

Práticas e Processos para uma Proposta de Programação Extrema em um Modo Jogável

Title: Practices and Processes for an Extreme Programming Proposal in a Playable Mode

Victor Travassos Sarinho¹

¹ Laboratório de Entretenimento Digital Aplicado (LEnDA)
Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana, BA – Brasil

vsarinho@uefs.br

Abstract. *There are several studies focused on identifying and defining gamification strategies in software development processes. These strategies also apply to agile methods, which can create a context of recognition and reward for the completion of activities in a software project. This paper presents a reinterpretation of the practices and stages of eXtreme Programming (XP) in a “playable mode”. As a result, a possible software development methodology based on agile gameplays for various system production stages is proposed.*

Keywords *Agile methods, eXtreme Programming, Practices and processes, Playable mode.*

Resumo. *Existem diversos trabalhos focados na identificação e definição de estratégias de gamificação em processos de desenvolvimento de software. Tais estratégias também se aplicam aos métodos ágeis, as quais conseguem criar um contexto de reconhecimento e recompensa para o cumprimento de atividades em um projeto de software. Este artigo apresenta uma releitura das práticas e etapas da eXtreme Programming (XP) em um “modo jogável”. Como resultado, tem-se uma possível metodologia de desenvolvimento de software baseada em gameplays ágeis para etapas de produção de sistemas diversos.*

Palavras-Chave *Métodos ágeis, Programação eXtrema, Práticas e processos, Modo jogável.*

1. Introdução

Quando se pensa na palavra “jogo”, a associação com conceitos de brincadeira, diversão ou até mesmo perda de tempo costuma ser realizada. No entanto, essa percepção de desperdício é um exemplo de como o tempo tende a remodelar antigas ideias e pensamentos, algo que se torna especialmente evidente com o conceito de gamificação, que busca aplicar mecânicas e dinâmicas bem-sucedidas dos jogos em situações do mundo real [Costa e Marchiori 2015].

De fato, é perceptível nos últimos anos a busca pelo engajamento de funcionários e colaboradores na tentativa de alcançar o estado de *flow* [Csikszentmihalyi et al. 2005] na realização de atividades alocadas para os mesmos. E para esta missão, a criação de sistemas de recompensa através da adoção de componentes de gamificação em processos do mundo real vem ganhando destaque, inclusive na Engenharia de Software (ES) [Pedreira et al. 2015].

Atualmente, existem trabalhos interessantes voltados para a identificação e definição de estratégias de gamificação em processos de desenvolvimento de software [García et al. 2017]. Neles, é perceptível a proposta de inclusão de *ranks*, *badges*, missões, entre outros componentes, no sentido de criar sistemas de recompensa capazes de aumentar o engajamento e a participação de pessoas na construção de sistemas diversos, uma atividade onde o fator humano ainda é de grande relevância no processo produtivo.

Por fator humano, metodologias ágeis surgiram nos últimos anos na tentativa de melhorar a qualidade dos softwares entregues, através do reconhecimento da importância e das limitações que os seres humanos possuem na produção de sistemas desejados [Al-Saqqa et al. 2020]. Neste sentido, práticas diversas vêm sendo propostas [Krancher 2020], destacando tanto a importância da distribuição do conhecimento da produção de sistemas entre seus colaboradores, como também da necessidade de se produzir pequenas *releases* de sistemas diretamente integradas com o cliente nos diferentes momentos de desenvolvimento e de utilização dos mesmos.

Fazendo um paralelo entre gamificação e o mundo ágil, é possível perceber que práticas de gamificação também se aplicam nos métodos ágeis, na medida em que ambos criam um contexto de reconhecimento e de recompensa para o cumprimento de atividades alocadas no processo. Contudo, existe um mundo de possibilidades no universo dos jogos que podem ser diretamente aplicadas ou servir de direcionamento no sentido de estimular o flow das pessoas envolvidas na produção de softwares desejados. Neste sentido, e fazendo uma alusão direta à *eXtreme Programming* (XP) [Anwer e Aftab 2017], este artigo apresenta uma releitura da execução de possíveis práticas e etapas da XP em um “modo jogável”, gerando como resultado uma possível metodologia de desenvolvimento de softwares com base em *gameplays* ágeis para etapas de produção de softwares.

