



**PROPOSTA DE UM FRAMEWORK ÁGIL PARA GESTÃO DE PROJETOS
DE DESENVOLVIMENTO DE JOGOS ELETRÔNICOS**
*PROPOSAL OF AN AGILE FRAMEWORK FOR PROJECT MANAGEMENT OF
ELECTRONIC GAME DEVELOPMENT*

André Luiz da Silva¹

Diogo Tavares da Silva²

RESUMO: Historicamente a indústria de jogos eletrônicos tem adotado métodos tradicionais de engenharia de software, como o método em cascata. Essas abordagens possuem a desvantagem de serem pouco dinâmicas e basicamente reativas, desestimulando a proatividade, o que acaba por contribuir para que um grande número de projetos falhe. Neste contexto, propõem-se um framework baseado em metodologias ágeis para o desenvolvimento de jogos eletrônicos, intitulado improved Game Agile Methods Applied (iGAMA), assim como levanta-se um conjunto de ferramentas CASE gratuitas de suporte para sua utilização. Por meio de levantamento bibliográfico e análise de trabalho correlatos, adquiriu-se o embasamento necessário para estudar os principais problemas da indústria de jogos e com base nestes problemas elaborar um estudo de caso para a aplicação do framework ágil proposto de modo a verificar sua eficácia em contorná-los. De acordo com o estudo realizado, verificou-se que a aplicação do iGAMA proporciona resultados positivos, concluindo-se que sua aplicação é capaz de mitigar os problemas enfrentados pela indústria de jogos, como o escopo irreal e a falta de motivação.

Palavras-chave: Desenvolvimento de Jogos Eletrônicos; Engenharia de Software; Metodologias Ágeis

¹ Bacharel em Sistema de Informações, UNIBARRETOS, 2016.

² Mestre em Ciência da Computação, UNESP, 2015.

ABSTRACT: Historically the gaming industry has adopted traditional methods of software engineering, such as the cascading method. These approaches have the disadvantage of being less dynamic and basically reactive, discouraging proactivity, which ends up contributing to a large number of projects that fail. In this context, a framework based on agile methodologies for the development of electronic games, called improved Game Agile Methods Applied (iGAMA), is proposed, as well as a set of free CASE support tools for its use. Through a bibliographical survey and related work analysis, we acquired the necessary basis to study the main problems of the gaming industry and based on these problems to elaborate a case study for the application of the proposed agile framework in order to verify its effectiveness in solving them. According to the study, it was verified that the application of iGAMA provides positive results, concluding that its application is able to mitigate the problems faced by the gaming industry, such as the unreal scope and the lack of motivation.

Keywords: Development of Electronic Games; Software Engineering; Agile Methods

1 Introdução

A indústria de jogos eletrônicos é uma das mais poderosas do setor de entretenimento, com faturamento de bilhões de dólares e gerando trilhões de horas de divertimento (PWC, 2015). No entanto, a multidisciplinaridade presente no processo de desenvolvimento de jogos decorrente da inclusão de diversos fatores de produção como artes gráficas e sonoras, construção de enredo, jogabilidade, sistemas de controle, inteligência artificial e fatores humanos, cria um cenário que incrementa ainda mais complexidade de elaboração.

Historicamente, a indústria de jogos tem adotado processos tradicionais de engenharia de software, como o método em cascata (waterfall). Essas abordagens possuem a desvantagem de serem pouco dinâmicas e basicamente reativas, não estimulando a proatividade, o que acaba por contribuir para que um grande número de projetos falhem. Em busca de contornar tal problema, um grande número de metodologias de

desenvolvimento de software foram propostas, sendo que as metodologias ágeis foram as que melhor se sobressaíram (SABBAGH, 2013).

Uma característica das metodologias ágeis é que elas são adaptativas ao invés de preditivas (LIBARDI; BARBOSA, 2010). Elas se adaptam ao desenvolvimento do projeto, ao invés de tentarem analisar de forma prévia tudo o que pode ou não acontecer no decorrer do desenvolvimento. Outro ponto positivo dos métodos ágeis são as entregas constantes de incrementos operacionais do software (LIBARDI; BARBOSA, 2010). Isso permite uma redução do tempo de espera do cliente para ver o software funcionando e perceber se algo não está como esperado, reduzindo o tempo de detecção de erros em requisitos.

Com relação a aplicação de métodos ágeis no desenvolvimento do nicho específico de jogos eletrônicos, destaca-se um estudo no qual se apresenta um conjunto de práticas ágeis agrupadas em forma de metodologia específica, chamada Game Agile Methods Applied (GAMA) (PETRILLO, 2008). A abordagem utilizada no GAMA consiste em conciliar práticas e conceitos de quatro métodos ágeis que são respectivamente o Desenvolvimento Enxuto de Software para definir a filosofia de trabalho, o Scrum para o gerenciamento das atividades e por fim a Modelagem Ágil e Programação Extrema (XP) para a execução do projeto. Apresenta-se, na Figura 1, a forma como os métodos ágeis que formam o GAMA se conciliam.

