

# Proposta de método para gestão ágil da visão no desenvolvimento de jogos digitais

Rafael Sales

contato@rafaelsales.com.br

## Resumo

Todas as indústrias inovadoras enfrentam enormes desafios no desenvolvimento de seus produtos, pelos altos níveis de incerteza, custos e riscos. Novas abordagens de gestão foram criadas, principalmente as baseadas no Manifesto Ágil que propõe um design iterativo, onde Scrum é o método-chave. A indústria de desenvolvimento de jogos digitais utiliza esse novo paradigma há anos, porém ainda sofre com o conflito entre as abordagens ágil e tradicional, baseada no guia PMBoK e no desenvolvimento em cascata.

Este artigo propõe um método híbrido de gerenciamento da visão de jogos digitais, reunindo as melhores práticas das abordagens ágil e tradicional, reduzindo assim os altos níveis de incerteza sobre o conceito do jogo e consequentemente os riscos de fracasso nos projetos. Com isso as chances de entregar jogos de sucesso aumentam significativamente, atingindo as expectativas dos jogadores com jogos divertidos.

**Palavras-chave:** gestão, visão, produção, ágil, scrum

## Abstract

Every innovative industry deals with huge challenges in the development of their products, because of high levels of uncertainty and risks. New management approaches has created to improve this environment, specially the paradigm created with the Agile Manifesto that uses an iterative design, where Scrum is the most broadcast method. The game development industry has been using this new paradigm for years, but there is yet an abyss between the Agile and the Traditional ways, strongly based on the PMBoK guide and the waterfall development.

This article purposes a hybrid model of vision management for games, joining the best practices of agile and traditional approaches, decreasing the high levels of uncertainty about the game concept and consequently the risks of failure. Because of that the chances to deliver successful games increase, achieving the expectations of players with fun games.

**Key-words:** management, vision, production, agile, scrum

## I. INTRODUÇÃO

Desenvolver jogos digitais nunca foi uma tarefa fácil, nem mesmo em seus primórdios. Com o enorme crescimento da indústria nas últimas décadas, os projetos ficaram ainda mais complexos, especialmente na gestão, já que equipes, custos e riscos cresceram bastante. Além disso, a concorrência existente

em qualquer plataforma é enorme, exigindo que a qualidade dos jogos seja cada vez maior, já que os usuários estão mais exigentes.

De acordo com KEITH [1], os maiores problemas atuais da indústria de desenvolvimento de jogos digitais são:

- Equipes trabalhando em planos separados, o que gera entrega de resultados desconectados entre times de game design, programação, arte e áudio;
- O jogo sempre está quebrado, cheio de bugs e com pouco polimento;
- Estimativas e planos são sempre otimistas, gerando um ambiente de entregas atrasadas ou incompletas;
- Gerentes estão sempre “apagando incêndio” e nunca têm tempo de focar no projeto como um todo;

O artigo apresentará uma proposta de método para gestão da visão do produto, por ser o processo inicial de qualquer jogo, onde muitas falhas acontecem ao desenvolver um conceito desalinhado com as expectativas dos principais *stakeholders*. Com uma visão clara para todas as partes envolvidas, os problemas acima citados são minimizados, reduzindo assim o alto índice de projetos fracassados.

Segundo relatório CHAOS Manifesto de 2010 [2], a taxa de projetos bem-sucedidos aumentou de 32% para 37% enquanto que a taxa de projetos fracassados diminuiu de 24% para 21%. Segundo pesquisa, o *Standish Group* aponta quatro razões para a melhoria significativa encontrada em 2010 em relação a 2008, onde uma delas é o uso de processos ágeis.

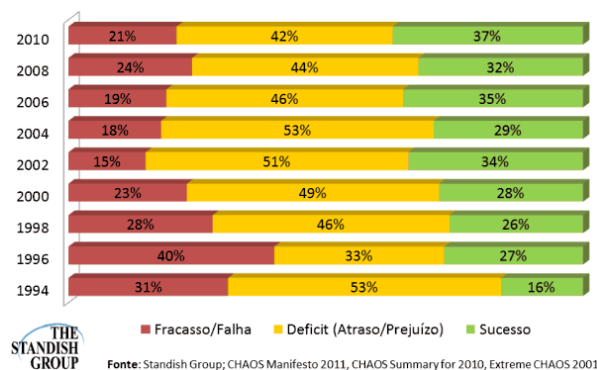


Figura 1: Relatório CHAOS Manifesto 2010 [2].

