
**COMO A MÁ IMPLEMENTAÇÃO DO SCRUM AFETA NEGATIVAMENTE O
DESENVOLVIMENTO DE JOGOS**

**HOW THE POOR IMPLEMENTATION OF SCRUM NEGATIVELY AFFECTS
VIDEOGAME DEVELOPMENT**

Leone Branquinho Raffo¹

Igor da Silva Elias²

RESUMO

Com base na crescente demanda da indústria de jogos por metodologias ágeis para o desenvolvimento de software, o framework Scrum se encontra como uma boa alternativa dentro dessa área. Este artigo de revisão literária tem como objetivo aprofundar o conhecimento sobre as más práticas de implementação dessa metodologia nas equipes de desenvolvimento e suas consequências. Além disso, busca-se analisar como este framework é implementado atualmente, identificar quais são as más práticas e como elas afetam o sucesso de um jogo. A metodologia utilizada nesta revisão consiste em pesquisas bibliográficas e a análise de estudos relevantes da área. Espera-se que os resultados desta pesquisa possam contribuir para melhorar o processo de desenvolvimento de jogos, fornecendo uma visão mais clara e precisa sobre os desafios relacionados à implementação dessa metodologia ágil em equipes de desenvolvimento. As principais conclusões desta revisão literária foram apresentadas neste estudo, evidenciando a importância de uma implementação adequada do Scrum para o sucesso no desenvolvimento de jogos.

392

Palavras-chave: scrum; desenvolvimento de jogos; implementação inadequada; metodologia ágil; más práticas.

ABSTRACT

Based on the increasing demand for agile methodologies in the video game industry, the Scrum framework finds itself as a good alternative in this area. This literature review article aims to delve into the bad practices of this methodology in development teams and its consequences. The article analyzes how this framework is currently being implemented, identifies the common bad practices, and assesses how they affect the success of a game. The research aims to contribute to improving video game development by providing a clearer and more precise understanding of the challenges related to the implementation of this agile methodology on development teams. The paper presents the main conclusions, highlighting the importance of adequate Scrum implementation for the success of video game development.

Keywords: scrum; videogame development; adequate implementation; agile methodology; bad practices.

¹ Centro Universitário Filadélfia de Londrina - UniFil

² Centro Universitário Filadélfia de Londrina - UniFil

INTRODUÇÃO

O desenvolvimento de jogos é um processo complexo que envolve diversas etapas, desde a concepção da ideia até o lançamento do produto final, conforme constatado por Politowski e Petrillo (2021). A adoção de metodologias ágeis, como o Scrum, tem se mostrado cada vez mais comum na indústria de jogos. Esse tipo de abordagem visa otimizar o processo de desenvolvimento e garantir a entrega de produtos de qualidade dentro do prazo estipulado (Ruonala *et al.*, 2017).

No entanto, a implementação da metodologia nem sempre é bem-sucedida e pode acarretar consequências negativas para o processo de desenvolvimento de jogos. Kasurinen (2016) cita em um de seus estudos que a aplicação inadequada do Scrum pode resultar em atrasos no desenvolvimento, falta de clareza sobre as tarefas a serem realizadas e, conseqüentemente, na entrega de produtos finais de qualidade inferior.

Diante desse contexto, o objetivo desta pesquisa é demonstrar como a implementação inadequada do Scrum afeta negativamente o desenvolvimento de jogos. Para atingir esse objetivo, primeiramente serão identificados os principais desafios enfrentados pelas equipes de desenvolvimento de jogos ao implementar o Scrum, descobrir quais são as principais causas de uma má implementação da metodologia e seus impactos na qualidade do produto final.

A motivação para a realização desta pesquisa é contribuir para a melhoria do processo de desenvolvimento de jogos, identificando os problemas que podem surgir com a implementação inadequada do Scrum, demonstrar que metodologias ágeis não agregam valor se não forem implementadas corretamente e guiar equipes de desenvolvimento na implementação do framework abordado.