Figura 1 - Diagrama do agrupamento dos métodos ágeis que compõem o GAMA.



Fonte: adaptado de Petrillo (2008).

Durante uma análise da aplicabilidade do GAMA notou-se que a metodologia proposta por PETRILLO (2008), embora coerente, não é muito clara e objetiva na definição das ações necessárias para a gestão ágil de projetos de todos os gêneros e tipos de jogos eletrônicos e que a metodologia específica não abrange todas as necessidades existentes na indústria de jogos.

Decorrente deste contexto, o objetivo deste trabalho é apresentar uma proposta de framework intitulado iGAMA (improved Game Agile Methods Applied), baseado no GAMA, que além de implementar seus principais princípios propõe deixar a metodologia mais clara e apropriada para a gestão de projetos de jogos eletrônicos com o intuito de preservar os processos criativos e a diversão do jogo. Juntamente à organização proposta é levantado um conjunto de ferramentas CASE (Computer-Aided Software Engineering) gratuitas especializadas aos requisitos e necessidades da indústria de jogos para apoiar sua utilização.

A fim de verificar a eficiência do framework proposto, apresentam-se ainda os resultados de sua aplicação em um estudo de caso, baseando-se no desenvolvimento de um jogo eletrônico de plataforma.

O artigo está organizado com as seguintes seções, além desta introdução. Na Seção 2 apresenta-se uma análise dos principais métodos utilizados no desenvolvimento de jogos eletrônicos. Em seguida, na Seção 3, define-se e descreve-se o framework proposto neste artigo bem como apresenta-se um conjunto de ferramentas CASE para auxiliar sua aplicação. Após, na Seção 4 apresenta-se um estudo de caso com o intuito de avaliar a eficácia do framework proposto. Posteriormente, na Seção 5, demonstram-se e discutem-se os resultados obtidos no estudo de caso. Por fim, na Seção 6, apresentam-se as conclusões dos estudos realizados e perspectivas para trabalhos futuros.

2 Análise dos métodos de desenvolvimento de jogos eletrônicos

A indústria de jogos tem adotado diferentes abordagens de engenharia de software, contudo o método tradicional em cascata é o mais utilizado. Estudos sobre a aplicação de outros métodos de desenvolvimento têm sido propostos, sendo que o estudo sobre a aplicação de métodos ágeis apresentado por PETRILLO (2008) é o mais notório.

Apresenta-se, nas próximas subseções, uma análise dos métodos tradicional e ágil de desenvolvimento de jogos eletrônicos.

2.1 Metodologia tradicional de desenvolvimento de jogos e problemas decorrentes de sua utilização

De acordo com NOVAK (2010, p. 404) as fases que compõe o processo tradicional de desenvolvimento de jogos eletrônicos são: conceito, pré-produção, prototipação, produção, alfa, beta, ouro e pós-produção. Cada fase envolve certos membros da equipe de desenvolvimento do jogo, concentra-se em objetivos específicos e devem ser seguidas estritamente em sequência.

Nas fases de conceito e pré-produção o objetivo principal é decidir em que consiste o jogo, identificar um público-alvo, avaliar os recursos da empresa, identificar um conceito que seja atraente aos desenvolvedores e tenha um mercado potencial, e elaborar o guia de estilo da arte do jogo e o plano de produção.

Após definir as principais informações sobre o jogo, seu público alvo e características, ocorre a fase de prototipação em que é elaborado um protótipo do jogo seguida das fases de produção, alfa e beta nas quais o jogo é codificado e testado.

Por fim, nas fases ouro e de pós-produção o jogo é finalizado e enviado ao distribuidor e alguns problemas ainda existentes são solucionados.

Com relação aos problemas do desenvolvimento tradicional, ao analisar vários post-mortems (documentos de pós-produção), FLOOD (2003) revela que sempre se encontram os mesmos problemas: projeto entregue atrasado, funcionalidades diferentes das que foram originalmente projetadas, muitas horas de retrabalho e produto final com muitos defeitos.

Há muitas ideias diferentes sobre o motivo dos projetos de jogos falharem, porém, a mais notória é a contextualização feita por SUNDSTRÖM (2012). O autor aponta como principais pontos críticos as fases que antecedem a produção (conceito, pré-produção e prototipação). É nessas fases que a equipe é formada, a gestão planeja o projeto e se prepara para começar a construir o jogo, porém isso raramente é feito seguindo todas as etapas. Muitas equipes invertem processos ou pulam etapas para lançar antecipadamente o jogo no mercado e muitas vezes quebram o planejamento inicial, o que contribui para a falha de seus projetos.

