

Scrum: Novas Regras do Jogo

Allan R. S. Araujo Juliana M. Silva Artur F. Mittelbach

Jynx Playware, Brasil

Resumo

Este trabalho tem o objetivo de mostrar os desafios demandados pela criação de um jogo publicitário, ou advergame e sugerir uma nova abordagem de processos para projetos de advergames.

Como forma de construir o estudo, se analisa a maneira como a Jynx Playware costumava desenvolver jogos (advergames) e sua decisão de mudar.

Assim, são discutidas as vantagens e desvantagens da nova abordagem adotada e, então, são sugeridos trabalhos que possam fazer com que a abordagem de criação de advergames possa ser melhorada tanto nos aspectos gerenciais, quanto nos aspectos de desenvolvimento.

Palavras-Chave: Advergames, Métodos Ágeis, *Scrum*.

Contato:

allan@jynx.com.br
artur@jynx.com.br
juliana@jynx.com.br

1. Advergames

Um jogo é uma atividade estruturada (ou semi-estruturada) onde os jogadores devem alcançar determinados objetivos, respeitando um conjunto de regras, envolvendo estímulos físicos e/ou mentais. Em geral, os jogos têm propósito de entretenimento, porém, podem ser utilizados com outros fins, como educação, treinamento ou publicidade [Crawford2003].

Sendo assim, "advergames" são jogos criados com vocação publicitária cujo principal objetivo é disponibilizar mensagem de propaganda para promover um produto, uma organização ou um ponto de vista [Chen2001]. Embora a mídia gerada pelos advergames seja bastante recente, existe uma forte tendência de crescimento das oportunidades nesta área. O poder de imersão causado pelos jogos e a grande quantidade de capital destinada pelas empresas para fins publicitários indicam que haverá num grande crescimento neste setor [IGDA2006]. Os principais objetivos de um advergame devem a capacidade de causar um forte poder de imersão do usuário, para que ele possa perceber a mensagem publicitária e, ao mesmo tempo, ser capaz de ter um grande e vasto número de usuários.

Um projeto de criação de um advergame consome de US\$ 10mil a US\$ 500 mil, a depender do escopo do jogo. Porém, o fator limitante de um projeto de advergame é, quase sempre, o cronograma. Afinal, um advergame está sempre ligado a uma campanha de marketing mais ampla através de outras mídias, como TV ou jornais. Além disso, um projeto de advergame é, normalmente, contratado por uma agência ou uma empresa (cliente final) para que uma empresa especializada na produção de jogos execute o projeto. Mas, de maneira geral, os clientes (seja a agência ou a empresa-cliente final) entendem pouco de jogos e dificilmente sabem exprimir suas expectativas com relação ao produto final [IGDA2006]. A equipe, como em qualquer jogo, é formada por membros de diferentes habilidades tendo um caráter fortemente multidisciplinar.

Por isso, a criação de um projeto de advergame deve ser baseada num processo de desenvolvimento e gerenciamento capaz suportar todo o fluxo de atividades para alcançar os objetivos do projeto.

2. Trabalhos Relacionados

Este trabalho é único e sem precedentes na área de processos para advergames. No entanto, existem outros trabalhos relacionados à área de processos para jogos, como o *Game Unified Process* [Flood2003], *Extreme Game Development* [Demachy2003], entre outros.

3. Jynx e a Velha Ordem

A Jynx é uma empresa focada na produção de jogos sérios, com ênfase em advergames. Durante muito tempo, o esforço para se trazer um projeto para a empresa acontecia em dois momentos distintos.

- **Negócios:** Onde ocorria o trabalho de prospecção, desenvolvimento de um "Game Concept" (uma estrutura que mostra os principais conceitos associados ao jogo) e a elaboração de uma proposta descrevendo as principais restrições associadas ao escopo, cronograma e custo para criação do produto;
- **Produção:** Se o projeto fosse aprovado, as informações criadas na primeira parte do esforço (negócios) eram praticamente ignoradas e era iniciado um extenso período de planejamento, utilizando PERTs [PMBOK2000] e outros tipos de gráficos. Então, o projeto era executado utilizando

algum método baseado no modelo em cascata ou no modelo iterativo. Neste ponto, o processo de desenvolvimento ainda era bastante orientado a documentação e o gerenciamento centralizado.