Fundamentação teórica

A fundamentação teórica é uma parte essencial de um artigo de revisão literária, pois apresenta os principais conceitos, autores e abordagens que embasam o tema de pesquisa. Neste trabalho, serão abordados dois escopos principais: Scrum e Desenvolvimento de jogos.

Scrum

O Scrum é um framework ágil iterativo e incremental de desenvolvimento de software para o gerenciar o desenvolvimento do produto. Essa metodologia desafia os pressupostos da abordagem "tradicional e sequencial" ao desenvolvimento de produtos e incentiva as equipes a se auto-organizarem, encorajando a colaboração próxima e online de todos os membros, bem como a comunicação presencial entre toda equipe e disciplinas no projeto. Um princípio-chave do Scrum é a "volatilidade dos requisitos", ou seja, reconhece o fato de que, durante os processos de produção, os clientes podem mudar de ideia sobre o que desejam e precisam, conforme Sachdeva (2016) descreve em um de seus trabalhos. Isso o torna extremamente importante no cenário de desenvolvimento de software, uma vez que sua flexibilidade permite que mudanças no produto sejam feitas com rapidez e eficiência, permitindo a entrega de um software que agrega valor ao cliente.

História

394

O Scrum é uma metodologia ágil de gerenciamento de projetos que tem suas raízes nas práticas desenvolvidas na indústria de software. Surgiu na década de 1980, quando Hirotaka Takeuchi e Ikujiro Nonaka publicaram um artigo seminal intitulado "The New New Product Development Game"(1986). Nesse artigo, os autores descreveram uma abordagem inovadora para o desenvolvimento de produtos que se baseava em equipes interfuncionais e iterativas, enfatizando a colaboração e a flexibilidade.

Essa abordagem foi fortemente influenciada pelos princípios do lean manufacturing e pelos conceitos de auto-organização e adaptação rápida. Schwaber e Sutherland (2017) formalizaram o Scrum como uma metodologia específica e a introduziram na indústria de software no início dos anos 90. Os autores desenvolveram o Scrum como uma resposta aos desafios enfrentados pelos projetos de software, caracterizados por mudanças frequentes nos requisitos, prazos apertados e incertezas inerentes ao desenvolvimento de software.

Ao longo dos anos, o Scrum evoluiu e se popularizou em várias indústrias além do desenvolvimento de software, incluindo gerenciamento de projetos,

marketing, pesquisa e desenvolvimento de produtos. Sua abordagem iterativa e incremental tem sido amplamente adotada devido à sua capacidade de responder rapidamente a mudanças, melhorar a comunicação e a transparência e entregar valor de forma mais eficiente.

Conceitos fundamentais

O Scrum é baseado em um conjunto de conceitos fundamentais que sustentam sua metodologia ágil de desenvolvimento de software. Esses conceitos são essenciais para compreender a estrutura e os princípios do Scrum.

O primeiro deles é a **Equipe Scrum**, composta por profissionais multidisciplinares que trabalham juntos para desenvolver o produto. Essa equipe é auto-organizada e colaborativa, responsável por definir as tarefas e garantir a entrega de incrementos de software funcionais em cada iteração. A equipe Scrum inclui papéis como o Product Owner, o Scrum Master e os membros do Time de Desenvolvimento.

O Scrum adota a abordagem de desenvolvimento iterativo e incremental, onde o trabalho é dividido em iterações chamadas de **Sprints**. Um Sprint é um período fixo de tempo, geralmente de duas a quatro semanas, durante o qual a equipe Scrum trabalha para entregar um incremento de software funcional. As Sprints são repetidas ao longo do projeto, permitindo que o produto seja desenvolvido em incrementos entregáveis.

O **Backlog do Produto** é uma lista priorizada de requisitos e funcionalidades do mesmo. É uma representação de alto nível das necessidades do cliente e dos objetivos do projeto. O Product Owner é responsável por gerenciar o Backlog do Produto, garantindo que os itens mais valiosos sejam priorizados e estejam prontos para serem implementados em cada Sprint.