2. Fundamentação Teórica

Metodologias de desenvolvimento de software visam aumentar a produtividade de equipes de desenvolvimento, encurtando o tempo de lançamento de soluções no mercado, reduzindo os custos de desenvolvimento e melhorando a satisfação do cliente [Lee e Chen 2023]. Para alcançar tais objetivos, o desenvolvimento ágil de software tem gradualmente gerado discussões públicas desde a década de 1990, o qual pode ser definido como um termo coletivo para trabalho colaborativo baseado em um conjunto de valores e princípios que se tornou uma abordagem padrão para a indústria de software [Naik e Jenkins 2019]. Trata-se de uma abordagem de desenvolvimento que surgiu como uma solução para a crise do software, onde a maioria dos projetos de desenvolvimento de software não conseguia atender aos requisitos dos usuários [Bera et al. 2023]. Neste sentido, metodologias ágeis defendem o planejamento adaptativo e o desenvolvimento evolutivo, compartilhando os mesmos valores do processo de software e incentivando respostas rápidas e flexíveis às mudanças por meio da entrega antecipada e da melhoria contínua dos sistemas produzidos [Sutherland e Sutherland 2014]. Mais ainda, o desenvolvimento ágil enfatiza o planejamento moderado, a cooperação orientada para as pessoas, a comunicação face a face, a auto-organização e gestão, e o desenvolvimento rápido dos sistemas desejados [Williams 2010].

Com relação ao *flow*, este apresenta a combinação de quatro elementos: motivação

intrínseca, máximo de concentração, estado emocional muito positivo e taxa de alto desempenho [Kamei 2014]. O estado de *flow* pode ser descrito como um estado mental que acontece quando uma pessoa realiza uma atividade e se sente totalmente absorvida em uma sensação de energia, prazer e foco total no que está fazendo [Csikszentmihalyi et al. 2005]. Ou seja, trata-se de um estado onde em sua essência é caracterizado pela imersão completa no que se faz, e por uma consequente perda do sentido de espaço e tempo. Assim, com foco e concentração máximos, toda a ação, todo o envolvimento e todo o pensamento fluem em sequência à ação até o final da realização da atividade em si [Csikszentmihalyi et al. 2005].

Apesar de desejado, o estado de flow constante tende a ser utópico e praticamente impossível de ser alcançado em cem por cento do tempo [Rossetti et al. 2022]. Neste sentido, uma relação mais próxima da realidade se apresenta com o estado de flow na intersecção entre o divertimento e o desafio, tendo o balanceamento de jogos sobre diferentes aspectos como um fator fundamental para que o estado de flow tenha maiores chances de ser obtido [Rossetti et al. 2022]. Em outras palavras, através de interfaces de jogos bem projetadas para seus respectivos usuários, tem-se que as mesmas podem ajudar a conseguir o balanceamento adequado de modo a fornecer as condições necessárias para fomentar o estado de flow [de Souza Teixeira e da Fonseca Ramos 2014].

Com relação ao conceito de gamificação, este apresenta mecânicas e dinâmicas identificadas em jogos capazes de serem aplicadas em diferentes processos, no sentido de criar um sistema de recompensa para captar a atenção e o interesse dos atores envolvidos. Já os jogos digitais fazem uso do conceito de entretenimento em interface altamente empáticas para a conquista da atenção e da interação do público alvo, através da aplicação de importantes elementos estéticos, tais como narrativa, desafio, socialização, dentre outros [Hunicke et al. 2004], no intuito de sensibilizar e conquistar seus jogadores para que eles permaneçam continuamente em um círculo mágico [Juul 2008] de jogabilidade. Neste sentido, através da aplicação de elementos de jogos e de conceitos de gamificação devidamente balanceados e contextualizados, pode-se proporcionar ao jogador os momentos necessários para que ele atinja o estado de flow, aumentando assim sua satisfação pessoal bem como sua produtividade no trabalho [Coutinho et al. 2021].

Iniciativas como *Planning Game* [Parsons 2014], Dojo de código-fonte [Luz e Neto 2012] e *Game-of-Games* (GoG) [Spil e Bruinsma 2016], [Sarinho 2020] mostram a possibilidade de se aplicar o uso de elementos de jogos digitais na produção de sistemas diversos, estendendo seu uso para além da simples inclusão de componentes de gamificação nos contextos dos processos de desenvolvimento aos quais foram agregados. Já do ponto de vista do uso de jogos em práticas ágeis, *Planning Poker* [Grenning 2002], *Extreme Hour* [Parsons 2014], “*XP Lego Game*” [Parsons 2014] e “*Red-Green-Go!*” [Embury et al. 2019] são exemplos interessantes que mostram como algumas práticas ágeis podem ser gamificadas de modo que a produção ágil de um sistema pode ser realizada de uma maneira divertida e empolgante. Ou seja, tendo como base tais iniciativas capazes de produzir sistemas jogando, tem-se a demonstração de que a integração de práticas de jogos e de gamificação às metodologias ágeis pode fomentar a colaboração e a criatividade entre os desenvolvedores, resultando em soluções inovadoras e eficazes dentro de um ambiente de trabalho dinâmico e motivador a partir de um brincar livre.

