

Um Processo Ágil Multidisciplinar de Desenvolvimento de Jogos para Estúdios Independentes

Carlos Eduardo Freitas¹Arthur de Castro Callado¹Danielle Rousy Silva²Paulyne Matthews Jucá^{1*}Universidade Federal do Ceará, Campus de Quixadá, Brasil¹Universidade Federal da Paraíba, Departamento de Informática, Brasil²

RESUMO

Por ser uma área multidisciplinar, existem divergências sobre se os jogos são produtos de arte que contém software ou produtos de software que contém arte. O problema com ambos os pontos de vista é que os aspectos importantes da produção geralmente podem ser subestimados. Este artigo apresenta um processo de desenvolvimento de jogos que pretende combinar processo de desenvolvimento de software com base no processo Scrum com processo de desenvolvimento artístico para fornecer um processo melhor equilibrado. O processo proposto também foi adaptado para equipes com dedicação de tempo parcial e com membros atuando em mais de um papel no processo, como acontece em projetos independentes ou *indie*.

Palavras-chave: Scrum, criação de jogos, desenvolvimento de jogos, elementos artísticos.

1 INTRODUÇÃO

Pesquisando em documentos de post-mortem de projetos de diferentes tamanhos, um time de pesquisadores [1] descobriu que a maior parte dos problemas no desenvolvimento de jogos está relacionada a problemas de gerenciamento. Por se tratar de uma área multidisciplinar que envolve a participação de profissionais de artes, som, diferentes profissionais de computação, game designer, entre outros, a comunicação e a coordenação de atividades entre as pessoas de diferentes *backgrounds* pode ser um desafio.

Uma forma de coordenar as atividades é através da adoção de um processo de desenvolvimento. Um processo de desenvolvimento pode ser descrito como um conjunto de atividades, artefatos criados e usados pelas atividades e papéis responsáveis pela execução das atividades. Existem, na literatura, algumas propostas de criação de processos de desenvolvimento e propostas de guias sobre características que os processos de desenvolvimento devem seguir [1][2][3]. Muitos desses trabalhos relatam adaptações feitas em modelos de processos de desenvolvimento de software para a área de jogos e alguns como [3] apresentam os resultados da aplicação do processo proposto em empresas já estabelecidas e com equipes dedicadas. Poucos tratam das necessidades e desafios da adaptação de processos de jogos para equipes não dedicadas e esse é o principal diferencial desse trabalho.

Utilizar um processo pode ser desafiador especialmente para pequenas empresas ou desenvolvedores *indie*. Existe sempre o medo que o processo torne o trabalho burocrático. Isso faz com que muitas empresas trabalhem com procedimentos “artesanais” criados a partir de necessidades vistas ao longo do tempo, ou até

mesmo sem padrões que garantam a qualidade do produto [4].

A multidisciplinaridade da área de jogos traz também outro desafio na proposição de um processo de desenvolvimento. Os processos criativos e de produção usados por artistas são, muitas vezes, muito diferentes dos utilizados por desenvolvedores de software. Mas o projeto do jogo como um todo precisa permitir que essas atividades das diferentes áreas estejam integradas.

Assim, o processo proposto por esse trabalho também visa aliar atividades de processos artísticos dentro de ciclos iterativos de desenvolvimento comumente definidos em processos de desenvolvimento de software ágeis como o Scrum. O processo proposto também toma como base outra diferença em comparação com os processos já definidos. Esse projeto define como público alvo as empresas independentes (*indie*) e os grupos não formais de desenvolvedores de jogos. Projetos *indie* são, por definição, projetos criados sem o suporte financeiro de grandes corporações. Nesse cenário, as equipes desses projetos *indie* são definidas por esse artigo como equipes mais reduzidas, podendo ter as empresas funcionários com dedicação de tempo parcial para o projeto, ou mesmo como projetos *indie* dos quais participam voluntários sem alocação fixa onde muitos têm outros empregos e utilizam o tempo livre para desenvolver seus projetos. Tais tipos de baixa dedicação de tempo dos membros da equipe criam novos desafios para o respeito a um processo de desenvolvimento, dificultando algumas práticas comuns em grandes/médias empresas. Por exemplo, pode não ser possível, em uma equipe *indie*, fazer uma reunião diária em horário determinado com toda a equipe.