De acordo com SUNDSTRÖM (2012), a fase de produção muitas vezes começa paralelamente à fase de pré-produção, o que significa que a equipe de desenvolvimento se baseia em uma documentação incompleta, em que comumente há falta de análise das

exigências do projeto e público-alvo. Isto leva a um planejamento não confiável que pode trazer confusão aos membros da equipe de produção.

2.2 Game Agile Methods Applied

Como enfatizado por PETRILLO (2008), muito dos problemas encontrados no desenvolvimento de jogos estão relacionados com o processo tradicional de desenvolvimento. Diante desta perspectiva, o autor analisou quais práticas das principais metodologias ágeis (Desenvolvimento Enxuto de Software, Programação Extrema (XP), Modelagem Ágil e Scrum) poderiam eliminar ou minimizar estes problemas.

Primeiramente, a filosofia de trabalho deve seguir os conceitos do Desenvolvimento Enxuto de Software que é baseado em princípios e práticas do Sistema de Produção da Toyota e apresenta um conjunto de ferramentas baseadas no pensamento enxuto que são úteis na compreensão do desenvolvimento ágil de software (STEFFEN, 2011).

Já o gerenciamento das atividades deve ser realizado por meio do Scrum, nele compreende-se que as práticas necessárias para cada projeto são muito específicas, logo não podem ser prescritas, pois não existe uma solução ou um conjunto de soluções para todas as situações.

Por fim, a execução do projeto é apoiada pelos métodos de Programação Extrema e Modelagem Ágil. Na Programação Extrema, os esforços da equipe de desenvolvimento são concentrados para gerar resultados rapidamente alinhados às necessidades de seus usuários, de modo a simplificar e organizar as tarefas por meio de técnicas eficazes, eliminando atividades redundantes. A Modelagem Ágil, por sua vez, é uma metodologia baseada na prática para modelagem e documentação eficaz de sistemas de software e é composta por uma coleção de valores, princípios e práticas derivados da Programação Extrema e que podem ser aplicadas no desenvolvimento de maneira leve e eficaz.

De acordo com os resultados obtidos no estudo feito por PETRILLO (2008), o GAMA se mostrou apropriado para o desenvolvimento de jogos, sendo que o projeto no qual o mesmo foi utilizado produziu resultados significativamente melhores do que o projeto no qual o processo tradicional foi utilizado.

3 improved Game Methods Agile Applied

Com o intuito de incrementar a organização da metodologia proposta no GAMA PETRILLO (2008), propõe-se a aplicação dos métodos ágeis que compõe a metodologia por meio de etapas. Estas etapas foram estendidas (melhoradas) para incorporar aprimoramentos ao GAMA, a fim de deixar a metodologia mais clara e apropriada para a gestão de projetos de jogos eletrônicos com o intuito de preservar os processos criativos e a diversão do jogo, bem como mitigar os detalhes apenas quando há informação necessária para isso. Para esta abordagem elaborada em forma de um framework se deu o nome de improved Game Agile Methods Applied (iGAMA).

Apresenta-se, na Figura 2, um diagrama de atividades das etapas estendidas e aprimoradas (destacadas) no iGAMA. Uma descrição de todo o processo é apresentada a seguir.

A implementação do iGAMA é dividida em duas fases principais denominadas fase exploratória e fase de execução. A fase exploratória compreende as etapas de definição dos conceitos básicos do jogo enquanto a fase de execução compreende as etapas de planejamento e execução das tarefas para a construção do jogo.

Analisando-se a fase exploratória, pode-se enumerar o seguinte fluxo de atividades:

1. Inicialmente, elabora-se o conceito do jogo seguindo as premissas do Desenvolvimento Enxuto de Software;
2. Após elaborar o conceito do jogo, criam-se os diagramas de casos de uso se utilizando dos princípios e valores da Modelagem Ágil. Em seguida, preenche-se o Product Backlog com as funcionalidades desejadas para o jogo baseando-se nos diagramas de casos de uso;
3. Com o preenchimento do Product Backlog finalizado, executa-se o planejamento de desenvolvimento do jogo por meio de uma reunião na qual são estabelecidas as diretrizes do projeto;
4. Após a conclusão do planejamento, para melhor descrever as funcionalidades desejadas e melhor entender os itens que compõe o Product Backlog, sugere-se a elaboração de cartões contendo as histórias de usuário, sendo que cada cartão deve

representar um caso de uso, aconselha-se também a criação de casos de teste e a definição das condições de satisfação para cada item de Backlog;

5. Com os cartões de histórias de usuário elaborados, segue-se para a etapa de planejamento do Release em que se elabora o plano de Release que é o documento que contém o nome e o cálculo de estimativas dos prazos do Release;
6. A fim de contornar possíveis erros e contratemplos, repete-se o planejamento até que todas as histórias tenham sido atendidas.