3. Definindo um Modo Jogável para a XP

Tendências recentes do estado da arte que adotam o desenvolvimento ágil vêm sendo exploradas na literatura, especialmente em computação na nuvem, big data e coordenação de equipes [Al-Saqqa et al. 2020]. Tratam-se de estudos que revelam novos métodos e práticas capazes de aumentar a eficiência e a qualidade dos processos de desenvolvimento de software em seus respectivos contextos. Outros trabalhos também apresentam como o design e a testagem de jogos de aprendizado voltados para a metodologia ágil podem ser usados como um meio para aumentar a compreensão dos estudantes sobre métodos ágeis, além de lhes dar a oportunidade de desenvolver suas habilidades criativas, organizacionais e analíticas [Parsons 2014].

No intuito de definir uma nova abordagem ágil dentro da perspectiva de jogos e gamificação, este trabalho busca a definição de metas jogáveis a serem aplicadas em um próximo nível de agilidade para os processos de software existentes. Para tal, tem-se a busca pela incorporação de ações do tipo “programar brincando”, “desafios de design” ou “competições de programação” em sessões de desenvolvimento “multiplayer colaborativas” na produção ágil de sistemas com base em abordagens de jogos existentes.

3.1. Práticas de Produção Jogáveis

Tomando como ponto de partida a metodologia XP, práticas de produção ágeis foram elencadas e aceitas com sucesso pela comunidade de ES, tais como programação em pares, refatoração de código, design simples, entre outras. Partindo destas práticas, elaborou-se uma releitura das mesmas dentro de situações e vivências proporcionadas por jogos em geral (Figura 1), tais como:

