

Um Processo de Desenvolvimento de Jogos Digitais adotando Práticas Ágeis

A Digital Game Development Process Adopting Agile Practices

Pedro Miranda Duarte de Castro¹, Sandro Ronaldo Bezerra Oliveira¹

¹Programa de Pós-Graduação em Computação (PPGCOMP) - Universidade Federal do Pará (UFPA), Belém - PA - Brasil

pedro.castro@icen.ufpa.br, srbo@ufpa.br

Abstract. Introduction: Digital games have volatile requirements that demand adapted methodologies for their development. However, there is a lack of specific guidelines in the literature to guide independent, multidisciplinary teams in developing and executing game development processes. **Objective:** To mitigate this gap, this study seeks to provide a detailed and easily instantiable methodology for these teams. **Methodology:** Thus, a literature review and a field research were conducted to subsequently model and validate a solution through a peer review with experts. **Results:** As a result, the GDP (Game Development Process) was obtained, a process that provides a validated framework that organizes team management and ensures the quality of the game design.

Keywords Games, Development Process, Software Engineering, Game Design, Agile.

Resumo. Introdução: Jogos digitais possuem requisitos voláteis que exigem metodologias adaptadas para o seu desenvolvimento. Contudo, há uma carência de orientações específicas na literatura para guiar equipes multidisciplinares independentes a desenvolver e executar processos de desenvolvimento para jogos. **Objetivo:** Para mitigar essa lacuna, busca-se fornecer uma metodologia detalhada e de fácil instanciização para essas equipes. **Metodologia:** Desse modo, realizou-se uma revisão bibliográfica e uma pesquisa de campo para posteriormente modelar e validar uma solução a partir de uma revisão por pares com especialistas. **Resultados:** Como resultado, obteve-se o GDP (Game Development Process), um processo que fornece uma estrutura validada que organiza a gerência de equipes e garante a qualidade do design do jogo.

Palavras-Chave Jogos, Processo de Desenvolvimento, Engenharia de Software, Game Design, Ágil.

1. Introdução

Jogos são atividades lúdicas e voluntárias que visam distrair e entreter indivíduos, a fim de que disputem objetivos não reais e simbólicos por meio de regras conhecidas e estabelecidas [Huizinga 2008]. O avanço das tecnologias digitais fez com que esses jogos fossem adaptados para a mídia digital, tornando-se *softwares* de computador. Caillois afirma que eles constituem sistemas de regras que definem o que pertence e o que não pertence ao jogo. Ou seja, determinam-se leis arbitrárias que cooperam com os recursos

lúdicos da atividade desenvolvida [Caillois 2017]. Dessa forma, como os jogos digitais se relacionam ao uso de dispositivos eletrônicos para sua execução e participação, com regras regidas pelo computador [Schuytema 2008], seu uso está estritamente dependente do meio digital.

Portanto, estratégias para a confecção e elaboração desses *softwares* são altamente recomendadas para garantir uma boa qualidade do produto final. Essas estratégias são desenvolvidas e gerenciadas pela Engenharia de *Software*, que é um segmento da Ciência da Computação que visa gerenciar a confecção de um *Software* [Oliveira 2006]. Mais especificamente, ela compreende o *software* como um produto, normalmente elaborado por meio de um projeto que, ao ser executado, corresponde a um processo que será adotado [de Pádua Paula Filho 2003].

Processo é uma sequência de atividades e métodos ordenados que são usados para atingir uma meta [de Pádua Paula Filho 2003]. Podendo ser utilizado em vários projetos, ele têm a função principal de criação, manutenção e teste de *software* [Hermano 2023]; portanto, é a ferramenta fundamental para gerenciar e controlar projetos de *software* [Humphrey 1989].

Apesar de haver inúmeros métodos de produção e de execução de processos de *softwares* tradicionais, percebeu-se que há uma ausência de documentação e de orientações direcionadas especificamente ao desenvolvimento de processos para jogos digitais. Isso se deve ao fato de que jogos digitais, apesar de serem *softwares* de computador, apresentam métodos de construção diferentes das metodologias padrões, necessitando de adaptações em seus processos de confecção [Benigno et al. 2024]. Portanto, a falta de processos adaptados, instanciáveis e genéricos para o desenvolvimento de jogos torna-se um problema para equipes especializadas na criação desses *softwares*.

Diante desse contexto, Petrillo afirma que os principais fatores para o fracasso de um jogo estão na gerência e não nas tecnologias [Petrillo et al. 2008]. Dessa forma, a lacuna de documentação será mitigada ao fornecer um modelo de processo instanciável específico para projetos de jogos, o que é fundamental para incentivar e apoiar o desenvolvimento deles em equipes pequenas e independentes.