Neste contexto, havia uma série de problemas que acabavam afetando a satisfação de vários stakeholders [Sommerville2003] do projeto, desde os desenvolvedores do produto até o cliente final. Alguns destes problemas podem ser classificados conforme abaixo:

- **Produto**
 - **Escopo:** O produto entregue sempre necessitava de um número considerável de ajustes para adequar-se às expectativas dos clientes;
 - **Cronograma:** Os prazos eram respeitados a custo de muitas horas-extras de trabalho dos desenvolvedores do produto do projeto;
 - **Custo:** Por conta das necessidades de ajustes e dos turnos extras de trabalho, o custo dos projetos excedia, com significativo grau de relevância, os números projetados.

Uma série de fatores contribuíram para o surgimento dos problemas. Entre eles, os processos de desenvolvimento utilizados. A empresa utilizava um processo de desenvolvimento que se baseava no modelo de ciclo de vida em cascata [Sommerville03]:

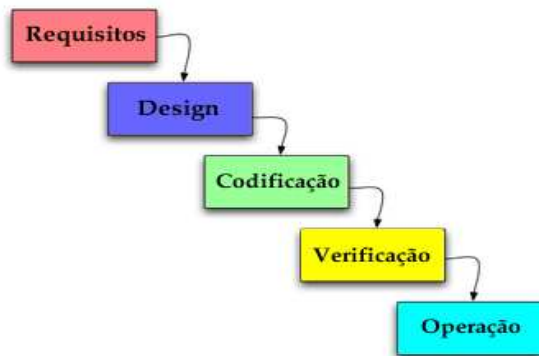


Figura 1: O Modelo de Ciclo de Vida em Cascata

Esta estrutura linear mostra-se pouco flexível para responder a mudanças e provoca uma série de causas dos problemas citados:

- **Feedback tardio:** Os clientes e membros da equipe têm visão limitada e passam grande parte do desenvolvimento sem opinar ou conhecer o estado do produto [Flood2003];
- **Ausência de Colaboração:** Neste tipo de estrutura, cada indivíduo tem um papel bem definido e se preocupa apenas com suas tarefas, promovendo a falta de colaboração entre os membros da equipe [Schwaber2004];
- **Inflexibilidade:** Por ser linear, sua capacidade de responder às mudanças de maneira ágil se torna praticamente nula, havendo um grande índice de retrabalho [Sommerville2003];

- **Orientação a Documentos:** A transição entre as fases deste tipo de modelo de ciclo de vida se dá geralmente através da aprovação de documentos [Sommerville2003]. Assim, a equipe de desenvolvimento fica mais focada em produzir um documento para servir de defesa contra o cliente, do que em produzir valor, de fato, para o cliente.

Além disso, a estrutura de gerenciamento de projetos era centralizada, com extensivos ciclos de planejamento, baseada em métodos e práticas tradicionais, como o PMBOK [PMBOK2000].

Sendo assim, as conseqüências destas abordagens para gerenciamento e desenvolvimento de projetos de advergimes é que, embora os atrasos fossem desprezíveis, os cronogramas eram mantidos em dia a custo de muitas horas extras, bastante retrabalho, pouco valor agregado para o cliente durante grande parte do processo, dificuldade de controlar os projetos complexos e intangíveis, etc. Portanto, a metodologia utilizada nos projetos era, claramente, não aderente às necessidades demandadas por um projeto de advergence.