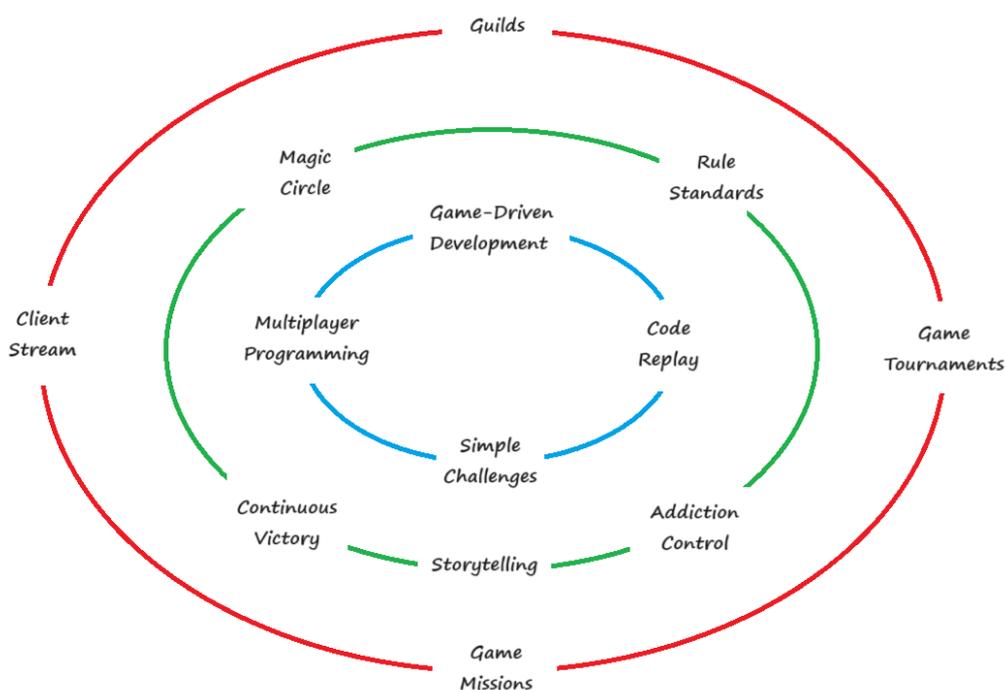


Figura 1. Práticas sugeridas para uma XP jogável. Fonte: Próprio Autor

- **Game-Driven Development:** Atividades a serem realizadas nas fases do processo de desenvolvimento devem ser realizadas através de mecânicas, dinâmicas e estéticas de jogos apropriadamente definidas. Desta forma, o ato de jogar, com início, meio e fim definidos, representado em uma experiência ficcional distante de qualquer problema ou risco existencial, capaz de prover ânsia e esperança tanto na possibilidade de perder como também na expectativa de vencer, de uma forma voluntária que gera um sentimento de recompensa, se torna o princípio básico a ser aplicado para desenvolver soluções para o domínio do problema alvo, buscando assim um nível diferenciado de imersão por parte dos colaboradores no processo de desenvolvimento em si.
- **Simple Challenges:** Desenvolvedores devem jogar partidas divididas em fases com pequenos desafios, onde as ações efetuadas pelos jogadores geram efeitos capazes de serem identificados no projeto. Considerando que ações de jogo serão usadas para a produção de sistemas desejados, é importante que a definição e a conquista destes pequenos desafios de jogos sejam orientadas para a produção de pequenos artefatos capazes de serem usáveis na montagem do software final alvo.
- **Multiplayer Programming:** O trabalho realizado por pessoas em processos de produção pode ser realizado de forma individual ou coletiva a depender das características do projeto. No caso dos jogos, partidas *single* ou *multiplayer* podem ser realizadas tanto no modo colaborativo como no modo competitivo, tanto em pequenos grupos como também no frenético *battle royale* [Choi e Kim 2018]. Com o acréscimo de *Head-Up Displays* (HUDs) que mostrem o desempenho de cada jogador e do seu grupo, tem-se modos alternativos interessantes de se realizar as atividades demandadas pelos projetos de uma forma jogável e gamificada para cada competidor ativo.
- **Code Replay:** Cada partida é única, e gera soluções únicas. Neste sentido, nada impede que os jogadores joguem novamente, só que desta vez executando as ações “corretas” para o projeto. Como resultado, múltiplos artefatos e *releases* com soluções parciais e completas podem ser geradas pelos colaboradores, dentro do ritmo contínuo de esforço que os jogadores empregam quando estão imersos em partidas de jogos desejados.
- **Rule Standards:** Assim como padrões de código podem ser adotados em projetos de software, padrões de jogabilidade e de ação-resposta podem ser definidos para cada partida de jogo a ser realizada em cada atividade do processo. Desta forma, se faz necessário definir um conjunto de regras de modo a impor aos jogadores um respeito a qualidade do projeto dentro das ações e respostas definidas e esperadas no jogo.
- **Addiction Control:** Jogadores quando gostam de um jogo, quando entram no estado de *flow* proporcionado pelos jogos, tendem a jogar o mesmo em um ritmo frenético. Neste sentido, deve-se ter o cuidado de se buscar evitar o “excesso de trabalho” em um determinado jogo, de modo a manter um ritmo sustentável para o projeto como um todo.
- **Continuous Victory:** Podemos dizer que cada fase ou partida finalizada em um jogo de um projeto já é uma vitória por si só. Os louros da vitória podem ficar para os ganhadores, mas o projeto em si é o grande vencedor uma vez que novas releases vitoriosas do projeto são obtidas como consequência de cada partida finalizada no jogo jogado.