Por fim, foi desenvolvido, no decorrer das pesquisas deste trabalho, um diagrama de atividades UML (*Unified Modeling Language*) para representar um processo de desenvolvimento de jogos chamado GDP (*Game Development Process*), visando as práticas ágeis recomendadas pela literatura. O GDP foi avaliado por especialistas em jogos por meio de uma revisão por pares, que será abordada com mais detalhes na Seção 4.2.

Além desta seção introdutória, a Seção 2 aborda trabalhos relacionados e suas contribuições para a produção de um processo de desenvolvimento de jogos. A Seção 3 explica o fluxo de trabalho para o desenvolvimento desta pesquisa e do processo confeccionado. A Seção 4 apresenta o processo proposto após as análises realizadas durante a revisão por pares. A seção 5 discorre sobre as principais adições do processo em relação ao que é abordado na literatura. Por fim, a Seção 6 apresenta uma visão geral deste trabalho, além de discutir as contribuições, as limitações e os trabalhos futuros.

2. Trabalhos Relacionados

Com a intenção de facilitar o desenvolvimento de jogos a partir de um processo que tem como principal objetivo ser instanciável e genérico para diferentes projetos de jogos, foi realizada uma pesquisa abrangente na literatura para definir e compreender as exigências e as necessidades desse processo. A partir disso, pôde-se analisar o referencial obtido e confeccionar o modelo com base em sua referência.

Entre os trabalhos relacionados, destacam-se: (1) **revisões terciárias da literatura**, que foram fundamentais para compreender o contexto da indústria de jogos e levantar os processos de desenvolvimento utilizados pelas equipes; e (2) **trabalhos com o propósito de desenvolver um modelo de processo**, a fim de obter referências sobre as práticas adotadas nos processos e como eles são utilizados no ambiente de desenvolvimento de jogos.

Entre as revisões terciárias, o trabalho [Benigno et al. 2024] foi excepcional para conhecer e levantar os processos de desenvolvimento utilizados na construção de jogos. Neste trabalho, foram observadas as diversas características a respeito dos jogos digitais, as metodologias utilizadas em seu desenvolvimento e o relacionamento entre as especificidades dos jogos e as adaptações de métodos da Engenharia de *Software*. Os autores apontam uma tendência para métodos iterativos, incrementais e ágeis em equipes de desenvolvimento, incluindo o *Scrum*, o Desenvolvimento Orientado a Modelos e o XP (*eXtreme Programming*). Devido ao fato dos requisitos de jogos serem muito voláteis, há a necessidade de práticas mais flexíveis, pautadas no *feedback*. Além disso, os autores defendem que as fases iniciais devem estar voltadas para definir um desenvolvimento mais direcionado, apontando para a elaboração de um GDD (*Game Design Document*)

A pesquisa relatada em [Politowski et al. 2016] explora a relação entre a taxa de sucesso, a taxa de fracasso, as dificuldades e os problemas encontrados em projetos relativos à metodologia utilizada. Em sua abordagem, os autores dividem as metodologias em quatro: (1) **Ágil**, abordagens iterativas com aprimoramentos contínuos no processo, entre elas o *Scrum*, o XP, o *Kanban*, o ASD (*Adaptative Software Development*) e o FDD (*Feature Driven Development*); (2) **Preditivo**, processos que derivam do modelo Cascata, exigindo prazos estritos, pois são lineares e sequenciais, necessitando que todos os seus requisitos sejam pré definidos, entre eles o próprio modelo Cascata e o RUP (*Rational Unified Process*); (3) **Ad-hoc**, processos que não são considerados nem ágeis nem preditivos, são totalmente personalizáveis pela equipe e geralmente são temporários, usados apenas para atender a uma determinada necessidade; e (4) **Sem-processo ou “code-&-fix approach”**, que são equipes que nunca utilizaram processos de desenvolvimento.

O resultado dessa pesquisa mostrou que as principais dificuldades são o atraso, a falta de documentação e, principalmente, escopos não realistas. Além disso, os autores apontam que um dos principais motivos pelos quais as equipes não adotam abordagens sistemáticas para desenvolver jogos é que elas não possuem conhecimento ou experiência para aplicá-las, além de possuírem equipes pequenas e pouco tempo. Por fim, os autores também demonstram que abordagens sistemáticas, sejam elas quais forem, apresentam taxas de sucesso consideravelmente mais elevadas do que as abordagens casuais.

O trabalho de Neto [Neto et al. 2015] realiza uma revisão da literatura sobre

processos de desenvolvimento para jogos educacionais. Apesar do contexto pedagógico não ser relevante para o presente trabalho, esta revisão evidenciou que os processos encontrados foram adaptações de metodologias já existentes na engenharia de *software*.