2 TRABALHOS RELACIONADOS

Em [4] foi proposta uma metodologia para o desenvolvimento de jogos, baseada em metodologias ágeis e prototipação denominada Origame. O Origame divide o processo geral em três fases: Design e Projeto, Produção e Implementação. Embora o processo Origame seja bem estruturado e sirva como um ótimo guia sobre os tipos de artefatos produzidos em cada fase do desenvolvimento, esse processo não define de forma satisfatória as atividades relacionadas ao desenvolvimento de software e não define claramente como atividades artísticas e de desenvolvimento se retroalimentam.

Em [3], foi feita uma adaptação na metodologia Scrum para ser aplicada em projetos de jogos. Essa adaptação ocorreu a partir da análise de problemas encontrados no decorrer da aplicação do Scrum em uma empresa de desenvolvimento de games. Essa adaptação também propõe a criação de um papel novo (*Game Design Master*), um artefato novo (*Game Design Wiki*) e uma reunião adicional ao Scrum (*Weekly Game Design Meeting*). Em muitos aspectos as conclusões encontradas pelos pesquisadores se assemelham às desta proposta como a separação dos times de artes e desenvolvimento com reunião de interação entre seus trabalhos e a manutenção de um GDD enxuto. Entretanto, a principal diferença é que o processo proposto tem um foco maior

*e-mail: paulyne@ufc.br

ao adicionar e propor dentro do processo Scrum os pontos onde são adicionadas as atividades relacionadas à integração com processos artísticos e também pelo foco em equipes não totalmente dedicadas, o que inviabiliza, por exemplo, a execução de reuniões diárias. Este trabalho também permite uma maior liberdade da equipe ao escolher a ferramenta para gerenciamento do GDD.

A seguir está descrito o processo artístico que serviu de base para o processo proposto.

3 PROCESSO ARTÍSTICO

A criatividade pode ser vista como a capacidade de criar coisas novas. Apesar da criatividade ser normalmente entendida como uma atividade livre e avessa às regras, existem muitas técnicas de criatividade e processos criativos.

Algumas técnicas de criatividade muito utilizadas incluem: *brainstorms*, os seis chapéus do pensamento, mapas conceituais, mapas mentais, caixas morfológicas, inversão de hipóteses e associação de ideias. Está fora do escopo deste trabalho apresentar as técnicas de criatividade. As técnicas podem ser utilizadas em diferentes atividades do processo criativo. Partes das etapas do processo criativo são usadas na definição do processo proposto nesse trabalho.

Um processo criativo muito citado na literatura de criatividade é o processo proposto por [5]. Ele foi originalmente criado para o marketing, mas é genérico o suficiente para ser adaptado para este trabalho. Os autores definem um processo criativo composto por 7 etapas: identificação do problema, preparação, incubação, aquecimento, iluminação, elaboração e verificação.

A identificação do problema é a etapa onde se pretende responder a pergunta “qual é o problema?”. Essa etapa pretende avaliar as necessidades da tarefa e envolve o uso de muitas das técnicas de criatividade.

A preparação é a etapa de coleta de informações necessárias para entender o problema. Utiliza-se de perguntas como “o quê?”, “quem?”, “quando?”, “onde?”, “como?” e “por quê?”.

A incubação acontece depois que todos os dados da preparação foram coletados. As pessoas conhecem suficientemente o problema a ser resolvido e agora precisam de tempo para refletir sobre ele. Os autores comentam que a incubação acontece com a mente em descanso.

O aquecimento é quando existe uma volta do trabalho consciente no desenvolvimento do problema. O uso de *brainstorms* pode acontecer nessa fase. Ela gera a iluminação que é quando a solução é criada pela primeira vez e corresponde ao momento “eureka”.

A elaboração é onde a solução é construída de fato. Na verificação, a solução proposta é validada.

Para esse trabalho, o processo criativo descrito anteriormente foi simplificado e gerou 3 novas etapas no processo proposto: concepção da ideia, criação artística e desenvolvimento do design. A concepção da ideia é composta da execução completa do processo criativo, utilizando diferentes técnicas de criatividade para onde o problema a ser resolvido é “qual jogo será desenvolvido?”. O artefato produzido ao final da fase de concepção da ideia é o GDD e dele é derivado o *backlog* do produto (ver a definição do processo Scrum).

4 SCRUM

Há cerca de dez anos, o Scrum foi formalizado como um processo ágil. Pouco tempo depois a sua aceitação e utilização por parte da comunidade se deu rapidamente, sendo hoje uma metodologia padrão em várias companhias [9].