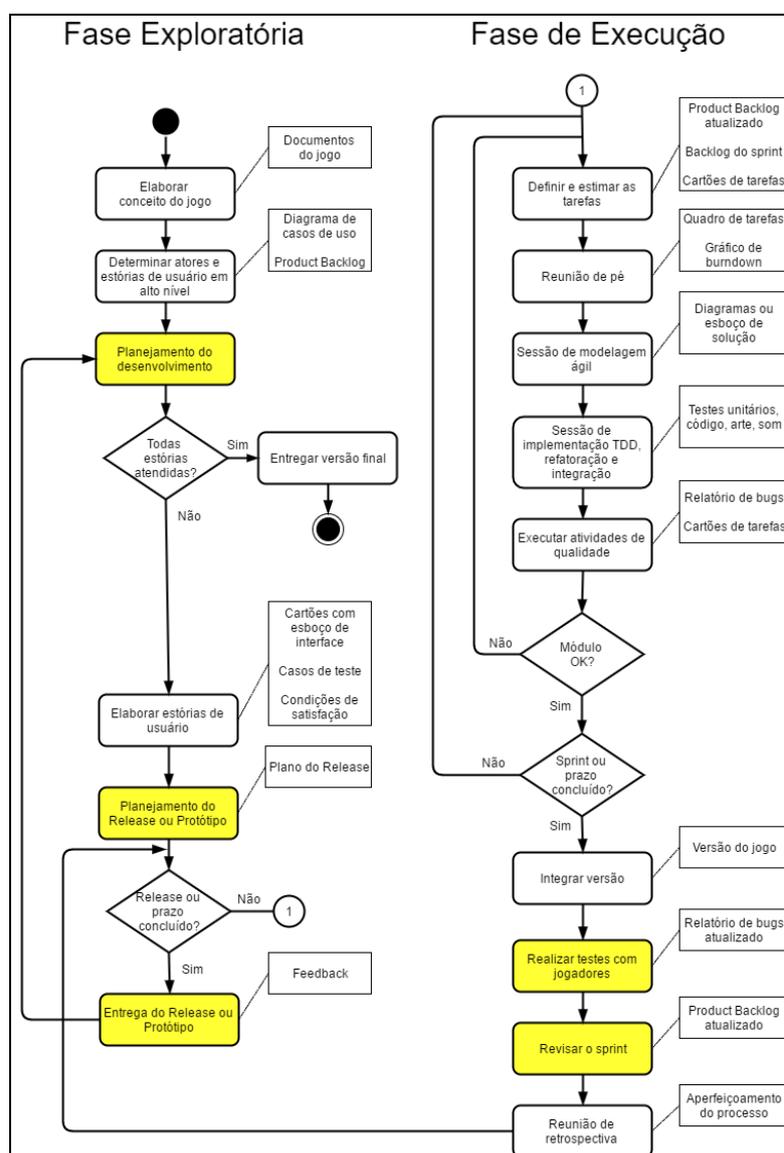
Após concluir a fase de planejamento, inicia-se a fase de execução. Nesta fase pode-se enumerar o seguinte fluxo de atividades:

1. Inicialmente, elabora-se o Backlog do Sprint e posteriormente definem-se e estimam-se as tarefas com base no Product Backlog e no Backlog do Sprint. Em seguida, com base nas tarefas, criam-se os cartões de tarefas;
2. Após criar os cartões de tarefas, inicia-se o Sprint com uma reunião entre as partes interessadas e os desenvolvedores a fim de elaborar o planejamento do quadro de tarefas do Sprint e analisar o desempenho do mesmo;
3. Posteriormente a reunião, inicia-se a sessão de modelagem utilizando os princípios e valores da Modelagem Ágil. Nesta etapa atualizam-se os diagramas de caso de uso e elaboram-se os esboços de soluções de problemas;
4. Com a modelagem realizada, inicia-se a implementação do software do jogo por meio da aplicação dos princípios e práticas da Programação Extrema;
5. Seguindo os princípios do desenvolvimento orientado a testes, ainda durante a etapa de implementação, executa-se testes a fim de detectar possíveis problemas de construção;
6. Após os testes, realiza-se a refatoração e a integração do código-fonte elaborado com os demais códigos-fonte desenvolvidos durante o Sprint;
7. A fim de manter a qualidade do produto, elaboram-se relatórios e cartões de tarefas com base nos problemas constatados durante os testes a fim de corrigi-los;
8. Com o módulo finalizado, conclui-se o Sprint. Em seguida, executa-se a integração do código-fonte desenvolvido no Sprint com o código-fonte do projeto principal;
9. Para assegurar a qualidade do desenvolvimento, realizam-se testes liberando versões prévias do jogo para que pessoas possam jogá-lo antecipadamente a fim de relatar possíveis problemas. Com isso, os relatórios de problemas e as tarefas necessárias para corrigir os problemas são atualizadas com base nas informações relatadas;