- **Storytelling:** Cada cliente possui uma história a ser contada sobre um problema. Cada história possui um início, um meio e um fim. Cada história pode ser contada de diferentes maneiras, de modo a criar uma trama capaz de imergir os jogadores no sistema de regras e padrões propostos pelo processo de produção para o jogo em questão. Cabe a cada gerente de projeto definir em que ponto da história e em qual sistema de regras ele irá colocar seus colaboradores para iniciarem suas partidas.
- **Magic Circle:** Um jogo só acontece quando os jogadores se encontram no círculo mágico, e para isso acontecer eles precisam ter um entendimento coletivo do que está acontecendo no jogo em si. Neste momento, os jogadores ganham propriedade sobre as regras, as ações e os padrões a serem seguidos, de modo a se tornarem mestres em suas partidas com o passar do tempo.
- **Guilds:** Grupos formados por jogadores, cada qual com sua especialidade, no sentido de planejar suas ações e cumprir as metas escolhidas no jogo. Guildas de sucesso sabem da importância de cada membro da equipe, e cada membro sabe quando e como deve atuar na partida de modo a garantir o sucesso do grupo na realização das missões alocadas.
- **Game Tournaments:** Existem momentos na produção de um projeto onde se faz necessário definir quais partidas e em quais fases serão jogadas pelos desenvolvedores, as quais trarão os melhores resultados que irão avançar para a próxima etapa até a entrega final do projeto. Neste sentido, cabe aos gerentes, em comum acordo com os jogadores e as metas de projeto, decidirem quando os torneios serão definidos e realizados, gerando troféus e reconhecimento de toda a equipe para os melhores jogadores que seguiram as regras definidas no jogo, bem como geraram bons resultados para o projeto.
- **Client Stream:** Pessoas gostam de assistir e opinar sobre partidas de jogos e isso não seria diferente com os clientes. Neste sentido, é importante criar diferentes formas de participação dos clientes em tempo real com as partidas realizadas, seja assistindo e interagindo no desenrolar das partidas, seja emitindo comentários e sugestões relevantes que devem ser analisadas pelas equipes de jogo no seu devido tempo.
- **Game Missions:** Todo jogo possui objetivos, possui missões a serem cumpridas, sejam elas primárias ou secundárias. Cabe a equipe de projeto escolher quais e quando determinados desafios deverão ser cumpridos de modo a completar a grande missão que é completar o jogo como um todo e finalizar o projeto planejado pela equipe.

3.2. Processo de Produção Jogável

Usando as práticas previamente indicadas, levantou-se algumas atividades que podem ser realizadas com base nesses princípios, de modo a definir um possível ciclo de execução de etapas para um processo de desenvolvimento ágil “jogável” (Figura 2):

1. *Apresentação e adaptação da Storytelling a ser usada nos jogos (Game plan):* Neste momento, é efetuado um planejamento de quais jogos serão usados no projeto a ser desenvolvido, modelando como os mesmos irão se apropriar do *Storytelling* do cliente para gerar resultados nas *Continuous Victories* futuras. Também é decidido como os jogadores irão tomar conhecimento dos problemas

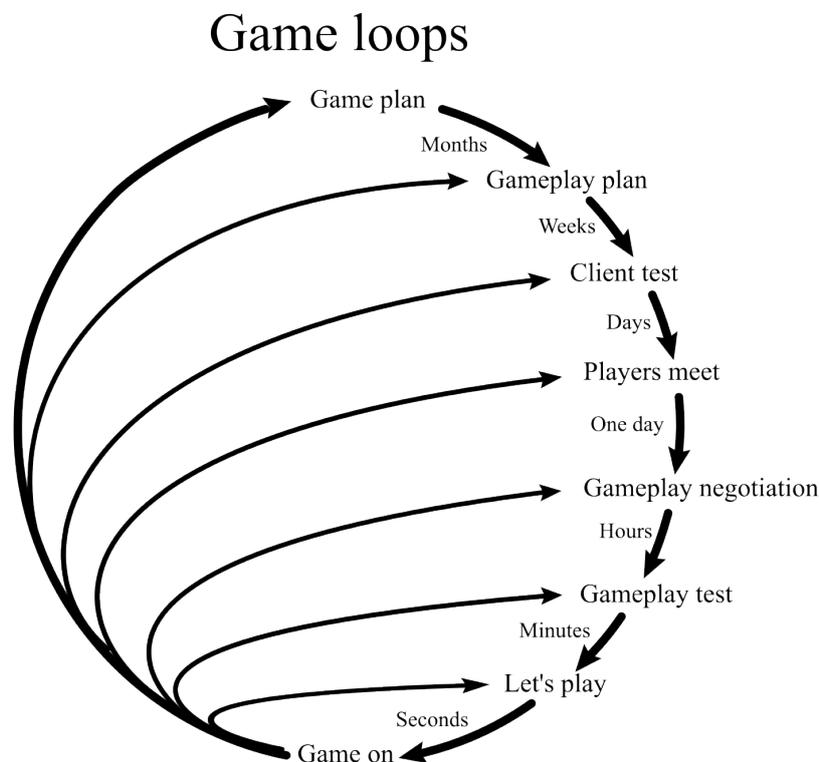


Figura 2. Etapas de processo sugeridas para uma XP jogável. Fonte: Próprio Autor

do cliente e dos seus resultados para o projeto em si de uma forma adaptada para cada jogo em questão.