O ciclo do Scrum acontece no decorrer de várias iterações bem definidas, chamadas de *sprint*. Cada uma destas iterações tem duração de 2 a 4 semanas, a ser definida pela equipe do projeto no

começo do desenvolvimento. Antes de cada *sprint*, realiza-se uma reunião de planejamento (*Sprint Planning Meeting*), onde o time (equipe) de desenvolvedores se reúne com *Product Owner* (representante do cliente) para definir e priorizar o que precisa ser feito, escolher e estimar as tarefas que o time deve realizar dentro de uma *sprint* [6].

Em seguida, temos a execução das *sprints*. Durante a execução de uma *sprint*, o time gerencia o andamento do desenvolvimento do projeto por meio de reuniões diárias rápidas (*Daily Meeting*), que não duram mais que 15 minutos, e observam o progresso do desenvolvimento através do gráfico chamado *Sprint Burndown*, que é atualizado diariamente [6].

Ao final de cada *sprint*, é feita uma revisão do produto gerado para verificar se tudo que foi definido no *Sprint Backlog* foi devidamente implementado. Esta revisão acontece na reunião de revisão (*Sprint Review*), na qual o time demonstra o produto gerado na *sprint* e valida se o objetivo foi atingido. Logo em seguida, realiza-se a reunião de retrospectiva (*Sprint Retrospective*), uma reunião para coleta de lições aprendidas feita com o objetivo de melhorar o processo, o time e/ou o produto para a próxima *sprint* [6].

Dentro do framework de processos Scrum, existem alguns papéis bem definidos com tarefas e finalidades diferentes durante o processo, sendo eles o *Scrum Master*, *Product Owner*, *Scrum Team*. A seguir uma descrição desses papéis de acordo com as definições de [7]:

- *Scrum Master* – “É responsável por garantir que o projeto seja realizado, gerindo a equipe de acordo com as práticas, valores e regras do Scrum e pelo avanço da equipe no projeto conforme o planejado”.
- *Product Owner* – É responsável por definir os itens que irão compor o *Product Backlog* (ou *backlog* de produto, que é a lista de funcionalidades a serem desenvolvidas) e por priorizar estes itens nas *Sprint Planning Meetings*. O *Product Owner* pode representar os desejos de parte dos interessados no desenvolvimento e em parte das funcionalidades listadas no *Product Backlog*, mas outros interessados que desejam alterar a prioridade de um item no *backlog* do produto devem dirigir suas solicitações ao *Product Owner*, pois apenas ele participa das reuniões de planejamento (*Sprint Planning Meeting*).
- *Scrum Team* – É o time de desenvolvimento do projeto.

O time deve controlar como as tarefas devem ser executadas, buscando sempre manter a organização. Durante a *sprint* não deve existir interferência externa. Isso ajuda a manter um foco por não precisar lidar com mudanças constantes durante a *sprint*. Esse é um dos principais papéis do *Scrum Master*: evitar que o time tenha qualquer desvio do objetivo traçado [6].

O Scrum não foi feito para atender apenas às necessidades encontradas no desenvolvimento de software, mas criado no intuito de atender a qualquer projeto complexo, de qualquer área, principalmente os que propõem inovações. Neste sentido, é comum ver sua grande utilização na área do desenvolvimento de jogos, já que, em sua maioria, são projetos complexos e multidisciplinares [8].

Entretanto, considerando que o desenvolvimento de jogos é multidisciplinar e, apesar do Scrum se propor a ajudar no controle de processos de produção em geral e trazer muitos benefícios vindos do processo mais focado na construção que na auditoria dos processos como acontece com processos tradicionais de software, esse trabalho identificou algumas necessidades de adaptação para o Scrum a serem feitas pensando em equipes de dedicação de tempo parcial e pequenos grupos de desenvolvimento. Dentre essas necessidades de adaptação, podemos listar: a dificuldade de realizar reuniões diárias, a

dificuldade de realizar a comunicação frente a frente, a dificuldade de realizar reuniões com todos os membros da equipe sempre, a necessidade de maior independência de trabalho entre membros, especialmente a separação do time de artistas do time de desenvolvedores, a inexistência de cliente externo, pois os jogos não são contratados e sim frutos de produção independente, membros com múltiplos papéis no desenvolvimento e por isso, não especialistas, o que pode levar a maiores erros de estimativa.