O **Backlog da Sprint** é uma lista detalhada de tarefas que precisam ser concluídas durante uma Sprint. Essas tarefas são extraídas do Backlog do Produto, levando em consideração as prioridades definidas. O Time de Desenvolvimento seleciona as tarefas que eles acreditam que podem concluir dentro do período da Sprint e se compromete a entregá-las.

As **Reuniões diárias** são breves encontros diários realizados pela equipe Scrum para sincronizar o trabalho e identificar possíveis obstáculos. Essas reuniões

têm um formato estruturado, onde cada membro da equipe compartilha o que realizou desde a última reunião, o que planeja fazer até a próxima e quaisquer impedimentos que estejam enfrentando. O objetivo é promover a transparência e a colaboração entre os membros da equipe.

Compreender os conceitos fundamentais do Scrum é crucial para utilizar efetivamente essa metodologia ágil de desenvolvimento de software em um projeto. Esses conceitos foram retirados do “Guia Scrum” por Schwaber e Sutherland (2017) e fornecem as bases para a implementação e aplicação bem-sucedida do Scrum.

Aplicação

A aplicação do Scrum no desenvolvimento de software tem sido amplamente adotada em diferentes contextos e organizações. Nesta seção, serão apresentados alguns estudos e casos de aplicação do Scrum.

Em um estudo realizado por Schwaber (2004) foram analisadas aplicações do Scrum em 40 projetos de desenvolvimento de software. Os resultados mostraram que a maioria dos projetos obteve melhorias significativas em termos de produtividade, qualidade do produto e satisfação do cliente. Além disso, o estudo destacou a importância da colaboração entre os membros da equipe e a flexibilidade proporcionada pelo Scrum.

Outro estudo conduzido por Dybå e Dingsøyr (2008) investigou a aplicação do Scrum em uma organização de desenvolvimento de software. Os resultados indicaram que a adoção do Scrum levou a uma redução no tempo de desenvolvimento, maior satisfação do cliente e maior capacidade de resposta a mudanças nos requisitos do projeto. Além disso, o estudo identificou que a comunicação e a colaboração entre os membros da equipe foram aprimoradas com a implementação do Scrum.

Esses estudos e casos de aplicação demonstram a eficácia e os benefícios do Scrum na prática do desenvolvimento de software. A aplicação adequada do Scrum pode promover a agilidade, a colaboração e a entrega de valor ao cliente.

Desenvolvimento de jogos

A área de desenvolvimento de jogos combina criatividade, arte e tecnologia para criar experiências interativas envolventes e imersivas. Com o avanço da tecnologia e o crescente interesse dos jogadores, o desenvolvimento de jogos evoluiu rapidamente ao longo dos anos, resultando em uma indústria próspera e diversificada.

Ao longo desta seção, exploraremos como o desenvolvimento de games evoluiu desde os primórdios dos jogos eletrônicos até os modernos de alta qualidade. Discutiremos os conceitos fundamentais que impulsionam o desenvolvimento, como a narrativa, a jogabilidade e o design de personagens. Além disso, abordaremos as diferentes metodologias e processos utilizados no desenvolvimento de jogos, desde abordagens mais tradicionais até métodos ágeis, como o Scrum.

Ao compreendermos a evolução do desenvolvimento de jogos, os conceitos fundamentais que o sustentam, as metodologias e processos utilizados, bem como as tecnologias e ferramentas empregadas, poderemos ter uma compreensão mais ampla dessa indústria em constante transformação.

Também será apresentado um panorama das tecnologias e ferramentas que impulsionam o desenvolvimento de jogos, desde motores gráficos avançados até linguagens de programação específicas. Além disso, destacaremos os desafios enfrentados pelos desenvolvedores, como a gestão de escopo, a otimização de desempenho e a garantia de qualidade.

Ao fim desta seção, espera-se que os leitores tenham sido introduzidos ao estado atual do cenário de desenvolvimento de jogos, bem como uma visão das perspectivas futuras dessa indústria em constante evolução.