2. *Definição de Simple Challenges conforme as Rule Standards que serão usadas nos jogos escolhidos para o projeto (Gameplay plan):* A depender da missão, do contexto do jogo e dos jogadores na partida, alguns desafios e regras podem ser aplicados ou não para cada jogo, tais como: número de encaixes possíveis de palavras não ambíguas para a criação de *user stories*; tempo de resposta para escolher uma regra de mutação a ser aplicada em testes de software apresentados em um ritmo “frenético”; ou escolha de ações que o jogador pode realizar para a simplificação ou composição de interfaces conforme a User eXperience (UX) em um design de interface atômico [Frost 2016], [Odushegun 2023]. Vale salientar que possíveis missões tutoriais também poderão ser definidas nessa etapa, melhorando assim o desempenho e a imersão inicial dos jogadores nas primeiras partidas de cada jogo.
3. *Transmissão e acompanhamento contínuo dos comentários na Client Stream (Client test):* Trata-se de uma etapa continuamente realizada pelos clientes no projeto, através do acompanhamento dos HUDs e do status de cada jogo, bem como da tomada de decisão junto com a equipe de desenvolvimento de possíveis alterações no modo de jogo como um todo, uma vez que os clientes representam na prática os mestres de cada partida jogada no processo de desenvolvimento.
4. *Atualização da Storytelling e do status das Game Missions no jogo (Players*

meet): Com a repetida execução de partidas dos jogos escolhidos, de modo a realizar suas *Simple Challenges* conforme as *Rule Standards* definidas, novos status de jogos são regularmente obtidos. Nesses status, novas regras e desafios podem ser definidos e liberados para os jogadores, bem como novas atualizações de *Storytelling* dos jogos, de modo a garantir variabilidade e surpresa para os jogadores envolvidos. Vale salientar que como vários jogos distintos podem ser usados na produção do sistema final, o status de cada um deles após jogar suas partidas pode influenciar nos demais jogos que se encontram com poucas partidas jogadas, ou até mesmo sem nenhuma partida jogada.

5. *Escolha da Guild e da Game Mission a ser jogada (Gameplay negotiation)*: O desenvolvimento de um sistema possui várias missões em paralelo que podem ser realizadas em cada jogo disponibilizado, tais como produção de telas, programação de códigos, escrita de testes, etc. Cabe aos jogadores escolherem seus parceiros e quais missões iniciais irão se aventurar.
6. *Code Replay contínuo com monitoração do Addiction Control (Gameplay test)*: Com a definição de *Simple Challenges* conforme as *Rule Standards* definidas, múltiplas partidas são repetidamente jogadas pelos jogadores, de modo a gerar múltiplos artefatos capazes de serem usados no projeto do sistema. Neste sentido, se faz necessário definir e realizar o monitoramento do ritmo de jogo de cada jogador, de modo a não gerar problemas nos mesmos devido ao “excesso de trabalho” pelo ato contínuo e viciante que os jogos podem proporcionar aos mesmos.
7. *Busca do Magic Circle para os jogadores envolvidos conforme o modo de Multiplayer Programming escolhido (Let’s play)*: Seja colaborativo ou competitivo, seja individual ou em grupo, cabe ao gerente e as equipes decidirem como querem jogar cada jogo proposto, de modo a garantir momentos divertidos e agradáveis de produção de sistemas com os mesmos.
8. *Registro das Continuous Victory de cada partida jogada e realização de Game Tournaments (Game on)*: Para cada partida jogada, artefatos gerados do sistema em produção precisam ser documentados e organizados em protótipos e releases do sistema. Torneios internos também podem ser realizados neste momento no projeto, a depender da urgência ou do atual ritmo de trabalho da equipe para finalização de releases a serem entregues para os clientes.
9. *Repetição dos passos a partir do passo inicial até a finalização das Game Missions elencadas (Game loops)*: Com base no desempenho dos jogadores nas partidas, nos comentários dos clientes e no avanço do desenvolvimento do projeto, cabe a equipe de desenvolvimento decidir quais readaptações serão necessárias para fins de se obter o projeto final desejado com sucesso.

3.3. Vantagens e Desvantagens em um XP Jogável

A metodologia XP tradicional possui diversas vantagens [Al-Saqqa et al. 2020], [Abrahamsson et al. 2017], tais como:

- Desenvolvimento incremental: suportado através de pequenas e frequentes liberações do sistema.
- Melhoria da produtividade: com um rápido feedback para muitas versões que podem ser construídas a cada dia e que só são aceitas se passarem nos testes.