5 PROCESSO PROPOSTO

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. Essas atividades usam rótulos para identificar visualmente de maneira mais simples o volume de trabalho de cada parte da equipe. No projeto de avaliação, verde significa atividade da equipe de artes e azul representa atividade de desenvolvimento, por exemplo. A estimativa de esforço de cada atividade a ser realizada é feita pelos integrantes da equipe correspondente pela execução dessa atividade como forma de melhorar a estimativa.

Outra diferença acontece dentro da execução das *sprints* quando o processo é dividido em dois subprocessos (um para atividades artísticas e outro para desenvolvimento do software) que se retroalimentam. O subprocesso artístico foi pensado para ser executado pela equipe de design em paralelo ao desenvolvimento lógico do jogo. Neste sentido, elas ocorrem após a definição do *Sprint Backlog* do jogo que inclui tarefas de arte e de desenvolvimento de software. No início de cada *sprint*, acontece a reunião de planejamento como definida no Scrum. A única adaptação necessária é na escolha das atividades da *sprint* que devem levar em consideração o esforço disponível para cada subprocesso (artístico e software), pois as duas subequipes podem não ter tamanho e capacidade produtiva igual.

As atividades do processo proposto são descritas em mais detalhes a seguir:

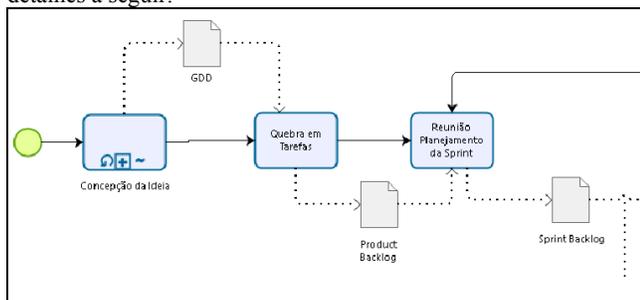


Figura 1: Início da execução do processo proposto.

Concepção da Ideia - Para que um jogo comece a ser produzido, as ideias precisam ser criadas e aprovadas pelas equipes de desenvolvimento e de design e pelo o supervisor do projeto. Neste sentido, o objetivo dessa etapa é definir o que será o jogo. Em paralelo à concepção da ideia, há o desenvolvimento da documentação do projeto, neste caso o *game design document* (GDD) do jogo, que descreve em detalhes os conceitos e os principais elementos de jogabilidade deste. Para o Scrum, o processo começa da definição do *backlog* do produto, mas no processo proposto o ponto inicial é essa etapa de concepção que tem como objetivo criar o GDD. É importante ressaltar que o GDD é um documento vivo e não precisa estar totalmente terminado para começar o processo. Sempre que o GDD receber

atualizações, essas mudanças serão refletidas no *backlog* de produto.

Quebra em tarefas - Nesta etapa, a equipe discute para definir uma lista com todas as funcionalidades e elementos desejados para o jogo, usando como referência o GDD. Dependendo do estilo do jogo, essa lista vai variar. Para um jogo estilo plataforma pode-se definir as fases e cenários que o jogo vai possuir, detalhar os personagens presentes em cada fase e os itens a serem coletados. Para um jogo estilo *infinity run* (corrida infinita) será definido o cenário onde o jogo se passa, os elementos a se coletar, os personagens jogáveis. As ideias geradas aqui formarão o *backlog* do produto do jogo que não precisa estar completo no início do projeto.

Backlog do Produto (Product Backlog) - É uma lista de funcionalidades, oriundas da definição do GDD, que precisam ser desenvolvidas. Ele inclui atividades de artes e de desenvolvimento. Pode incluir atividades de investigação, prototipação e teste também. Ele pode receber ou perder atividades sempre que o design do jogo mudar. Esse processo sugere que as atividades de subequipes diferentes sejam representadas visualmente por cores diferentes para permitir a identificação visual do esforço por tipo de atividade dentro dessa equipe multidisciplinar. O uso do *backlog* do produto permite entender quanto de esforço falta para acabar o desenvolvimento e, como as atividades são rotuladas por tipo, seu uso também pode permitir perceber se a equipe precisa de ajuda extra para algum aspecto do jogo devido ao acúmulo de atividades de um tipo (artes ou software) durante o desenvolvimento. Por exemplo, se as atividades de música do jogo de todas as fases ficam sempre para depois, essa será uma atividade que representará problemas.