Conceitos Fundamentais

Dentro do área de desenvolvimento de jogos, existem conceitos fundamentais que são essenciais para compreender a natureza e os elementos-chave de um jogo. Nesta seção de estado da arte, exploraremos alguns desses conceitos para estabelecer uma base sólida de compreensão.

De acordo com um estudo realizado por Squire (2006), um dos conceitos fundamentais é a *narrativa*, que se refere à história contada pelo jogo. Esta pode ser linear, seguindo uma sequência predefinida de eventos, ou ramificada, permitindo que o jogador tome decisões que afetam o desenrolar da história. A narrativa contribui para a imersão do jogador e cria um contexto significativo para a jogabilidade.

Salen e Zimmerman (2004) cita em seu estudo mais três conceitos importantes, sendo eles a *jogabilidade*, *design de personagens* e *design de níveis*. Esta primeira, abrange as mecânicas e as interações que definem como o jogador interage com o jogo. A mesma pode incluir desafios, quebra-cabeças, combate, exploração e outras atividades que proporcionam diversão e engajamento ao jogador. A qualidade desse conceito é um fator-chave para o sucesso de um jogo.

O *design de personagens* é outro aspecto crucial no desenvolvimento de jogos. Personagens bem projetados possuem características distintas, motivações claras e evolução ao longo do jogo. Eles podem ser jogáveis ou não jogáveis, e desempenham papéis fundamentais na narrativa e na jogabilidade.

Além disso, é importante considerar o *design de níveis*, que envolve a criação de ambientes e espaços que o jogador irá explorar. Os níveis devem ser projetados de forma a incentivar a progressão, oferecer desafios adequados e proporcionar uma experiência envolvente para quem estiver jogando.

Esses são os principais conceitos do desenvolvimento de jogos. Compreender esses elementos e como eles se entrelaçam é essencial para criar jogos de qualidade e envolventes.

Metodologias e processos

O desenvolvimento de jogos envolve uma variedade de metodologias e processos que visam facilitar a criação de games de maneira eficiente e organizada. Nesta seção de estado da arte, exploraremos algumas das metodologias e processos mais comumente utilizados na indústria de desenvolvimento de jogos.

Uma abordagem amplamente adotada é a *metodologia cascata*, também conhecida como modelo de desenvolvimento em cascata. Aleem, Capretz e Ahmed (2016) descreve em seu estudo que metodologia é caracterizada por uma abordagem sequencial, em que cada etapa do processo de desenvolvimento é concluída

antes de avançar para a próxima. As principais etapas do modelo cascata incluem análise de requisitos, projeto, implementação, testes e manutenção. Essa abordagem é especialmente útil quando os requisitos do jogo são claros e estáveis desde o início do projeto.

Outra abordagem popular é das *metodologias ágeis*, que se baseiam em princípios de flexibilidade, colaboração e resposta rápida às mudanças. Dentro desse escopo, destaca-se o *Scrum*, que divide o desenvolvimento em ciclos iterativos chamados de *sprints*. Cada sprint tem uma duração fixa e resulta em um incremento do jogo, permitindo um desenvolvimento iterativo e adaptativo. Schwaber e Sutherland (2017) afirma que o Scrum enfatiza a colaboração da equipe, a comunicação eficaz e a entrega contínua de valor.

Além disso, é comum o uso de *metodologias híbridas* que combinam elementos de diferentes abordagens para atender às necessidades específicas de cada projeto. Essas metodologias adaptativas permitem flexibilidade na gestão de projetos de desenvolvimento de jogos, combinando vantagens de diferentes metodologias para otimizar o processo de criação do jogo (Cho, 2010).

A seleção da metodologia e do processo adequados depende das características do projeto, do tamanho da equipe, da disponibilidade de recursos e de outros fatores relevantes. Cada metodologia e processo possui suas vantagens e desafios, e é essencial adaptá-los às necessidades e realidades de cada projeto de desenvolvimento de jogos.