- Simplicidade: mantida com a refatoração constante do código.
- Melhoria da qualidade: através do desenvolvimento de testes automatizados antes de integrar uma funcionalidade no sistema.

Por outro lado, o XP também sofre das seguintes limitações [Al-Saqqa et al. 2020], [Abrahamsson et al. 2017]:

- Falta de suporte a equipes distribuídas, pois foca na comunidade e na co-localização.
- Desenvolvimento orientado por testes requer treinamento técnico adicional para os membros da equipe envolvidos.
- Documentação informal para manter os detalhes importantes do projeto.
- Envolvimento real do cliente é eficaz, mas é estressante e demorado.

Tendo como base as práticas e o ciclo de vida indicado para a execução de uma XP jogável, é possível perceber que as vantagens da XP tradicional conseguem ser mantidas, bem como algumas desvantagens identificadas conseguem ser solucionadas. Com relação as vantagens, o desenvolvimento no XP jogável permanece incremental, com a adaptação da *Storytelling* do cliente em *Simple Challenges* e *Game Missions* a serem aplicadas. Os ganhos de produtividade, simplicidade e qualidade também são passíveis de serem obtidos, através do *Code Replay* em uma *Continuous Victory* que gera resultados baseados em *Rule Standards* tendo seu *Addiction Control* aplicado.

Já com relação as desvantagens da XP, o problema da falta de suporte a equipes distribuídas consegue ser resolvido com a aplicação de *Game Tournaments* apresentando *Game Missions* a serem resolvidas pelas *Guilds* formadas em seu trabalho de *Multiplayer Programming*. Missões tutoriais podem ser definidas na etapa de *Gameplay plan* de modo a garantir o treinamento adicional de uma maneira rápida e divertida. A manutenção de uma documentação formal pode ser obtida com os resultados e feedbacks das partidas realizadas, a qual pode ser definida na fase de *Gameplay plan*. Por fim, o envolvimento do cliente pode se tornar dinâmico e divertido, sendo realizado na transmissão e no acompanhamento contínuo dos comentários pela *Client Stream* na fase de *Client test* do projeto.

4. Conclusões e Trabalhos Futuros

Fatores humanos envolvidos na produção de sistemas diversos representam hoje em dia um desafio amplamente considerado nas metodologias ágeis existentes. Gamificação de processos realiza a implantação de abstrações jogáveis nas diferentes fases de produção de um sistema, permitindo a monitoração em tempo real do estado corrente de um projeto e de seus colaboradores. Com a aplicação de um modo jogável em metodologias de desenvolvimento de software, tem-se a possibilidade de estender o uso desta gamificação para um outro nível, através da inclusão de demais elementos de jogos existentes de modo a garantir uma abordagem jogável para o desenvolvimento de projetos desejados.

Assim, este trabalho apresentou a preocupação e a proposição de uma integração mais abrangente de elementos de jogos em processos de software, de modo a criar contextos imersivos e balanceados capazes de motivar de forma espontânea para um brincar livre membros de equipes em um processo jogável, bem como evitar a geração de ambientes nocivos e altamente competitivos para os seus desenvolvedores. Como

resultado, a aplicação de um modo jogável na metodologia XP defende que etapas de um processo de ES também podem ser totalmente realizadas por jogos, ampliando assim as possibilidades de entretenimento e de imersão durante o processo de desenvolvimento. O desafio neste caso consiste em pensar nas atividades da ES sendo realizadas a partir de mecânicas, dinâmicas e estéticas de jogos embutidas, a exemplo da incorporação de Game-of-Games (GoG) na produção de jogos digitais desejados, algo que pode ser definido como uma oportunidade interessante de investigação do uso integrado de conceitos de jogos na ES para a produção de softwares de domínios específicos.

Por fim, como trabalhos futuros, pretende-se investigar e repensar a possibilidade de aplicação de um modo jogável para outros processos de desenvolvimento, no sentido de entender até que ponto eles podem ser viáveis na execução dos mesmos em um brincar livre e espontâneo. A avaliação de ganhos de produtividade com o aumento da jogabilidade na produção de sistemas, bem como a viabilidade da inserção de interfaces de manipulação direta empáticas e gamificadas dentro das etapas de produção de sistemas, também serão investigadas e construídas em um futuro próximo.