10. Após os testes com jogadores, realiza-se a revisão do Sprint a fim de atualizar o Product Backlog, além de uma reunião de retrospectiva para aperfeiçoar o processo de desenvolvimento;
11. Por fim, novos Sprints e Releases são executados ao longo do trabalho até que seus prazos terminem ou o escopo seja totalmente satisfeito.

É importante ressaltar que cada projeto possui suas próprias particularidades e o planejamento e execução podem ser adaptados conforme a necessidade de cada projeto, de acordo com as premissas do Desenvolvimento Enxuto de Software.

Figura 2 – Diagrama de Atividade das Etapas iGAMA



Fonte: adaptado pelos autores.

3.1 Ferramentas CASE de apoio ao iGAMA

Para apoiar a aplicação do iGAMA é proposto um conjunto de ferramentas CASE. Para esta escolha foram considerados principalmente os critérios de gratuidade, disponibilidade frente a ferramentas semelhantes e popularidade.

Deste modo as seguintes ferramentas são propostas:

- Gestão do código-fonte: para gestão do código-fonte é utilizada a aplicação web GitHub (<https://github.com>) que é voltada para hospedagem de projetos que utilizam o Git, que é um sistema de controle de versão distribuído e um sistema de gerenciamento de código-fonte, com ênfase em velocidade. A ferramenta \textit{Github} funciona como uma espécie de rede social que permite que seus usuários compartilhem experiências de desenvolvimento e gestão de projetos;
- Gestão do projeto: para gestão do Scrum no projeto é utilizada a aplicação web ScrumHalf (<http://myscrumhalf.com>) que contém uma interface amigável e relativamente simples com uma grande quantidade de funções, por exemplo, seu quadro de tarefas virtual que facilita o acompanhamento do desenvolvimento do trabalho, ainda por meio desta aplicação é possível obter uma gama de relatórios fundamentais para analisar o desempenho do projeto. Esta aplicação é utilizada por grandes organizações como, por exemplo, o Ministério do Planejamento, Desenvolvimento e Gestão do Brasil, o SEBRAE (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas) e a Coppe-UFRJ (Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro);
- Testes: para a execução de testes é utilizado o serviço web Travis CI (<https://travis-ci.org>) que é aplicado juntamente com o GitHub para a execução de testes automatizados, este serviço também possibilita o controle de execução de testes de software, a comparação dos resultados esperados com os resultados reais, a configuração das pré-condições de teste e funções de controle e relatórios de teste.

4 Verificação do iGAMA por meio de estudo de caso

Para assegurar a eficácia da utilização de métodos ágeis no desenvolvimento de jogos eletrônicos foi elaborado um estudo de caso no qual o framework iGAMA juntamente com seu conjunto de ferramentas CASE foram aplicados.

Para contemplar o objetivo do estudo de caso, como projeto-alvo para aplicação dos métodos ágeis, foi escolhido a implementação de um jogo eletrônico de plataforma utilizando a linguagem de programação Python versão 2.7.12 juntamente com a biblioteca multiplataforma Pygame versão 1.9.1. A seleção do estilo de jogo se deve principalmente a fatores como jogabilidade e popularidade do estilo em plataforma. A seleção da linguagem de programação, por sua vez, se deve a fatores como gratuidade, popularidade e capacidade de desenvolvimento ágil de aplicações.

O estudo de caso foi conduzido entre os meses de julho e setembro do ano de 2016 e buscou realizar uma análise empírica da aplicação do iGAMA no desenvolvimento de um jogo real, na qual foram coletadas métricas de desenvolvimento durante todo o processo de produção do jogo, a fim de validar a eficácia dos métodos ágeis que compõem o framework proposto.

Os dados foram coletados utilizando-se as ferramentas CASE propostas e também entre os stakeholders do projeto. Para tal, foram adotados métodos qualitativos como questionários eletrônicos e também métodos semi-quantitativos, como coleta de métricas. Os dados coletados passaram por análise e validação. Este processo tratou da validação quanto à consistência, precisão e contextualização dos dados coletados. Os métodos utilizados na análise e validação foram a auditoria e relacionamento dos dados coletados.