4. Metodologias Ágeis

Diante do caos, era preciso mudar. Não mudar as práticas, mas os valores. Em 2001, um grupo de renomados engenheiros de software reuniu-se em Utah (EUA), e lançou um manifesto divulgando princípios que toda metodologia ágil deveria respeitar [Xispê2001]:

"Nós estamos descobrindo melhores maneiras de desenvolver software fazendo e ajudando outros a fazê-lo. Através desse trabalho chegamos aos valores:

***Indivíduos e interações** sobre processos e ferramentas.
Software funcionando sobre documentação detalhada.
Colaboração do cliente sobre negociação de contrato.
Responder a mudanças sobre seguir um plano.*

Isto quer dizer, enquanto existe valor nos itens à direita, nós valorizamos mais os itens à esquerda".

Atualmente, metodologias ágeis como *Extreme Programming* e *Scrum* são utilizadas em um grande número de grandes empresas, como Ford, BMW, Nokia, BenQ, Borland, Microsoft, etc. [Xispê2001].

5. Coragem para Mudar

Em ambientes de negócios extremamente dinâmicos e complexos, o caos surge naturalmente. Quando se está desenvolvendo algo que é intangível, é bastante difícil prever e controlar o trabalho nestes projetos. Nestes ambientes críticos, é preciso ter uma metodologia capaz de respeitar estas características e fazer "brotar" ordem a partir do caos. Assim, a Jynx decidiu adotar uma nova metodologia de

desenvolvimento baseada nos valores ágeis e incorporou algumas das melhores práticas sugeridas em cada uma delas.

5.1 Scrum e a Jynx

Scrum é uma metodologia ágil que busca uma forma empírica de lidar com o caos, em detrimento a um processo bem definido [Schwaber2004].

Em ambientes complexos, um "processo bem definido" tem sua probabilidade de sucesso drasticamente reduzida quando aplicado ao desenvolvimento de um produto. Uma das principais idéias do Scrum para atacar este problema é a implantação de um controle descentralizado, que é capaz de lidar mais eficientemente com contextos pouco previsíveis [Schwaber2004].

Para tanto, o gerenciamento é distribuído através de três agentes independentes [Beedle2004]:

- **Product Owner:** É o responsável pelo retorno sobre o investimento do projeto. Seu papel é garantir que o jogo entregue atenda aos anseios do patrocinador do projeto;
- **Team Members:** Ou equipe. São todos integrantes comprometidos em desenvolver o jogo do projeto, alcançando os objetivos. Composto por programadores, *game designers*, músicos, artistas, etc.;
- **ScrumMaster:** É responsável por guiar o time, intermediar negociações entre o "Product Owner" e o Time. Seu principal papel é ensinar e acompanhar a utilização do *Scrum*.

Embora o principal foco seja entregar valor ao cliente, os membros de um time de projeto *Scrum* devem utilizar alguns artefatos como apoio ao gerenciamento descentralizado e simples [Schwaber2004]:

- **Product Backlog:** É uma lista priorizada (pelo *Product Owner*) de todas as histórias de usuário [Cohn2004] que o jogo deve conter;
- **Sprint Backlog:** É uma lista das histórias mais relevantes, quebrada em tarefas com suas respectivas estimativas de duração do presente momento, até o fim do *Sprint*, isto é, da iteração. É definida pelo time, e negociada com o *Product Owner*. O time deve procurar manter esta lista sempre atualizada;
- **Impediment List:** É uma lista que contém os impedimentos para o time alcançar os objetivos. Tais impedimentos devem ser resolvidos pelo *ScrumMaster*.
- **Reports:** São relatórios capazes de mostrar o andamento do projeto. Existem vários: *Sprint Burndown*, *Product Burndown*, *Bug History*, entre outros. Estes reports são muito importantes para controle e tomada de decisão dentro do projeto.

Ao final de um *Sprint* é entregue um *Product Increment*, isto é, uma versão do jogo, contendo todas as histórias implementadas naquele *Sprint*.