Reunião de Planejamento da Sprint - Nesta etapa, a equipe se reúne com o *Scrum Master* para definir o que será feito na *sprint*, levando consigo as funcionalidades e elementos já separados na etapa anterior presentes no *backlog* de produto e o tempo estimado que estas irão levar para serem produzidas. A estimativa, ao contrário do Scrum tradicional, é feitas apenas pelos membros da equipe que participam das atividades daquele tipo (arte por artistas e software por desenvolvedores), independente do nível de especialização do profissional. As decisões tomadas aqui são documentadas no *backlog* da *sprint*.

Backlog da Sprint (Sprint Backlog) - É uma lista de tarefas que a equipe se compromete a realizar durante uma *sprint*. Os itens do *backlog da sprint* são extraídos do *backlog* do produto pela equipe.

Criação Artística - Esta etapa é um subprocesso Ad-hoc no qual as atividades contidas dentro dele não possuem uma ordem de execução definida e se repetem quantas vezes forem necessário. As atividades deste subprocesso são: identificação do problema, preparação, incubação, aquecimento e iluminação e podem envolver todas as técnicas de criatividade que sejam necessárias. A definição destas atividades está presente na seção 4.2. Este subprocesso vai guiar a equipe de design na construção dos elementos artísticos do jogo. Nem todas as atividades precisam ser executadas para ter-se um produto final de design. O intuito do subprocesso é dar suporte à construção artística no que for necessário para o jogo. O resultado deste subprocesso é a arte conceitual do jogo.

Arte Conceitual do Jogo (concept art) - Segundo a comunidade artística, *concept art* é a forma de ilustração na qual o objetivo principal é transportar uma representação visual de um design, ideia, e/ou modo para uso em filmes, vídeo games, ou revistas/livros antes do produto final.

Desenvolvimento do Design - Aqui a equipe de design vai produzir as artes finais, sejam elas *sprites* 2D ou modelos 3D, baseados na arte conceitual do jogo. Podem incluir também quaisquer outros produtos artísticos como sons e músicas.

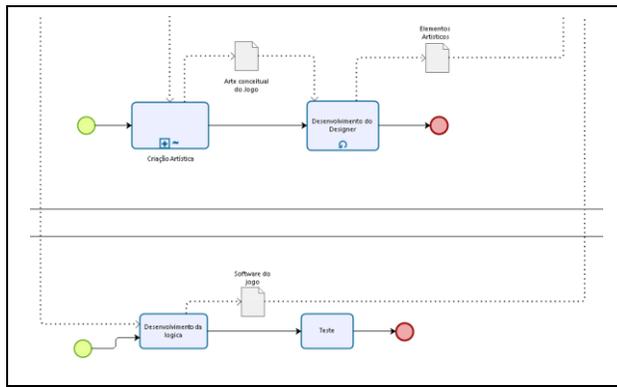


Figura 2: Início da execução interna da *sprint*.

Desenvolvimento da Lógica - Esta etapa acontece em paralelo à construção dos elementos artísticos. Nela, a equipe de software desenvolve as funcionalidades do jogo. São exemplos de atividades de desenvolvimento de lógica: pular, correr, movimentação, poderes, pontuação, algoritmos de inteligência artificial. Toda a lógica por trás do jogo será feita nesta etapa.

Por se tratar de equipe com dedicação de tempo parcial ou mesmo esporádica, pode ser difícil realizar as reuniões diárias. Muitas vezes os projetos andam de maneira independente ou ficam parte da semana estacionados esperando a disponibilidade dos membros da equipe. Assim, a realização de reuniões diárias pode ficar prejudicada. Entretanto, é de vital importância que as equipes se reúnam periodicamente. Assim, esse processo recomenda a ocorrência de pelo menos 1 reunião semanal conjunta com a participação de membros de todos os times (reunião integrada) e pelo menos 1 reunião semanal para cada subequipe de preparação para a reunião integrada. Assim, cada subequipe identifica suas dependências, pedidos de mudança e *status* do desenvolvimento antes de encontrar com toda a equipe.

Testes - Estes testes estão ligados à funcionalidade lógica do jogo e podem incluir testes unitários automatizados. Seu objetivo é verificar se há bugs na execução de eventos do jogo e corrigi-los.

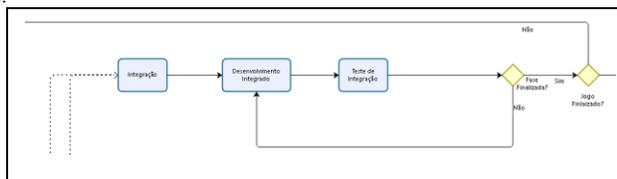


Figura 3: Final da execução interna da *sprint*.