## II. TRABALHOS RELACIONADOS

### A. Agile Game Development with Scrum

KEITH [1] aborda de maneira profunda como utilizar Scrum no desenvolvimento de *games*. Ele apresenta técnicas alinhadas com as peculiaridades da indústria de desenvolvimento de jogos digitais, com questionamentos, ajustes e estudos de casos pertinentes a esta pesquisa.

Referente à visão de produto, o autor aprofunda discussões sobre o Scrum como uma ferramenta ideal para agregar valor ao negócio.

### B. Gerenciamento Ágil de Projetos: aplicação em produtos inovadores

AMARAL [3] compara as abordagens tradicional e ágil no gerenciamento de projetos, propondo um modelo referencial que reúna o melhor dos extremos. O modelo proposto pelos autores é extremamente completo, porém não é focado nas dificuldades comuns da indústria de desenvolvimento de games, como o trabalho de KEITH.

Os autores dedicam um capítulo inteiro para discussão da visão e sugerem um método de gestão ágil de visão, que serviu como inspiração para este artigo.

## III. GESTÃO DE PROJETOS TRADICIONAL E ÁGIL

### A. Abordagem Tradicional

O surgimento do gerenciamento de projetos (GP) aconteceu por volta de 1950 com a criação de um conjunto que reunia técnicas, ferramentas e conceitos, como o gráfico de Gantt, o método do caminho crítico e a análise PERT [4]. Entre as décadas de 1950 e 1970 foram criadas as primeiras associações profissionais, como o PMI e o IPMA, tornando-se os alicerces de qualidade de gestão de projetos no mundo. Nesse momento, essas técnicas foram aplicadas em grandes projetos de construção civil, defesa e aeroespacial [5], e hoje são fundamentais em todas as áreas de negócio, inclusive no ramo do entretenimento e em jogos digitais.

No final da década de 1990 surgiram os “corpos de conhecimento” (BOKs – *Bodys of Knowledge*), que reuniam as melhores práticas de gerenciamento de projetos após meio século de empirismo. O mais difundido é o “Guia PMBoK” desenvolvido pelo PMI (*Project Management Institute*) que apresenta práticas, técnicas e ferramentas que funcionaram de forma eficiente em projetos importantes no passado.

Esses “corpos de conhecimento” são chamados atualmente de literatura ou abordagem tradicional de gerenciamento de projetos. O PMI acredita que o PMBoK deve servir como guia para empresas que desenvolvem qualquer tipo de projeto (serviços, produtos, etc.) e para diferentes áreas, como construção civil, saúde, software, etc. Por outro lado, a maior crítica sobre a utilização da abordagem tradicional acontece no desenvolvimento de projetos inovadores, cujo produto possui características novas para a empresa e algumas vezes para a indústria e o mundo. Nesses casos o nível de incertezas é muito alto, sem parâmetros eficientes para comparação e sempre com

equipes inexperientes com as características inovadoras do produto.

Na indústria de TI o termo tradicional é usado para o desenvolvimento em cascata (*Waterfall*), onde a linha de produção é totalmente linear, sem margens para iterações durante o desenvolvimento, como a imagem a seguir.

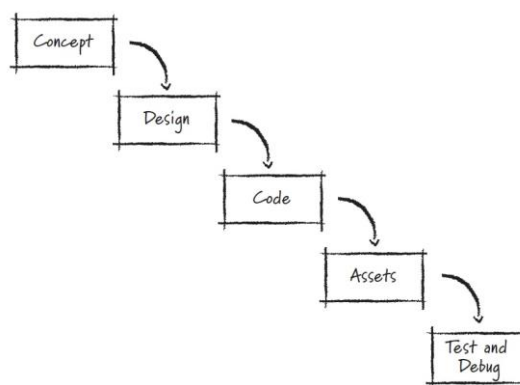


Figura 2: Desenvolvimento em cascata (waterfall) [1].