Tecnologias e ferramentas

No desenvolvimento de jogos, uma ampla gama de tecnologias e ferramentas é utilizada para criar experiências interativas e imersivas. Nesta seção de estado da arte, abordaremos algumas das principais tecnologias e ferramentas empregadas no desenvolvimento de jogos.

Uma das tecnologias mais relevantes é a utilização de *motores de jogos*, que são plataformas de software projetadas para simplificar e agilizar o processo de desenvolvimento. Esses motores oferecem recursos como renderização gráfica, física, gerenciamento de ativos e ferramentas de criação de níveis, permitindo que os desenvolvedores se concentrem na jogabilidade e na narrativa do jogo (Andrade, 2015).

Além dos motores de jogos, as *linguagens de programação* desempenham um papel fundamental no desenvolvimento. Tunnel e Norbistrath (2022) evidencia em seu estudo que linguagens como C++ e Java são amplamente utilizadas devido à sua eficiência, flexibilidade e suporte à criação de sistemas complexos. Linguagens de script, como Lua e Python, também são populares para prototipagem rápida e desenvolvimento de jogos indie.

Roth e Lichtman (2022) constam que outra tecnologia importante é a *realidade virtual (RV)* e a *realidade aumentada (RA)*, que oferecem novas formas de interação e imersão no mundo dos jogos. A RV proporciona uma experiência virtual completa, enquanto a RA mescla elementos virtuais com o mundo real. Ambas as tecnologias estão ganhando destaque dentro do cenário de desenvolvimento de jogos, oferecendo novas possibilidades de jogabilidade e experiências inovadoras.

No que diz respeito às ferramentas, os *editores de níveis* são amplamente utilizados para projetar e construir ambientes de jogo. Essas ferramentas permitem a criação de terrenos, modelagem de objetos, inserção de elementos interativos e ajuste da iluminação. Exemplos populares de editores de níveis incluem o Unreal Editor, o Unity Editor e o CryEngine Sandbox.

Além disso, o autor também referencia as *ferramentas de controle de versão*, que desempenham um papel importante no desenvolvimento colaborativo de jogos, permitindo que equipes de desenvolvimento trabalhem em conjunto, controlem as alterações e gerenciem o código-fonte do jogo. Exemplos comuns de ferramentas de controle de versão incluem o Git, o SVN e o Mercurial.

Essas são apenas algumas das tecnologias e ferramentas utilizadas no desenvolvimento de jogos. Com o avanço da tecnologia, novas ferramentas e tecnologias continuam a surgir, oferecendo oportunidades empolgantes para os desenvolvedores explorarem e inovarem.

Desafios

O desenvolvimento de jogos é uma área que enfrenta uma série de desafios únicos. Cagiltay (2007) afirma que um dos principais desafios é a complexidade crescente dos jogos, que exigem equipes de desenvolvimento maiores e mais especializadas. Além disso, a importância de criar jogos que sejam visualmente atraentes e

imersivos para os jogadores. Essa demanda por gráficos de alta qualidade aumenta a pressão sobre os desenvolvedores para criar jogos que sejam visualmente impressionantes.

Outro desafio significativo é a necessidade de equilibrar a jogabilidade e a narrativa do jogo. Eles devem proporcionar uma experiência envolvente e cativante, ao mesmo tempo em que oferecem desafios e recompensas aos jogadores. Além disso, a integração de elementos como inteligência artificial e realidade virtual introduz novos desafios técnicos e criativos. (Freeman *et al.*, 2020).

Ainda assim, é importante ressaltar que o desenvolvimento de jogos também oferece oportunidades emocionantes e inovadoras. Rodrigo-Yanguas *et al.*(2021) demonstra em um de seus trabalhos as recentes tendências e inovações, como a realidade virtual, realidade aumentada e jogos baseados em inteligência artificial, estão moldando a maneira como os jogos são criados e experimentados, oferecendo novas possibilidades e desafios para os desenvolvedores.