Referências

- Abrahamsson, P., Salo, O., Ronkainen, J., e Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.
- Al-Saqqa, S., Sawalha, S., e AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11).
- Anwer, F. e Aftab, S. (2017). Latest customizations of xp: A systematic literature review. *International Journal of Modern Education and Computer Science*, 9(12):26.
- Bera, P., Wautelet, Y., e Poels, G. (2023). On the use of chatgpt to support agile software development. In *The Second International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE 2023) co-located with the 35th International Conference on Advanced Information Systems Engineering (CAiSE 2023)*, volume 3414, pages 1–9. CEUR.
- Choi, G. e Kim, M. (2018). Battle royale game: In search of a new game genre. *International Journal of Culture Technology (IJCT)*, 2(2):5.
- Costa, A. C. S. e Marchiori, P. Z. (2015). Gamificação, elementos de jogos e estratégia: uma matriz de referência. *InCID: Revista de Ciência da Informação e Documentação*, 6(2):44–65.
- Coutinho, L. L., Vieira, I. D. P., e de Souza, J. C. P. (2021). O estado de flow na alta performance de líderes organizacionais the flow state in high performance of organizational leaders. *Brazilian Journal of Development*, 7(8):83333–83348.
- Csikszentmihalyi, M., Abuhamdeh, S., e Nakamura, J. (2005). Flow. *Handbook of competence and motivation*, pages 598–608.
- de Souza Teixeira, E. A. e da Fonseca Ramos, F. (2014). Interações e literacias: notas sobre o design de interfaces e a experiência de uso. *Ciência da Informação*, 43(3).

- Embury, S. M., Borizanov, M., e Jay, C. (2019). Red-green-go! a self-organising game for teaching test-driven development. *Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom*, pages 415–441.
- Frost, B. (2016). *Atomic design*. Brad Frost Pittsburgh.
- García, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., e Penabad, M. (2017). A framework for gamification in software engineering. *Journal of Systems and Software*, 132:21–40.
- Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*, 3:22–23.
- Hunicke, R., LeBlanc, M., Zubek, R., et al. (2004). Mda: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, page 1722. San Jose, CA.
- Juul, J. (2008). The magic circle and the puzzle piece.
- Kamei, H. (2014). Flow e psicologia positiva: estado de fluxo, motivação e alto desempenho. *Goiânia: IBC*.
- Krancher, O. (2020). Agile software development practices and success in outsourced projects: The moderating role of requirements risk. In *Agile Processes in Software Engineering and Extreme Programming: 21st International Conference on Agile Software Development, XP 2020, Copenhagen, Denmark, June 8–12, 2020, Proceedings 21*, pages 56–72. Springer.
- Lee, W.-T. e Chen, C.-H. (2023). Agile software development and reuse approach with scrum and software product line engineering. *Electronics*, 12(15):3291.
- Luz, R. B. e Neto, A. (2012). Usando dojos de programação para o ensino de desenvolvimento dirigido por testes. *Anais do Simpósio Brasileiro de Informática na Educação*, 23(1).
- Naik, N. e Jenkins, P. (2019). Relax, it's a game: Utilising gamification in learning agile scrum software development. In *2019 IEEE Conference on Games (CoG)*, pages 1–4. IEEE.
- Odushegun, L. (2023). Aesthetic semantics: Affect rating of atomic visual web aesthetics for use in affective user experience design. *International Journal of Human-Computer Studies*, 171:102978.
- Parsons, D. (2014). Creating game-like activities in agile software engineering education. In *Proceedings of the Australasian Software Engineering Conference, Education Track, Sydney, Australia*.
- Pedreira, O., García, F., Brisaboa, N., e Piattini, M. (2015). Gamification in software engineering—a systematic mapping. *Information and software technology*, 57:157–168.
- Rossetti, R. D. et al. (2022). A influência do medo no flow.
- Sarinho, V. T. (2020). Applying user stories as game elements and interactions in a game of games design proposal. *SBC-Proceedings of SBGames*.
- Spil, T. A. e Bruinsma, G. (2016). Designing serious games with the game of games. In *Proceedings of the European Conference on Games-based Learning*, pages 634–643.

Sutherland, J. e Sutherland, J. (2014). *Scrum: the art of doing twice the work in half the time*. Crown Currency.

Williams, L. (2010). Agile software development methodologies and practices. In *Advances in computers*, volume 80, pages 1–44. Elsevier.