Foram executados ainda testes de jogabilidade, permitindo que pessoas jogassem versões prévias do jogo em busca de problemas, assim como testes dos módulos e seus elementos. Para tal, elaborou-se uma rotina para testar cada módulo ou elemento do jogo individualmente, assegurando que a produção atingisse o nível de satisfação proposto inicialmente.

5 Resultados e Discussões

Ao fim da execução do estudo de caso apoiado pelo iGAMA, ressalta-se que foi produzido um jogo eletrônico funcional, com o escopo totalmente satisfeito, de acordo com as exigências colocadas pelo stakeholder e com finalização antecipada ao tempo previsto. Apresentam-se, na Figura 3, alguns quadros do jogo em execução.

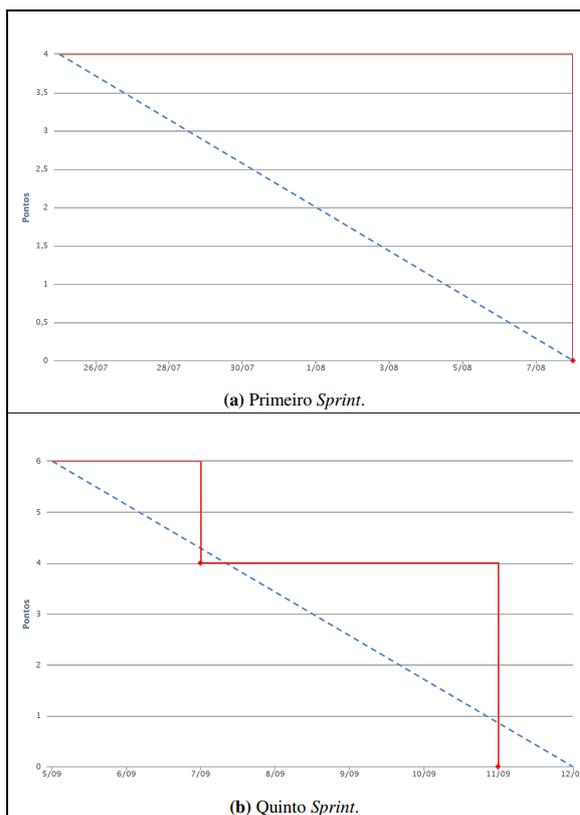
Figura 3 – Quadros do Jogo em Execução



Fonte: elaborado pelos autores.

Em relação ao gerenciamento provido pelo Scrum, os gráficos de Burndown e Burnup gerados por meio da aplicação web ScrumHalf foram de suma importância para o bom aproveitamento do projeto. Apresentam-se, na Figura 4, os gráficos de Burndown coletados durante o primeiro e o quinto Sprint do projeto.

Figura 4 – Gráfico de Burndown da quinta sprint



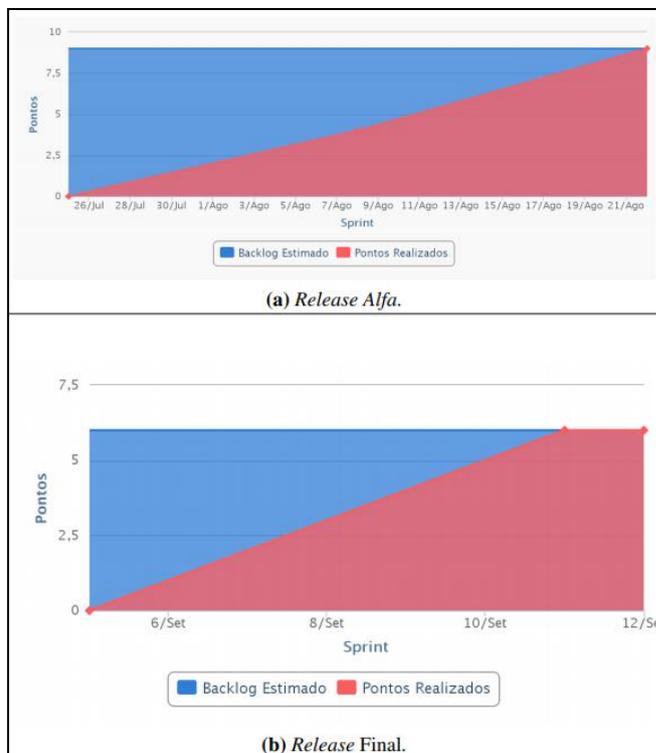
Fonte: elaborado pelos autores.