Entre os processos elaborados, o artigo [Battistella e von Wangenheim 2016] apresenta um processo para a criação de jogos educacionais que dá ênfase tanto aos processos de desenvolvimento de *software* quanto aos aspectos de *design* instrucional, fator ausente nos demais processos de jogos educativos. Apesar do processo desenvolvido neste artigo ser direcionado a jogos educacionais, é muito interessante notar o esforço para elaborar atividades bem explicativas, papéis definidos e o fluxo de desenvolvimento do jogo, desde a análise do jogo até a concepção, a implementação, entre outras fases.

Os autores em [Freitas et al. 2017] citam que os maiores problemas no desenvolvimento de jogos estão na gerência [Petrillo et al. 2008], e isso se deve ao fato de que usar um processo é “assustador” para desenvolvedores independentes e com pouca experiência em Engenharia de Software. Devido a esse fator, os processos *ad-hoc* ou outros tipos de processos que não garantem qualidade de *software* são mais utilizados [Góes e de Almeida 2012]. Os autores afirmam que os processos artísticos são muito diferentes dos processos utilizados pelos desenvolvedores; portanto, deve haver uma conciliação entre as diferentes equipes do projeto. Nesse contexto, o processo artístico serve como base para estabelecer as demais estruturas e tarefas do projeto:

“Assim, o processo proposto soma as três atividades geradas com base no processo artístico (concepção da ideia, criação artística e desenvolvimento do *design*) combinadas com um processo *Scrum*. O processo inicia com a atividade de concepção da ideia que produz o GDD (artefato obrigatório com nível de detalhamento definido pela equipe), que por sua vez gera a lista de atividades necessárias para desenvolver o jogo.”

O trabalho [Neto et al. 2021] apresenta um processo de desenvolvimento baseado em três modelos principais: (1) *Scrum*; (2) *Extreme Game Development (XGD)*, uma adaptação do XP para jogos; e (3) *Game Waterfall Process*, uma adaptação do modelo cascata para jogos. Neste processo, definiram-se vários papéis que atuam em diferentes frentes no desenvolvimento do jogo, além de definir tarefas e artefatos contextualizados no ambiente de jogos, com ênfase na praticidade e no detalhamento técnico.

Portanto, o presente artigo possui um propósito semelhante ao dos trabalhos apresentados nesta seção, mas propõe elaborar um processo de desenvolvimento que tenha foco tanto na técnica quanto no *design* do jogo; de forma flexível e instanciável por outras equipes de desenvolvimento.

3. Metodologia da Pesquisa

A metodologia utilizada na pesquisa deste trabalho envolveu na consolidação de uma base teórica acadêmica estruturada antes de atuar na parte prática do trabalho, garantindo que o processo desenvolvido seguisse as práticas recomendadas na literatura.

Para isso, foram definidas as seguintes tarefas: (1) Pesquisar sobre processos de desenvolvimento; (2) Compreender o que são jogos e defini-los; (3) Realizar pesquisa bibliográfica para levantar processos de desenvolvimento de jogos; (4) Realizar uma

Pesquisa Participante; (5) Desenvolver um processo baseado nas pesquisas elaboradas; e (6) Realizar uma verificação do processo a partir da adoção da revisão por pares.

As pesquisas sobre processos de desenvolvimento foram realizadas por meio de buscas na literatura sobre Engenharia de *Software*, projetos de *software* e processos de desenvolvimento. Essas pesquisas estabelecem uma base teórica essencial sobre a Engenharia de *Software* e seus campos de atuação. Ao pesquisar mais sobre os processos, buscou-se compreender sua definição, suas utilidades e suas etapas de desenvolvimento.

Após essa etapa, buscaram-se informações sobre a definição de jogos e de jogos digitais. Essa busca deu-se pela necessidade de compreender o que são jogos; quais são as diferenças entre um jogo tradicional e um *videogame*; e de que forma esses jogos podem ser definidos e representados.

Prosseguindo, a pesquisa bibliográfica sobre processos de desenvolvimento para jogos foi executada visando apenas o levantamento desses processos. Nesse sentido, não foram levadas em consideração quais atividades eram realizadas nesses processos, nem como se davam seus fluxos; apenas quais processos estavam sendo utilizados e que tipo de metodologia foi desenvolvida, adaptada, entre outros. Apesar desse foco na pesquisa, considerou-se somente a natureza do jogo, isto é, se o jogo possuía propósitos educativos ou não. Dessa forma, devido ao fato de que jogos educativos e *serious games* possuem requisitos específicos para sua confecção [Battistella e von Wangenheim 2016], considerando que fatores pedagógicos são levados em conta, seus processos também sofrem alterações significativas que não foram consideradas para o foco do processo proposto neste trabalho.