Metodologia de Pesquisa

401

A metodologia de pesquisa foi conduzida em algumas etapas para garantir uma abordagem abrangente e organizada. O processo seguiu as seguintes etapas:

- **Busca Inicial:** Inicialmente, foram realizadas buscas extensivas utilizando as plataformas Google Scholar e Semantic Scholar.
- **Seleção por Nome:** Os artigos resultantes da busca foram primeiramente avaliados com base em seus títulos.
- **Filtragem por Abstract:** Após a seleção inicial, procedeu-se com uma análise mais detalhada dos primeiros 50 artigos escolhidos, considerando os resumos de cada um.
- **Amplitude Temporal:** Dada a carência de artigos recentes na área de pesquisa, a seleção abrangeu publicações desde o ano 2000 até os dias atuais.
- **Importação e Organização:** Os artigos selecionados foram importados para o software JabRef, proporcionando uma organização eficiente da bibliografia.

Essa abordagem metodológica visa garantir a inclusão de trabalhos relevan-

tes desde o início da área de estudo até os avanços mais recentes, permitindo uma revisão abrangente da literatura existente. Esses passos foram fundamentais para a identificação, seleção e organização dos recursos bibliográficos essenciais para este estudo.

Desenvolvimento

Desafios na implementação do Scrum

A implementação do Scrum em equipes de desenvolvimento de jogos pode ser um processo complexo e repleto de desafios. Embora este framework seja uma metodologia ágil eficaz, sua aplicação no cenário específico do desenvolvimento de jogos apresenta obstáculos distintos que podem impactar negativamente o processo e o resultado final. A seguir, serão discutidos os desafios significativos que podem surgir.

Conforme descrito por Cardozo *et al.*(2010), processo de desenvolvimento de jogos é inerentemente complexo, envolvendo diversas disciplinas, como design, arte, programação e música, que devem ser integradas harmoniosamente. A falta de compreensão do Scrum pode tornar ainda mais complexa a gestão dessas múltiplas dimensões, levando a conflitos e atrasos.

A resistência à mudanças também se mostra um desafio que gestores enfrentam a muito tempo. Akif e Majeed (2012) afirma em um de seus trabalhos que a adoção do Scrum envolve uma mudança significativa na cultura de desenvolvimento. A resistência à mudança por parte dos membros da equipe e gerentes pode prejudicar a eficácia da implementação do framework.

O desenvolvimento de jogos frequentemente opera sob prazos apertados e orçamentos limitados, portanto o gerenciamento de recursos e prazos é um fator crítico e um desafio comum para os times de desenvolvimento. Uma implementação inadequada do Scrum pode tornar ainda mais desafiador o gerenciamento eficaz desses recursos e o cumprimento dos prazos estabelecidos, conforme constata Aleem, Capretz e Ahmed (2016).

Outros dois desafios são a manutenção da comunicação e colaboração e a adaptação às frequentes mudanças nos requisitos. Enquanto ao primeiro, as equipes

podem enfrentar dificuldades em realizar reuniões diárias eficazes, comunicar mudanças nos requisitos e manter uma visão compartilhada do projeto. Quando se trata das mudanças frequentes, é constatado que os requisitos de jogos tendem a evoluir à medida que as ideias se desenvolvem ao longo do projeto. A implementação inadequada da metodologia pode dificultar a adaptação a essas mudanças, levando a conflitos e retrabalho. (Mckenzie *et al.*, 2021).

O desenvolvimento de jogos exige um alto grau de criatividade e inovação. A pressão por entregas regulares pode, às vezes, sufocar a criatividade necessária para a criação de jogos inovadores. Hron e Obwegeser (2022) afirma em um de seus estudos que encontrar um equilíbrio entre a estrutura do Scrum e a liberdade criativa é crucial.

