tabela 1: Leituras e suas contribuições às questões de pesquisa.

4.1 Desafios Técnicos (QP1)

Essa pergunta de pesquisa foi a que gerou o maior volume de resultados entre todos os textos analisados. Os materiais revisados destacam diversos problemas, incluindo a fragilidade dos códigos devido a mudanças constantes, a necessidade de escrever códigos que sejam testáveis, a presença de partes não determinísticas no código, o tempo elevado de execução dos testes e a ausência de ferramentas adequadas.

Algumas das equipes que implementaram testes automatizados precisaram criar ferramentas que atendiam suas necessidades específicas. O desenvolvedor da leitura 5 criou todo um *framework* de testes que se conectava a diversos tipos de máquinas, consoles, computadores e kits de desenvolvimento de console, para realizar múltiplos tipos de testes, além de capturar dados de performance juntamente com *screenshots* para ajudar a encontrar em quais partes do sistema os gargalos de performance aconteciam, facilitando encontrar pontos a serem otimizados.

No texto 4, a equipe optou por realizar diversas modificações no *framework* de testes da *Unreal Engine* para otimizar o tempo de execução dos testes. No uso padrão, o processo de renderização e criação dos ambientes de teste dentro do jogo consumiam muito tempo, impactando negativamente na experiência dos desenvolvedores.

4.2 Desafios Culturais (QP2)

Os desafios culturais foram identificados principalmente a partir de relatos de profissionais que atuam em empresas que utilizam ou tentaram implementar testes automatizados. Entre os problemas mencionados, um dos principais foi a falta de disciplina das equipes, que muitas vezes consideraram o tempo dedicado à criação de testes automatizados como um esforço desnecessário, preferindo focar no desenvolvimento de funcionalidades principais do código.

Além disso, a resistência à adoção dessas técnicas também foi evidenciada. Na leitura 8, por exemplo, um desenvolvedor criou, em seu tempo livre, uma ferramenta para testes de integração e E2E diretamente no jogo. Essa ferramenta utilizava os recursos do sistema, gerava saídas de dados integráveis ao pipeline de CI/CD e ao sistema de QA da equipe, mesmo com todo o potencial e funcionalidade da ferramenta, levou cerca de um ano para que a equipe compreendesse seu valor e começasse a utilizá-la de forma eficiente.

4.3 Envolvimento da equipe de QA (Q3)

Empresas que já utilizavam de testes automatizados combinavam sua aplicação, que verificam de forma mais rápida e precisa as partes mais lógicas do sistema, com testes manuais feitos pela equipe de controle de qualidade, que podem fazer testes em outras dimensões do sistema como por exemplo o fator diversão ou como uma música se encaixa no momento atual, esses testes mais exploratórios e de experiências. Em casos como o da leitura 9, 50% dos erros críticos eram encontrados pelos testes manuais dos *tester* e os outros 50% pelos automatizados.

Essas equipes também realizavam testes direcionados com base nos resultados dos testes automatizados de desempenho. Esses testes automatizados eram eficazes para identificar as áreas do sistema que causavam quedas significativas de desempenho e quantificar a seriedade desses problemas. No entanto, como eles não esclareciam as causas específicas das falhas, somente onde aconteceu e a gravidade, as equipes de qualidade realizavam uma segunda camada de testes

manuais para investigar detalhadamente os motivos por trás dessas quedas e determinar as ações corretivas necessárias.

4.4 Resultados positivos dos testes (QP4)

Quando aplicados corretamente, os testes automatizados produziram resultados positivos em alguns projetos. O benefício mais frequentemente destacado foi a facilidade em detectar problemas de regressão, permitindo que os desenvolvedores tivessem mais confiança em suas implementações. Essa confiança possibilitou a realização de alterações mais frequentes e significativas no código, sem o receio de introduzir novos problemas ou comprometer funcionalidades já existentes.

Nas leituras 4 e 9, a implementação de testes automatizados também resultou em uma redução considerável no tempo de *deployment*. No caso específico do jogo mencionado na leitura 4, o processo de lançamento de uma nova versão, que antes levava cerca de 10 dias, foi reduzido para apenas 1,5 dias. Essa melhoria representou um avanço significativo para a equipe, permitindo entregas mais rápidas, com maior qualidade e confiabilidade.