Um processo baseado em *Scrum* é extremamente simples e adaptativo. Em um projeto, existem basicamente as seguintes etapas [Teamsystem2005]:

- **Preparação:** Onde se define o *business-case*, *game concept* (conceito do jogo), *Product Backlog* inicial e outras premissas ligadas aos projetos;
- **Sprints:** São iterações realizadas, uma após a outra, para entregar gradativamente as histórias que compõem o jogo. Dentro de cada *Sprint* são realizadas algumas atividades:
 - **Scrum Planning Meeting:** É o início de um *Sprint*, onde o *Product Owner* tem a oportunidade de atualizar a priorização dos itens do *Product Backlog* e definir juntamente com a equipe um *Product Increment* a ser entregue ao cliente ao final do *Sprint*;
 - **Daily Scrum Meeting:** É um encontro diário realizado pela equipe onde os membros discutem aquilo em que trabalharam, no que irão trabalhar e possíveis impedimentos que estejam atrapalhando o progresso do trabalho. Este encontro é uma maneira eficiente de manter os membros cientes dos objetivos e impedir que o projeto "saia do rumo";
 - **Criação do Product Increment:** A finalização das histórias definidas para um determinado *Sprint* marca a realização de um *Product Increment*. *Game Designers*, por exemplo, definem detalhes de jogabilidade sobre as histórias e, programadores criam testes para as histórias e então, fazem o código necessário para que consiga passar pelos casos de teste definidos para aquela história. Similarmente, outros membros da equipe trabalham de maneira colaborativa de forma a realizar todas as histórias definidas para aquele *Sprint*. Um músico trabalha em parceria com programadores, *game designers* com artistas gráficos, etc. Além disso, diariamente, os membros da equipe devem atualizar a estimativa de esforço necessário para finalizar cada história;
 - **Sprint Review:** Neste momento, a equipe exhibe o *Product Increment* construído ao *Product Owner*, que é responsável por validar e/ou solicitar ajustes para que o jogo se torne adequado aos anseios do cliente;
 - **Sprint Retrospective:** Objetiva identificar os pontos positivos e negativos do *Sprint* que entregou o último *Product Increment* e busca corrigir os problemas encontrados.
 - **Atualização do Product Backlog:** O *Product Owner* é responsável por re-priorizar toda lista de itens do *Product Backlog* para que um próximo *Sprint* possa ser iniciado motivado pelos mais prioritários itens.

- **Encerramento:** É a última etapa de um projeto utilizando *Scrum*. Ocorre após a finalização de todos *Sprints* e é marcada pela entrega do jogo final que motivou a criação do projeto.

Portanto, é sobre tudo isto que trata *Scrum*. É sobre reconhecer que mudanças são inerentes e aceitá-las. É sobre o desafio de fazer a ordem emergir do caos. E, principalmente, é sobre promover a maturidade das pessoas e satisfazer as necessidades do cliente.

6. Conclusões e Trabalhos Futuros

A adoção de uma metodologia ágil, como o Scrum, tem permitido à Jynx Playware atacar com coragem todos os principais problemas de gerenciamento e desenvolvimento de projetos de jogos em ambientes de negócios dinâmicos e com grande probabilidade de mudanças.

A nova abordagem foi capaz de integrar visões de negócio às visões de produção, unificou a linguagem de todos envolvidos através do uso de histórias, etc. No entanto, ainda carece de um maior refinamento do ponto de vista de práticas e suporte ao desenvolvimento.

Por isso, um trabalho está sendo desenvolvido para formalizar o *Agile Game Process (AGP)* [Araujo2006], uma metodologia ágil para projetos de advergames. Em sua essência, a metodologia utiliza *Scrum* como estrutura principal. Contudo, o processo é muito mais amplo e robusto que o *Scrum*. O processo é descrito através da notação do SPEM (*Software Process Engineering Metamodel*) [OMG2005], um padrão para descrição de processos. Assim, o processo é organizado através de papéis, disciplinas, fluxos de atividades, artefatos, práticas recomendadas, etc. Além disso, o AGP se aprofunda nas disciplinas de Negócios, *Game Design*, Arte e Som, o que normalmente não ocorre com a maioria dos processos de software. Isto se faz necessário porque jogo não é apenas software.