Referências bibliográficas indicam que esses desafios na implementação inadequada do Scrum têm sido objeto de discussão em vários contextos do desenvolvimento de jogos. Para superar esses obstáculos, é fundamental que as equipes de desenvolvimento de jogos recebam treinamento adequado em Scrum e adaptem a metodologia às particularidades da indústria de jogos, equilibrando criatividade com disciplina ágil.

403

Principais Causas da Má Implementação do Scrum

A inadequada implementação do Scrum em equipes de desenvolvimento de jogos pode ser atribuída a diversas causas identificadas na literatura. Tais causas são essenciais para a compreensão dos motivos pelos quais esta metodologia pode ser mal aplicada e quais fatores contribuem para a sua inadequação.

A falta de treinamento e conhecimento em Scrum é considerada por Akif e Majeed (2012) um dos principais fatores que contribuem para a má implementação do Scrum é a falta de treinamento e conhecimento apropriado do framework. Quando membros da equipe e gestores não possuem o devido treinamento, mal-entendidos sobre sua aplicação correta podem surgir.

O autor também consta que a resistência à mudança é uma barreira significativa para a implementação adequada do Scrum. Quando equipes e gestores relutam em adotar as práticas do Scrum, a integração do framework no processo de desenvolvimento é prejudicada.

Com a falta de treinamento, surge a má administração do backlog, que de acordo com McKenzie *et al.* (2021) é uma parte crucial do Scrum, e sua administração inadequada pode resultar em atrasos e problemas de qualidade. Isso inclui a falta de priorização adequada, descrições de itens de backlog incompletas ou imprecisas e dificuldades em manter a fila atualizada.

A definição da duração dos sprints é outro aspecto crítico do Scrum. A falta de compreensão sobre como determinar a duração apropriada dos sprints pode levar a problemas de escopo e qualidade, com sprints muito curtos ou muito longos (Cardozo *et al.*, 2010).

A identificação dessas causas é fundamental para aprimorar a implementação do Scrum nas equipes de desenvolvimento de jogos, mitigando os obstáculos que podem surgir durante o processo.

Impacto da Falta de Treinamento em Scrum no produto final

A ausência de treinamento adequado em Scrum pode acarretar consequências relevantes no produto final do desenvolvimento de jogos, conforme documentado anteriormente.

Quando as equipes de desenvolvimento de jogos carecem de treinamento apropriado em Scrum, podem encontrar dificuldades na aplicação das práticas do framework, resultando em uma redução na qualidade do produto final, consta Akif e Majeed (2012).

Além disso Hron e Obwegeser (2022) afirmam que, atrasos no desenvolvimento são desencadeados pela falta de capacitação em Scrum, resultando no aumento de custos e no prazo de produção do jogo.

Cardozo *et al.* (2010) também relata que a insatisfação do cliente, é inevitável uma vez que a empresa desenvolvedora não consiga entregar um jogo de qualidade, já que a ocorrência de bugs e outros problemas tornam a experiência frustrante e chata.

Esses impactos podem ter sérias consequências para o êxito do projeto de desenvolvimento de jogos, destacando a importância do treinamento adequado em Scrum para assegurar a qualidade e a satisfação do cliente no produto final.

CONCLUSÃO

O presente estudo buscou investigar a relação entre a má implementação do Scrum e seu impacto negativo no desenvolvimento de jogos.

A questão de pesquisa inicial, "Como a má implementação do Scrum afeta negativamente o desenvolvimento de jogos?" direcionou a investigação. Também foi estabelecida a hipótese de que a má implementação da metodologia está associada a atrasos, aumento de custos e insatisfação do cliente no desenvolvimento de jogos.

Ao analisar os principais resultados da pesquisa, concluiu-se que a má implementação do Scrum desencadeia, de fato, uma série de desafios significativos no processo de desenvolvimento de jogos. Os desafios identificados na implementação inadequada incluem a resistência à mudança, a falta de treinamento e a dificuldade em lidar com mudanças frequentes nos requisitos. Além disso, as principais causas da má implementação do framework abrangem a falta de conhecimento sobre o mesmo e a dificuldade de adaptação às necessidades específicas dos projetos de desenvolvimento de jogos. O impacto da falta de treinamento da metodologia no produto final inclui a redução da qualidade do produto, aumento de custos, atrasos no desenvolvimento e insatisfação do cliente.