Paralelamente às atividades anteriores, realizou-se uma pesquisa participante, que resultou na experiência prática sobre as atividades realizadas nos ambientes de desenvolvimento de jogos. Essa participação deu-se por meio do desenvolvimento de um jogo digital em um laboratório de engenharia de *software*, resultando na apresentação de artigos no XXIV SBGames (Simpósio Brasileiro de Jogos e Entretenimento Digital) [de Castro et al. 2025, Shibata et al. 2025].

Para elaborar o processo, utilizou-se a ferramenta *Astah UML* para a confecção do diagrama de atividades referente ao modelo proposto. Esse processo foi realizado por meio de referências a outros processos, adotando práticas ágeis, como o *Kanban*, e inspirando-se no modelo *Scrum*. Além do processo, também foi elaborado, paralelamente, um documento descritivo contendo informações sobre as atividades/tarefas constantes nos fluxos, que são: finalidade, artefatos de entrada e de saída, procedimentos e recursos humanos e não humanos.

Por fim, realizou-se uma revisão por pares para garantir que o processo apresentado seja verificado por especialistas na área, evitando vieses e obtendo um posicionamento técnico definitivo.

4. Resultados

O processo elaborado neste trabalho está apresentado na Figura 1. Este processo é a versão definitiva, que contempla as alterações realizadas a partir das solicitações feitas pelos especialistas durante a execução da revisão por pares, que será abordada com mais detalhes na Seção 4.2. É importante notar que os fluxos usados para caracterizar o

processo são definidos por atividades/tarefas, contendo a representação gráfica (formato de um documento) de artefatos de entrada (lado esquerdo) e de saída (lado direito), bem como os papéis responsáveis mapeados nas raias que dividem o fluxo. O detalhamento de cada atividade/tarefa constante nos fluxos do processo encontra-se definido em <https://zenodo.org/records/19768846>, que contempla a finalidade, os artefatos de entrada e saída, o procedimento, os recursos humanos e os recursos não humanos.

4.1. Game Development Process

O GDP é um processo de desenvolvimento de jogos modelado por meio de um diagrama de atividades UML, com foco em equipes independentes. Nesse contexto, ele não possui documentações rígidas a serem seguidas, o que favorece sua instanciação por diferentes equipes. Este processo abrange desde a concepção da ideia do jogo até o seu lançamento e pós-lançamento. Dessa forma, este processo é utilizado tanto para garantir a organização do desenvolvimento do jogo a nível de processo, ou seja, de gerência, quanto a qualidade do jogo a nível de *game design*, mecânicas, arte, entre outros.

O GDP também contempla validações com as *Personas* interessadas no jogo, de modo que o *feedback* obtido nessa validação é utilizado para verificar se elas estão satisfeitas e interessadas nas implementações do jogo, além de coletar possíveis erros na codificação e na execução do jogo. Cada atividade enumerada dos fluxos é detalhada a seguir em subfluxos, para um melhor entendimento de sua execução.

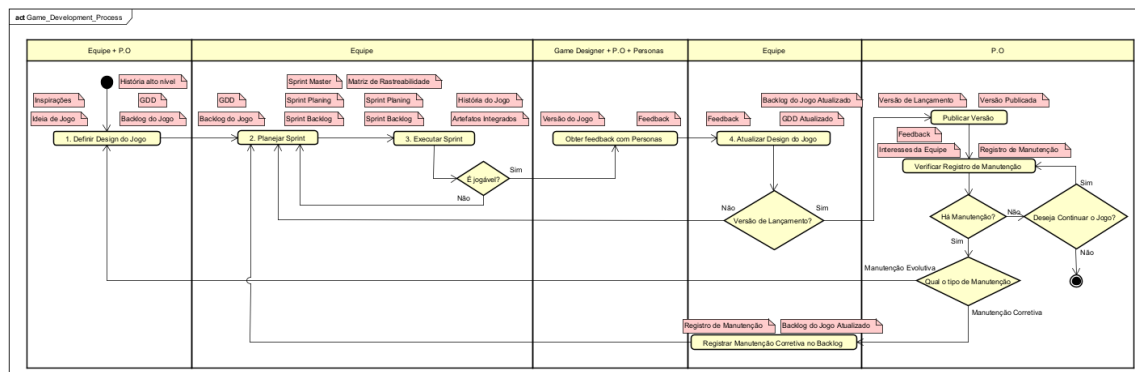


Figura 1. Fluxo principal do GDP

O subfluxo da atividade “1. Definir Design do Jogo”, representada na Figura 2, foi concebido para determinar e documentar informações sobre o *design* do jogo, alinhando as ideias e as inspirações do *game designer* com o restante da equipe. Para isso, realiza-se um *brainstorm* em paralelo ao desenvolvimento do *Backlog* e do GDD, a fim de garantir que as ideias sejam devidamente registradas e alinhadas ao *design*, para a elaboração de um GDD consistente. Assim, assegura-se a conformidade entre o *Backlog* do jogo e o GDD, mantendo coerência entre o *design* e o desenvolvimento.