Integração - Nesta etapa, os conteúdos produzidos pela equipe de design e pela equipe de software serão mesclados. Seu objetivo é juntar cada elemento artístico a sua funcionalidade lógica dentro do jogo, bem como ligar os componentes aos seus elementos visuais como texturas para o chão, cenário, céu.

Desenvolvimento da Integração - Esta é uma das etapas mais importantes para o processo. Aqui as equipes de design e software se reúnem para fazer as adequações necessárias em ambos os lados, quer seja alguma funcionalidade que precise ser alterada para se encaixar ao elemento artístico ou um elemento artístico que não se encaixou bem à funcionalidade e precisa ser alterada. Neste momento, as duas equipes devem estar mais próximas para discutirem o que precisa se adequar para que resulte no produto final. Essa etapa pretende suprir as deficiências da dedicação em tempo parcial e separação em equipes independentes, pois permite o trabalho em paralelo, mas exige um momento de combinação de esforços.

Teste de Integração - Esta etapa vai executar testes após a junção do design com o software, verificando funcionalidades e sua execução com a arte.

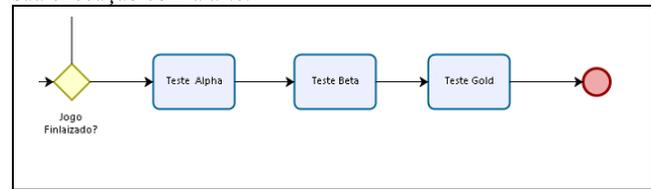


Figura 4: Execução final do processo.

Testes Alpha - Primeira versão do seu jogo pronto para o teste. A versão alpha é feita para ajudar no apontamento dos diversos bugs que a equipe não viu e dar uma amostra das funções ou características de certas funções do seu jogo.

Teste Beta - Beta é a segunda rodada de testes do jogo. As versões beta costumam ser tanto abertas quanto fechadas, e claro, usadas para levantamento de *feedback* como na alpha. Ela deve estar mais polida e perto da finalização, mas a beta ainda pode ter bugs e funcionalidade não totalmente finalizadas.

Teste Gold - Esta é a versão final do jogo, neste ponto o jogo já pode ser lançado ao público, pois o produto está totalmente acabado e os jogadores podem ter acesso total ao jogo. Embora o jogo esteja totalmente terminado, há muitas chances em que um jogador possa encontrar bugs. Nesses momentos, os desenvolvedores devem liberar *patches* para corrigi-los.

6 CONCLUSÃO

Esse artigo apresenta uma proposta de processo de desenvolvimento de jogos que combina o Scrum com processos criativos para melhor atender a multidisciplinaridade da área de forma equilibrada. O processo proposto também tem como alvo equipes com dedicação de tempo parcial ou grupos pequenos de desenvolvedores como acontece nos estúdios *indie*.

O processo apresentado está sendo avaliado no desenvolvimento de um jogo realizado por uma equipe com dedicação de tempo parcial para melhor avaliação.

REFERÊNCIAS

- [1] F. Petrillo, M. Pimenta, F. Trindade e C. Dietrich, "Houston, We Have a Problem...: A Survey of Actual Problems in Computer Games Development," em *Proceedings of the 2008 ACM Symposium on Applied Computing*, Fortaleza, 2008.
- [2] A. Godoy e E. Barbosa, "Game-Scrum: An Approach to Agile Game Development," em *SBGAMES*, Florianópolis, 2010.
- [3] A. Laubisch e E. Clua, "Scrum4games: Uma aplicação do scrum para projetos de games focada em game design," em *IX Proceedings do Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, Florianópolis, 2010.
- [4] R. Santos, V. Góes e L. Almeida, "Metodologia origame: um processo de desenvolvimento de jogos," em *XI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGAMES)*, Brasília, 2012.
- [5] R. Dualibi e H. Simonsen, em *Criatividade e marketing*, São Paulo, Makron Books, 2000.
- [6] "Scrum Alliance-transforming the world of work," [Online]. Available: <https://www.scrumalliance.org/>. [Acesso em 16 outubro 2016].
- [7] K. Schwaber e M. Beedle, *Agile Software Development with Scrum*, Editora Prentice Hall, 2007.
- [8] C. Keith, *Agile game development with Scrum*, Pearson Education, 2010.
- [9] H. Kniberg, em *Scrum e XP Direto das Trincheiras*, C4Media, 2007.