Ao analisar os gráficos de Burndown, percebe-se que o primeiro Sprint (Figura 4a) foi encerrado com a entrega dos itens de Backlog dentro do prazo previsto, porém fora da linha ideal prevista pela aplicação. O segundo Sprint demonstrou uma evolução em relação a este panorama, entretanto os itens de Backlog ainda foram entregues fora da linha ideal prevista, porém de maneira mais acentuada. Devido ao aprimoramento contínuo proporcionado pelos métodos ágeis, nota-se a partir do terceiro, quarto e por fim quinto Sprint (Figura 4b) que a linha de previsão foi superada gradativamente, sendo o último Sprint finalizado antes do prazo. Este resultado revela que o aprimoramento contínuo estimulado pelos métodos ágeis propiciou uma evolução constante do desenvolvimento do projeto ao longo dos Sprints.

Observa-se também essa evolução por meio dos gráficos de Burnup, especialmente no Release Final, no qual nota-se a entrega do produto de software final antecipada a data

prevista. Apresentam-se, na Figura 5, os gráficos de Burnup dos Releases Alfa (Figura 5a) e Final (Figura 5b) demonstrando também a evolução proporcionada.

Figura 5 - Gráficos de Burnup dos Releases



Fonte: elaborado pelos autores.

A análise dos fatores de desenvolvimento é um importante método utilizado para validar a aplicação de métodos ágeis (SATO, 2007). Os fatores de desenvolvimento mais relevantes são o nível pessoal de experiência com processos adaptativos, o dinamismo que reflete a adaptabilidade em relação as mudanças de requisitos, a criticabilidade que demonstra o nível crítico das perdas devido ao impacto dos problemas, a reincidência de problemas que determina a quantidade de problemas reincidentes e a resolução de problemas que indica quantos problemas foram solucionados em tempo hábil (SATO, 2007).

Os fatores de desenvolvimento do estudo de caso foram medidos por meio dos resultados obtidos em um formulário de avaliação dos métodos ágeis respondido pelo desenvolvedor do estudo para constatar o nível de satisfação obtido com a aplicação do iGAMA. As respostas foram coletadas entre os dias 08 de agosto e 11 de setembro de

2016. Apresentam-se, na Tabela 1, os resultados da avaliação do framework iGAMA por meio dos fatores de desenvolvimento.

Tabela 1 - Avaliação do framework iGAMA.

Sprint	Experiência pessoal ^a	Dinamismo ^a	Reincidência de problemas	Resolução de problemas ^b	Criticabilidade ^c
1	5	4	0	5	1
2	5	4	0	2	1
3	5	5	0	0	0
4	5	5	0	1	1
5	5	5	0	0	0

^a Os valores numéricos representam: 5 – excelente; 4 – bom.

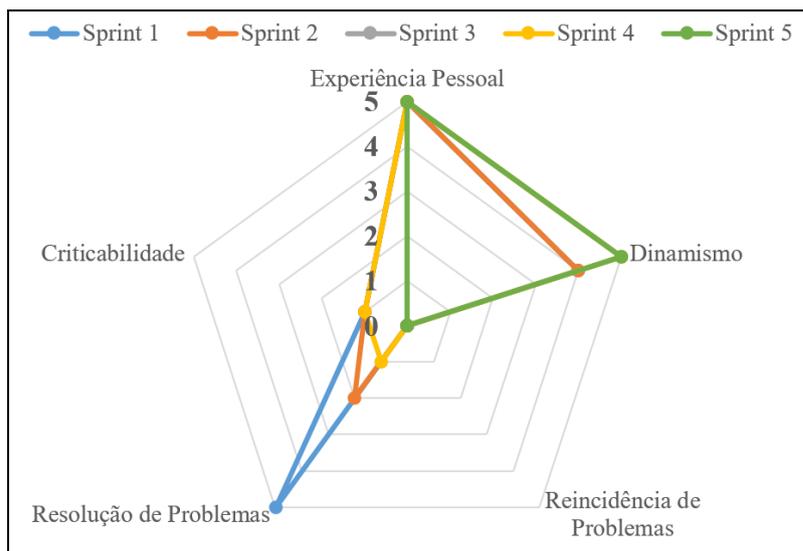
^b Valor numérico zero indica que não ocorreram problemas.

^c Os valores numéricos refletem as perdas devido ao impacto dos problemas e representam: 1 – impactou, porém todos os problemas foram contornados; 0 – não impactou

Utilizando-se dos dados coletados, apresenta-se, na Figura 6, o gráfico de radar dos fatores de desenvolvimento do iGAMA. Uma análise complementar é discutida a seguir.