Na atividade “1.1 Desenvolver Backlog”, cujo fluxo é representado pela Figura 3, a criação do *Backlog* do jogo é realizada de forma organizada, visando a dependência entre tarefas e suas prioridades no jogo. Conforme mais ideias de histórias de usuário (*User Story*) são geradas no GDD, essas histórias são transformadas em tarefas para que sejam registradas corretamente no *Backlog*, e suas prioridades e dependências são

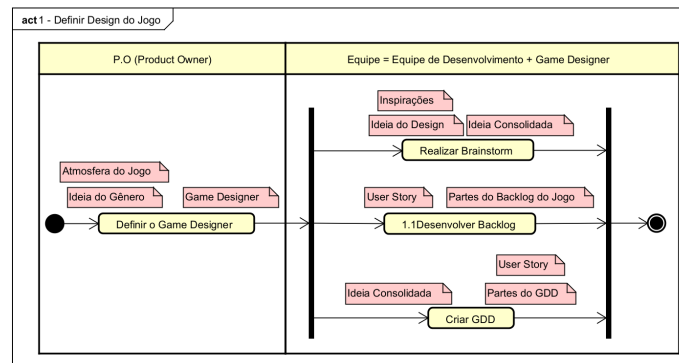


Figura 2. Atividade 1 - Definição de *design*

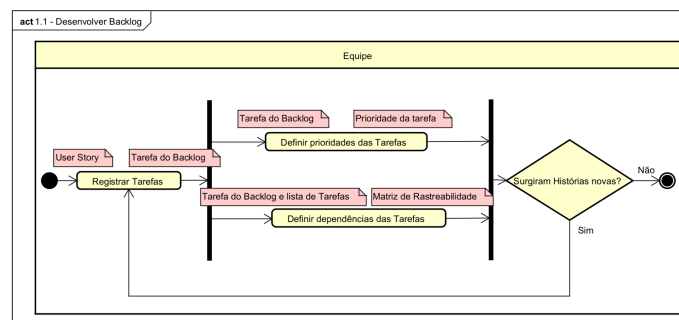


Figura 3. Atividade 1.1 - Desenvolvimento do *Backlog* do jogo

reavaliadas, considerando as demais tarefas já registradas. Essa abordagem prevê que a organização inicial do jogo previna problemas futuros de escopo.

A atividade “2. Planejar Sprint”, cujo fluxo é representado na Figura 4, segue a estrutura da cerimônia da *Sprint planning* prevista pelo *Scrum*, onde o foco está em definir os objetivos da *Sprint*, sua duração e quem serão as pessoas responsáveis pelas tarefas definidas [Scrum Alliance 2026]. A principal adaptação que está presente no GDP é o artefato de entrada “Prazo para um Épico”, que é a expectativa de obter uma seção do jogo “jogável” e apresentável, com diversos elementos de *gameplay* integrados, de modo que seja possível a um usuário jogar e testar as funcionalidades implementadas.

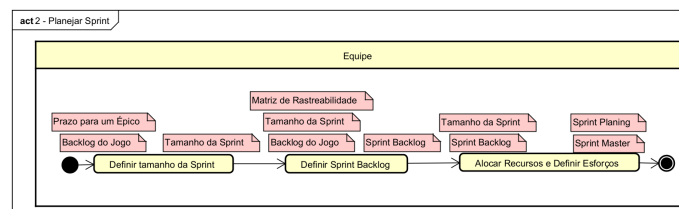


Figura 4. Atividade 2 - Planejamento de *Sprint*

Na atividade “3. Executar Sprint”, cujo fluxo pode ser visto na (Figura 5), as tarefas definidas no *Sprint Backlog* são realizadas pela equipe e integradas pelos desenvolvedores no ambiente de programação do jogo. É válido ressaltar que, na atividade “3.1 Executar Tarefas do Backlog”, cujo fluxo por ser visto na Figura 6, a execução das tarefas do *Backlog* está inteiramente contextualizada para ambientes de desenvolvimento de jogos, com atores multidisciplinares e artefatos específicos, como *level design*, *concept*

arts, história do jogo, programação, entre outros definidos no fluxo.