Recentemente o PMI incorporou o conceito ágil em seu corpo de conhecimento, criando uma certificação específica para gestão ágil de projetos, chamada PMI-ACP. Isso é só mais um sinal de que a união entre as abordagens tradicional e ágil podem ser harmoniosas.

### B. Abordagem Ágil

Diante desse cenário, um grupo experiente de desenvolvedores de software se reuniu em 2001 para discutir alternativas de gestão que se adequassem mais a esse tipo de projeto. Com isso foi criado o Manifesto Ágil [6], que é tratado como a base da literatura ou abordagem ágil, baseada nos seguintes princípios:

- Indivíduos e interações são mais importantes que processos e ferramentas:
  - A confiança nas pessoas é essencial para que um processo ágil seja eficiente, já que são elas que definem as melhores formas de desenvolver os produtos. São as pessoas que percebem as falhas dos processos e assumem a responsabilidade de melhorá-los;
- Software funcional é mais importante que documentação compreensível:
  - Documentações de programação, design e testes são importantes, mas um software funcional é o que de fato agrega valor ao negócio, pois permite que a organização analise o produto e guie o projeto ao sucesso;
- Colaboração do cliente é mais importante que negociação em contrato:
  - Na abordagem ágil, acredita-se que o cliente não deve permanecer distante como é

comum na abordagem tradicional. Espera-se que o cliente faça parte do desenvolvimento do produto, ajudando na comunicação com os principais *stakeholders* e na priorização das características mais valiosas do produto;

- Responder a mudanças é mais importante que seguir um plano:
  - A agilidade da comunicação é essencial no desenvolvimento ágil, tornando possível que todos envolvidos no projeto sejam informados de novidades que devam ser tratadas imediatamente, para garantir que o produto entregue esteja alinhado incrementalmente com as necessidades do usuário final;

É importante enfatizar que a abordagem ágil não exclui a importância e necessidade da abordagem tradicional. O manifesto deixa claro que as empresas precisam aliar as duas abordagens para encontrar o melhor processo possível para sua realidade, mas para realizar isso é necessária muita experiência para que esta tarefa não se torne um risco alto para o projeto.

Para utilizar a abordagem ágil eficientemente é necessário que duas questões sejam implementadas nas empresas: a reformulação conceitual da cultura organizacional para alinhamento com os princípios ágeis citados acima; e a implementação de um método ágil que reúna práticas, ferramentas e processos que viabilizem a utilização desse novo paradigma. O método mais difundido é o Scrum.

### C. Scrum

Dentro do jovem universo ágil, nenhum método se destaca tanto quanto o Scrum, principalmente por ser um processo focado no time e no desenvolvimento de produtos.

O Scrum começa quando um *stakeholder* precisa de um produto. O método é utilizado em empresas de diversos portes e projetos de altos níveis de complexidade e para que seja eficientemente utilizado é necessário que os times entendam os princípios ágeis e as técnicas propostas pelo Scrum.

A principal instituição que rege a evolução do Scrum é a *Scrum Alliance*, que criou recentemente o *Agile Atlas* [7], uma enciclopédia online que reúne todas as explicações necessárias para a eficiente utilização do método.

Antes de tudo, é crucial entender os principais papéis do Scrum:

- *Product Owner* tem a responsabilidade de decidir o que será desenvolvido e em qual prioridade. Na indústria de *games* é o Diretor do jogo;
- *Scrum Master* é o líder servidor que ajuda o time e a empresa a usar o Scrum da melhor maneira possível. Na indústria de desenvolvimento de jogos é o Produtor;
- *Development Team* desenvolve o produto de forma incremental, em uma série de ciclos curtos, chamados Sprints, com duração média de 2 a 4 semanas. No final

de cada Sprint o time precisa entregar uma versão nova do jogo;

No Scrum, as responsabilidades do gerente de projetos tradicional são divididas entre os três papéis acima. Contudo, para que isso funcione, é necessário que o time de desenvolvimento tenha maturidade para realizar autogestão, para que o Produtor consiga ter tempo de focar nas tarefas de *Scrum Master* e o Diretor possa focar nas tarefas de *Product Owner*.

A figura abaixo mostra o funcionamento do Scrum, que será explicado em detalhes abaixo.