Ao analisar os fatores de desenvolvimento, apresentados na Figura 6, observa-se que os fatores relacionados a experiência pessoal, dinamismo e resolução de problemas são evidenciados positivamente em todos os métodos ágeis avaliados, obtendo avaliações próximas ao valor máximo. Os fatores associados a criticabilidade e reincidência de problemas são muito baixos. Nota-se ainda, por meio dos fatores de desenvolvimento, que a aplicação do iGAMA auxiliou positivamente na gestão dos fatores no decorrer do desenvolvimento do projeto, obtendo pouca criticabilidade e nenhuma reincidência de problemas, porém proporcionando uma boa experiência pessoal, um bom nível de dinamismo e apoiando a resolução de problemas.

Figura 6 – Gráfico de Fatores de Desenvolvimento do iGAMA



Fonte: elaborado pelos autores.

Ao avaliar os resultados obtidos observa-se que o estudo alcançou bons resultados, mitigando os problemas inerentes a produção. Observa-se também que a aplicação do iGAMA proporcionou a evolução constante do projeto resultando em um produto final de qualidade e com todos seus requisitos satisfeitos.

6 Conclusão e Trabalhos Futuros

Com base na análise dos resultados obtidos com o estudo, conclui-se que a proposta do iGAMA auxilia na organização e no desenvolvimento de um fluxo de trabalho bem definido para a aplicação de metodologias ágeis para o domínio de jogos eletrônicos, colaborando deste modo para um bom aproveitamento do projeto e para a otimização dos resultados obtidos. Constata-se que os problemas verificados na indústria de jogos devido a utilização de métodos tradicionais de desenvolvimento foram minimizados e mitigados com a aplicação do framework ágil proposto. Deste modo, este estudo mostra-se como um ponto de partida e estímulo para o uso de métodos ágeis no processo de desenvolvimento e gestão de projeto de jogos eletrônicos.

Os resultados obtidos neste estudo motivam possibilidades para novas pesquisas. Uma perspectiva interessante para um trabalho futuro, seria a execução de um novo estudo

de caso, similar ao apresentado neste artigo, entretanto ao invés de ser desenvolvido em ambiente acadêmico o mesmo poderia ser efetuado em uma empresa comercial de desenvolvimento de jogos eletrônicos com a participação de um maior número de desenvolvedores. Desta forma, os resultados obtidos por meio do novo estudo poderiam ser comparados com os resultados deste estudo, avaliando se os mesmos continuam verdadeiros.

Referências

- FLOOD, K. **Game unified process**. 2003. Disponível em: <https://www.gamedev.net/resources/_/technical/general-programming/game-unified-process-r1940>. Acesso em: 26 abr. 2016.
- LIBARDI, P. L. O.; BARBOSA, V. **Métodos ágeis**. Monografia (Graduação em ciência da computação) – Universidade Estadual de Campinas, Campinas, 2010. Disponível em: <http://www.ft.unicamp.br/liag/Gerenciamento/monografias/monografia_metodos_ageis.pdf>. Acesso em: 12 maio 2016.
- NOVAK, J. **Game development essentials**. 2. ed. Massachusetts, EUA: Cengage Learning, 2010. 404 p.
- PETRILLO, F. **Práticas ágeis no processo de desenvolvimento de jogos eletrônicos**. Dissertação (Mestrado em ciência da computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008. Disponível em: <<http://hdl.handle.net/10183/22809>>. Acesso em: 2 maio 2016.
- PWC. **Video games key insights at a glance**. New York, 2015. Disponível em: <<http://www.pwc.com/gx/en/industries/entertainment-media/outlook/segment-insights/videogames.html>>. Acesso em: 21 abr. 2016.
- SABBAGH, R. **Scrum: gestão ágil para projetos de sucesso**. 1. ed. São Paulo: Casa do Código, 2013. 17-36 p.
- SATO, D. T. **Uso eficaz de métricas em métodos ágeis de desenvolvimento de software**. Dissertação (Mestrado em ciência da computação) — Universidade de São Paulo, São Paulo, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/45/45134/tde-06092007-225914/publico/dissertacao.pdf>>. Acesso em: 13 out. 2016.
- STEFFEN, J. B. **Lean para desenvolvimento de software: afinal, o que é isto?** IBM developerWorks, São Paulo, 2011. Disponível em: https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/lean_para_desenvolvimento_de_sw_o_que_c3_a9_isso_afinal12?lang=en>. Acesso em: 29 abr. 2016.
- SUNDSTRÖM, Y. **Game design and production: Frequent problems in game development**. Monografia (Licenciatura em design de jogos) — Universidade de Gotlândia, Gotlândia, Suécia, 2012. Disponível em: <<http://www.diva-portal.se/smash/get/diva2:604988/FULLTEXT01.pdf>>. Acesso em: 26 abr. 2016.