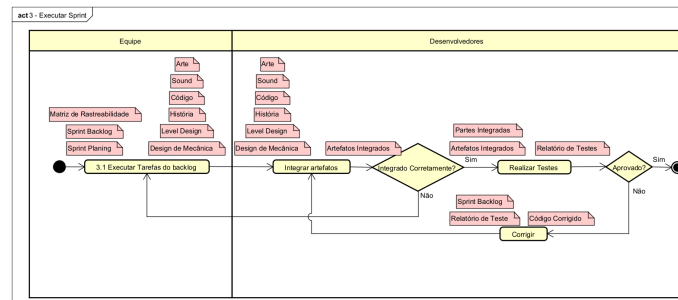


Figura 5. Atividade 3 - Execução da *Sprint*

Ainda na atividade de execução das tarefas do *Backlog*, vide Figura 6, a equipe do jogo realiza diversos trabalhos paralelos para otimizar o tempo de produção da *Sprint*. Esta abordagem particionada também organiza de forma mais pontual o que cada membro da equipe deve realizar. Como jogos são projetos multidisciplinares, a atividade nomeada “3. Executar Sprint”, sem detalhamentos, não engloba todos os atores presentes em uma equipe de desenvolvimento de jogos, nem a qualidade do *design* do projeto. É importante ressaltar que o material “*So You Wanna Make Games?*” foi essencial para compreender o processo de *concept art*, visto que se trata de uma etapa importante no desenvolvimento do *design* de um jogo [Riot Games 2025].

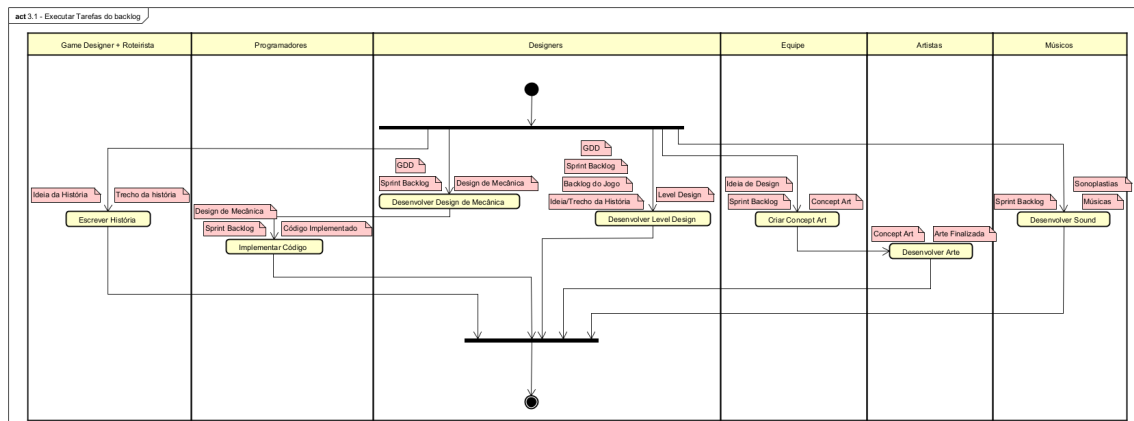


Figura 6. Atividade 3.1 - Execução das tarefas do *Backlog*

Após a execução das tarefas do *Backlog*, é necessário que haja a integração dos artefatos gerados nessa execução com as demais partes integradas, seguida de verificações e testes para garantir que tudo foi integrado e implementado corretamente, representadas pelas atividades “Integrar artefatos” e “Realizar Testes” definidas na Figura 5.

Por fim, a última atividade do processo chamada de “4. Atualizar Design do Jogo”, cujo fluxo pode ser visto na Figura 7, trata das possíveis mudanças que podem ocorrer no *design* do jogo após o *feedback* das *personas*. Essa atualização de *design* ocorre de forma sequencial, semelhante à sua definição (vide Figura 2). Desta forma, para atualizar o *design*, primeiro é necessário atualizar o GDD com as novas alterações e, posteriormente, registrar essas alterações no *Backlog* por meio de um procedimento idêntico ao da atividade “1.1 Desenvolver Backlog”, definida na Figura 3.

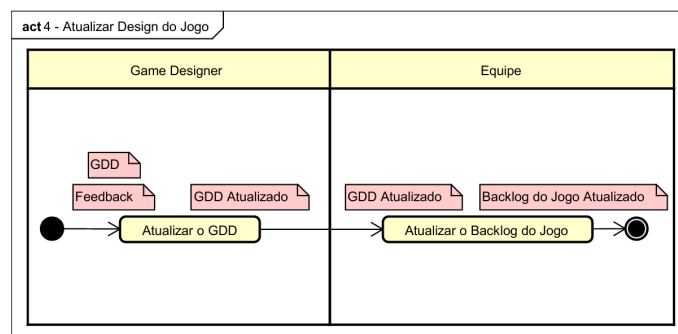


Figura 7. Atividade 4 - Atualizar *Design* do Jogo

4.2. Avaliação do Processo

A técnica de revisão por pares foi utilizada para verificar a qualidade do processo e obter um posicionamento imparcial a respeito dele, visto que todos os participantes da pesquisa atuaram ativamente em sua concepção e elaboração. Para isso, fez-se contato com dois especialistas em desenvolvimento de jogos e, posteriormente, eles foram requisitados, por meio de aplicativos de mensagem, para realizar a revisão em uma reunião remota.