Esta pesquisa também pode contribuir para a conscientização sobre os desafios associados à implementação do Scrum no contexto de desenvolvimento de jogos, incentivando futuros estudos e práticas que abordem essas questões de forma mais aprofundada.

REFERÊNCIAS

AKIF, R.; MAJEED, H. Issues and challenges in scrum implementation. *International Journal of Scientific & Engineering Research*, v. 3, n. 8, p. 1–4, 2012.

ALEEM, S.; CAPRETZ, L. F.; AHMED, F. Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, SpringerOpen, v. 4, n. 1, p. 1–30, 2016.

ANDRADE, A. Game engines: a survey. *EAI Endorsed Transactions on Serious Games*, v. 2, n. 6, 2015.

CAGILTAY, N. E. Teaching software engineering by means of computer-game development: Challenges and opportunities. *British Journal of Educational Technology*, Wiley Online Library, v. 38, n. 3, p. 405–415, 2007.

CARDOZO, E. S. et al. Scrum and productivity in software projects: a systematic literature review. In: *14th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. [S.l.: s.n.], 2010. p. 1-4.

CHO, J. J. An exploratory study on issues and challenges of agile software development with scrum. *All Graduate theses and dissertations*, p. 599, 2010.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, v. 50, p. 833–859, 08 2008.

FREEMAN, G. et al. "pro-amateur-driven technological innovation: Participation and challenges in indie game development. *Proceedings of the ACM on Human-Computer Interaction*, ACM New York, NY, USA, v. 4, n. GROUP, p. 1–22, 2020.

HRON, M.; OBWEGESER, N. Why and how is scrum being adapted in practice: A systematic review. *Journal of Systems and Software*, Elsevier, v. 183, p. 111110, 2022.

KASURINEN, J. Games as software: Similarities and differences between the implementation projects. In: *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*. [S.l.: s.n.], 2016. p. 33–40.

406

MCKENZIE, T. et al. Is agile not agile enough? a study on how agile is applied and misapplied in the video game development industry. In: IEEE. *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*. [S.l.], 2021. p. 94–105.

POLITOWSKI, C.; PETRILLO, F. A survey of video game testing. In: IEEE. *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*. [S.l.], 2021. p. 90–99.

RODRIGO-YANGUAS, M. et al. "the secret trail of moon (tstm)": Proof of concept and usability of a virtual reality chess gamified serious video game to treat adhd. (preprint). 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:241104572>. Acesso em: 23 out. 2023.

ROTH, A.; LICHTMAN, D. The community game development toolkit. *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:253762192>. Acesso em: 23 out. 2023.

RUONALA, H.-R. et al. *Agile game development: A systematic literature review*. Helsingfors universitet, 2017.

SACHDEVA, S. Scrum methodology. *Int. J. Eng. Comput. Sci*, v. 5, n. 16792, p. 16792–16800, 2016.

SALEN, K.; ZIMMERMAN, E. *The Game Design Reader: A Book of Lenses*. The MIT Press, 2004. Disponível em: <https://mitpress.mit.edu/books/game-design-reader>. Acesso em: 23 out. 2023.

SCHWABER, K. *Agile Project Management with Scrum*. [S.l.]: Microsoft Press, 2004.

SCHWABER, K.; SUTHERLAND, J. *Scrum Guide*. 2017. Disponível em: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. Acesso em: 23 out. 2023.

SQUIRE, K. From content to context: Videogames as designed experience. *Educational Researcher*, v. 35, p. 19 – 29, 2006.

TUNNEL, R.-H.; NORBISRATH, U. A survey of estonian video game industry needs. *Journal of Education and Learning*, 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:251204249>. Acesso em: 23 out. 2023.