5 Conclusão

Jogos digitais são *softwares* tão como qualquer outro aplicativo, mas diversos problemas podem ser encontrados ao se aplicar técnicas de testes automatizados, dificuldades estas que não são encontrados frequentemente em sistemas de serviços. Isso reflete a complexidade inerente ao desenvolvimento de jogos, onde aspectos técnicos e criativos frequentemente se entrelaçam, exigindo soluções específicas e inovadoras.

Nos textos analisados, frequentemente os desenvolvedores precisaram criar ferramentas específicas aos desafios encontrados no desenvolvimento de seus projetos. Mostrando que existe uma falta de ferramentas abrangentes no mercado, mas que aos programadores que reconhecem os benefícios dos testes automatizados estão dispostos a ir além de seus escopos de trabalho, dedicando esforços para aumentar a confiabilidade de todo o sistema que estão trabalhando.

Sobre a cultura de testes automatizados em jogos digitais é possível observar que nem todos os projetos são iguais e que exigem o desenvolvimento dos testes, mas as equipes que fazem uso tem sim um retorno e quem quer começar utilizar precisa pensar bem em como vender a ideia para toda uma equipe que pode não concordar com isso.

Em conclusão, este trabalho mostra a importância de compreender as particularidades do desenvolvimento de jogos digitais e as implicações dos testes automatizados nesse contexto. Embora hajam desafios técnicos e culturais significativos, os benefícios potenciais, como maior confiabilidade e eficiência no desenvolvimento, destacam-se como incentivos valiosos.

6 Referências Bibliográficas

- [1] Richard J Adams, Palie Smart, and Anne Sigismund Huff. Shades of grey: guidelines for working with the grey literature in systematic reviews for management and organizational studies. *International journal of management reviews*, 19(4):432–454, 2017.
- [2] Krishan Arora. The gaming industry: A behemoth with unprecedented global reach. *Forbes*, November 17 2023.
- [3] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology*, 106:101–121, 2019.
- [4] GDC. About gdc, <https://gdconf.com/about-gdc>, Acesso em: 2024.
- [5] Sidra Iftikhar, Muhammad Zohaib Iqbal, Muhammad Uzair Khan, and Wardah Mahmood. An automated model based testing approach for platform games. In *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 426–435. IEEE, 2015.
- [6] Jussi Kasurinen, Maria Palacin-Silva, and Erno Vanhala. What concerns game developers? a study on game development processes, sustainability and metrics. In *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)*, pages 15–21, 2017.
- [7] Gabriel Lovreto, Andre T Endo, Paulo Nardi, and Vinicius HS Durelli. Automated tests for mobile games: An experience report. In *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SB-Games)*, pages 48–488. IEEE, 2018.
- [8] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. 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, 2014.
- [9] Nathalya Stefhany Pereira, Phyllipe Lima, Eduardo Guerra, and Paulo Meirelles. Towards automated playtesting in game development. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 349–353. SBC, 2021.
- [10] Cristiano Politowski, Yann-Gaël Guéhéneuc, and Fabio Petrillo. Towards automated video game testing: Still a long way to go. In *Proceedings of the 6th international ICSE workshop on games and software engineering: engineering fun, inspiration, and motivation*, pages 37–43, 2022.

- [11] Cristiano Politowski, Fabio Petrillo, and Yann-Gaël Guéhéneuc. A survey of video game testing. In *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*, pages 90–99, 2021.
- [12] Davide Spadini, Maurício Aniche, and Alberto Bacchelli. PyDriller: Python framework for mining software repositories. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2018*, pages 908–911, New York, New York, USA, 2018. ACM Press.
- [13] Margaret-Anne Storey, Leif Singer, Brendan Cleary, Fernando Figueira Filho, and Alexey Zagalsky. The (r) evolution of social media in software engineering. *Future of software engineering proceedings*, pages 100–116, 2014.
- [14] Marco Tulio Valente. Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, 1, 2020.
- [15] E Wenger. Communities of practice: The organizational frontier. *Harvard Business Review*, 2000.
- [16] James A Whittaker. What is software testing? and why is it so hard? *IEEE software*, 17(1):70–79, 2000.