Os avaliadores, ambos especialistas em jogos, realizaram a revisão simultaneamente. O Avaliador 1 avaliou a qualidade do processo a nível de Engenharia de *Software*, pois possui mestrado e doutorado na área, sendo responsável por apontar erros de fluxo, a necessidade de artefatos específicos, entre outros. Enquanto isso, o Avaliador 2 avaliou a aplicação prática e a realidade do desenvolvimento de jogos, pois atua como organizador em grupos de desenvolvimento de jogos da universidade em que leciona, contribuindo para a inclusão de tarefas importantes para o *design* do jogo.

Os revisores acessaram o processo apenas no momento da avaliação. A reunião foi gravada para que a coleta de comentários ocorresse de forma íntegra, e conforme o GDP era apresentado, os avaliadores levantavam dúvidas e recomendações. Para sua análise, utilizou-se uma Inteligência Artificial para transcrever a revisão por pares e categorizar todas as sugestões levantadas pelos revisores. As melhorias e correções apontadas pelos especialistas podem ser acessadas em: <https://zenodo.org/records/19769001>.

Todos esses achados foram implementados na versão apresentada do processo na Seção 4, com exceção da sugestão referente ao detalhamento de objetivos de *marketing* e de vendas, pois o escopo atual do processo está restrito à gerência e à produção do projeto.

Por fim, os especialistas apontaram fortalezas após a análise do processo, a saber: **Divisão de processo e subprocesso**, a divisão detalhada dos fluxos mais complexos em sub-fluxos, como a execução da *Sprint*, foi fundamental para visualizar a sequência lógica do processo de forma clara; **Contextualização de papéis**, a delimitação de papéis específicos (programador, *game designer*) está de acordo com a área de jogos, onde as funções são especializadas; **Uso de termos genéricos para as Tarefas**, a utilização do termo “tarefa” foi elogiada devido ao fato de que, no desenvolvimento de jogos, o *Backlog* é essencialmente multidisciplinar, e seu uso abrange as necessidades artísticas, sonoras e de *design*; **Organização do processo**, a estrutura de desenvolvimento de jogos foi considerada bem organizada e visível, pois os especialistas afirmam que o processo desenvolvido serve como um ponto de partida sólido para organizar projetos;

a **Boa documentação**, os especialistas afirmam que o documento auxiliar produzido está bem escrito e fornece um guia considerável para compreender o processo, reforçando a qualidade acadêmica e a facilidade de entendimento do modelo proposto.

5. Discussão

Os trabalhos analisados neste artigo sugerem metodologias e estratégias para conciliar a engenharia de *software* e o desenvolvimento de jogos digitais. Uma das estratégias levantadas é a definição do GDD antes do desenvolvimento do *backlog* [Freitas et al. 2017]. O GDP, por outro lado, sugere a criação desses artefatos paralelamente para garantir que o *design* do jogo seja feito de forma integrada entre a equipe. Essa abordagem permite que equipes independentes atuem colaborativamente na elaboração do *design* e do *backlog*, de modo que não seja necessário definir completamente o *design* para documentar as tarefas que serão adicionadas.

Além disso, este trabalho definiu com mais detalhes e contextualização os papéis envolvidos no desenvolvimento de um jogo, diferentemente do *Scrum* tradicional, que utiliza terminologias genéricas como “Equipe *Scrum*” para definir os recursos humanos das tarefas das *Sprints*. Essa definição contextualizada também resolve a grande diferença entre as equipes artísticas e técnicas do desenvolvimento de jogos [Freitas et al. 2017], conciliando suas atividades em um processo integrado e contextualizado.

Por fim, o GDP lida com a volatilidade dos requisitos de jogos, evidenciada em [Aleem et al. 2016, Benigno et al. 2024], ao verificar os *feedbacks* das *personas*, atualizando o GDD e o *backlog* em sequência. Essa solução arquitetural garante o aspecto ágil do processo, pois lida com os resultados do ciclo de execução anterior.

6. Conclusões

Neste trabalho foi desenvolvido um processo com o propósito de mitigar a lacuna de documentação encontrada na literatura a respeito de metodologias para o desenvolvimento de jogos aliadas ao *game design*. Nesse sentido, as avaliações realizadas apontam para o sucesso do objetivo do trabalho, que obteve êxito ao produzir um modelo de processo que serve para organizar equipes pequenas e garantir a boa qualidade do *design* de um jogo.