Logo, os próximos passos serão utilizar o AGP como metodologia para projetos de advergames em algumas organizações e obter algumas métricas sobre o processo. Dessa forma, será possível melhorá-lo e refiná-lo de acordo com a necessidade de cada organização.

Agradecimentos

Agradecimentos à Jynx Playware (e seus membros) e à UFPE (CIn e CAC). Todos foram fundamentais para a realização deste trabalho.

Referências

- [CRAWFORD2003] CRAWFORD C. CHRIS CRAWFORD ON GAME DESIGNER (PAPERBACK). EDITORA NEW RIDERS GAME. EDIÇÃO: 1ª. ANO: 2003.
- [CHEN2001] CHEN, JANE. CAN ADVERGAME BE THE FUTURE OF INTERACTIVE ADVERTISEMENT? ANO: 2001.
- [IGDA2006] IGDA. CASUAL GAME WHITE PAPER. ANO: 2006.
- [FLOOD2003] FLOOD, MARK. 2003. GAME UNIFIED PROCESS. DISPONÍVEL EM: [HTTP://WWW.GAMEDEV.NET/REFERENCE/ARTICLES/ARTICLE1940.ASP](http://www.gamedev.net/reference/articles/article1940.asp). ACESSADO EM: SETEMBRO/2006.
- [DEMACHY2003] DEMACHY, T. EXTREME GAME DEVELOPMENT. DISPONÍVEL EM: [HTTP://WWW.GAMASUTRA.COM/RESOURCE_GUIDE/20030714/DEMACHY_01.SHTML](http://www.gamasutra.com/resource_guide/20030714/demachy_01.shtml). ACESSADO EM: SETEMBRO/2006.
- [SOMMERVILLE2003] SOMMERVILLE, I. SOFTWARE ENGINEERING. EDITORA ADDISON-WESLEY PRESS. EDIÇÃO: 6ª. PAG. 37-39. ANO 2003.
- [PMBOK2000] PROJECT MANAGEMENT INSTITUTE. PROJECT MANAGEMENT BODY OF KNOWLEDGE. ANO: 2000.
- [SCHWABER2004] SCHWABER, K. AGILE PROJECT MANAGEMENT WITH SCRUM. MICROSOFT PRESS. EDIÇÃO: 1ª. ANO: 2004.
- [XISPE2001] XISPÊ. 2001. DISPONÍVEL EM: [HTTP://WWW.XISPE.COM.BR](http://www.xispe.com.br) ACESSADO EM: AGOSTO/2006.
- [BEEDLE2004] BEEDLE, M. ET AL. AGILE DEVELOPMENT WITH SCRUM. EDITORA ADDISON-WESLEY PRESS. EDIÇÃO: 1ª. ANO: 2004.
- [COHN2004] COHN, M. USER STORIES APPLIED FOR AGILE SOFTWARE DEVELOPMENT. EDITORA ADDISON-WESLEY PRESS. EDIÇÃO: 1ª. ANO: 2004.
- [TEAMSYSTEM 2005] SCRUM FOR TEAM SYSTEM. DISPONÍVEL: [HTTP://WWW.SCRUMFORTEAMSYSTEM.COM](http://www.scrumforteamssystem.com) ACESSADO EM: AGOSTO/2006.
- [ARAUJO2006] ARAUJO, ALLAN. AGILE GAME PROCESS: METODOLOGIA PARA PROJETOS DE ADVERGAMES. TRABALHO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, CENTRO DE INFORMÁTICA, UNIVERSIDADE FEDERAL DE PERNAMBUCO. ANO: 2006.
- [OMG2005] OBJECT MANAGEMENT GROUP. SOFTWARE PROCESS ENGINEERING METAMODEL SPECIFICATION. DISPONÍVEL EM: [HTTP://WWW.OMG.ORG/TECHNOLOGY/DOCUMENTS/FORMAL/SPEM.HTM](http://www.omg.org/technology/documents/formal/spem.htm). ACESSADO EM: SETEMBRO/2006.