Desta forma, acredita-se que as principais contribuições deste trabalho estão concentradas na usabilidade do GDP — que foi projetado para ser utilizado em diferentes projetos — e na contextualização de sua documentação com o desenvolvimento de jogos ao tratar de atores e de tarefas multidisciplinares.

Entretanto, apesar do foco do processo ser a generalização de seus fluxos, oferecendo liberdade para as equipes, a ausência de *templates* na documentação e a falta de detalhamento na elaboração do GDD foram identificadas como limitações. Além disso, as atividades de execução do *Backlog* poderiam ser mais otimizadas caso fosse elaborado um fluxo sequencial específico para elas, em vez de um paralelismo generalizado; sendo necessário, portanto, reavaliar a documentação e os fluxos de desenvolvimento para aprimorar ainda mais a usabilidade do GDP.

Por fim, pretende-se realizar trabalhos futuros para tratar as limitações. Embora o processo tenha sido verificado por um painel de especialistas, ele não foi implantado, aplicado ou testado em um ambiente de desenvolvimento prático, havendo, portanto, planejamento e intenção de desenvolver projetos gerenciados pelo GDP.

Referências

- Aleem, S., Capretz, L. F., e Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4(1):6.
- Battistella, P. E. e von Wangenheim, C. G. (2016). Engaged: Um processo de desenvolvimento de jogos para ensinar computação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 380.
- Benigno, M. V. B., Teixeira, E. N., e Werner, C. (2024). Processos de software para jogos digitais: uma revisão terciária da literatura. *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 665–680.
- Caillois, R. (2017). *Os jogos e os homens: a máscara e a vertigem*. Editora Vozes Limitada.
- de Castro, P. M., Soares, B. Y. C., de Lima Alves, C., de Lima, M. S., Quaresma, T. M., Shibata, N. N., de Souza Albuquerque, D., Cruz, L. P., e Oliveira, S. R. B. (2025). O design do mapa de fases do tekohá: Um jogo para conservação do meio ambiente adotando características da cultura e das lendas paraenses. *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 73–83.
- de Pádua Paula Filho, W. (2003). *Engenharia de software*, volume 2. Ltc.
- Freitas, C. E., Callado, A. d. C., Jucá, P., e Silva, D. (2017). Um processo ágil multidisciplinar de desenvolvimento de jogos para estúdios independentes. *Proceedings of SBGames*, pages 1232–1235.
- Góes, R. A. S. V. A. e de Almeida, L. F. (2012). Metodologia origame: um processo de desenvolvimento de jogos. *Proceedings of SBGames*.
- Hermano, P. (2023). Processo de desenvolvimento de software: conheça as etapas. Disponível em: <https://www.attri.com.br/blog/processo-de-desenvolvimento-de-software/>. Acesso em: 18 jan. 2026.
- Huizinga, J. (2008). *Homo ludens: proeve eener bepaling van het spel-element der cultuur*. Amsterdam University Press.
- Humphrey, W. S. (1989). *Managing the software process*. Addison-Wesley Longman Publishing Co., Inc.
- Neto, J. C., Reinehr, S., e Malucelli, A. (2015). Processo de desenvolvimento para jogos eletrônicos educacionais: uma revisão de literatura. *Revista Brasileira de Informática na Educação*, 23(02):84.
- Neto, T. D. et al. (2021). Agile game development process: um processo para o desenvolvimento de jogos.
- Oliveira, S. R. (2006). Processo de software: princípios, ambientes e mecanismos de execução. *Exame de Qualificação do Programa de Doutorado. Centro de Informática–Universidade Federal de Pernambuco. Recife*. Disponível em www.ufpa.br/srbo.

- Petrillo, F., Pimenta, M., Trindade, F., e Dietrich, C. (2008). Houston, we have a problem... a survey of actual problems in computer games development. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 707–711.
- Politowski, C., De Vargas, D., Fontoura, L. M., e Foletto, A. A. (2016). Software engineering processes in game development: a survey about brazilian developers' experiences. In *Proceedings of the XIV Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*.
- Riot Games (2025). Concept art. Riot Games - Art Edu. Acessado em: 6 de julho de 2026.
- Schuytema, P. (2008). *Design de games: uma abordagem prática*. Cengage Learning.
- Scrum Alliance (2026). What is Scrum? Disponível em: <https://www.scrumalliance.org/about-scrum>. Acesso em: 04 fev. 2026.
- Shibata, N. N., de Castro, P. M., de Souza Albuquerque, D., Cruz, L. P., de Lima, M. S., Soares, B. Y. C., de Lima Alves, C., Quaresma, T. M., e Oliveira, S. R. B. (2025). Conectando conservação ambiental e cultura paraense por meio do jogo tekohá. *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 814–825.