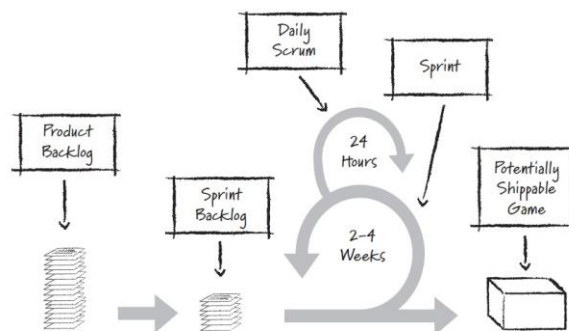


Figura 3: Scrum [1].

Os artefatos abaixo são essenciais para a utilização do Scrum:

- *Product Backlog* é a lista priorizada de *features*, organizada para guiar a sequência de desenvolvimento do jogo. O responsável pelo *Product Backlog* é o Diretor. Esse artefato será tratado em mais detalhes nos próximos capítulos;
- *Sprint Backlog* é o plano detalhado para o desenvolvimento do produto em um Sprint;
- Produto incremental é o resultado esperado no final de cada Sprint, ou seja, uma versão do jogo com novidades relevantes;

Existem cinco atividades ou reuniões essenciais no Scrum:

- *Product Backlog Refinement* é o processo de manter o *Product Backlog* atualizado e detalhado o suficiente para ser utilizado nas reuniões de *Sprint Planning*;
- *Sprint Planning* é a reunião que acontece sempre no início de cada Sprint, onde o diretor explica os itens mais importantes do *Product Backlog* e o time de desenvolvimento estima o que é viável desenvolver no período determinado para o Sprint. O produtor lidera essa reunião, tornando-a o mais eficiente possível;
- *Daily Scrum* é uma reunião diária onde todos os membros do time de desenvolvimento devem dizer o que fizeram ontem, o que farão hoje e se existem impedimentos para concluir suas tarefas. O Produtor lidera a reunião e ajuda o time a limpar qualquer impedimento que surja;

- *Sprint Review* é a reunião que acontece no final de todos *Sprints*, onde o time deve apresentar a nova versão do jogo para o Diretor, que verifica se as novidades atendem suas expectativas e principalmente as necessidades e objetivos do projeto;
- *Sprint Retrospective* é uma reunião onde o time se reúne para identificar pontos de melhoria no processo de desenvolvimento de jogo. Acertos e erros são levantados, além de possíveis soluções para resolver esses erros são propostas pelo time;

Chegando neste ponto, é hora de repetir tudo. Essa é a essência do design iterativo: repetir o processo em ciclos curtos para descobrir se as suposições iniciais estavam certas. Após algumas sessões, o time ganha experiência e confiança no processo de produção.

#### IV. GERENCIANDO A VISÃO

Visão é um termo muito usado na literatura de planejamento estratégico, no sentido de visão de negócio, que ajuda a nortear para onde se deve seguir. Kotter afirma que “se uma pessoa não conseguir comunicar a visão e obter uma reação de entendimento e interesse de alguém em cinco minutos ou menos, as organizações e pessoas envolvidas não estarão prontas para as transformações necessárias”[8].

Na abordagem ágil, a visão do produto serve para obter um senso comum e orientar o trabalho da equipe, similar a visão corporativa para o conjunto de ações da empresa. No Scrum, o responsável pela visão do produto é o *Product Owner*, ou Diretor.

##### A. Visão e Escopo

Segundo o PMBoK [9], existem dois tipos de escopo: do produto e do projeto. O escopo do produto são as características e funções que descrevem um produto, serviço ou resultado. O escopo do projeto é o trabalho que precisa ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas. Para fins de comparação com a definição de visão acima, o tipo de escopo que mais interessa a esta pesquisa é o escopo do produto.

Como já visto, a visão do produto está conectada a abordagem ágil, assim como o escopo do produto está relacionado à abordagem tradicional. Segundo AMARAL [3], para definir a visão na abordagem tradicional é utilizada uma grande quantidade de documentos com muitos textos e poucos ícones visuais. Mesmo sendo previsto que a visão seja revista na fase de controle e monitoramento, o time não é estimulado a iterar essa visão, já que o desenvolvimento em cascata não prevê tempo para iterações.

Já na abordagem ágil a criação da visão é encontrada por todo processo de desenvolvimento do jogo, já que as equipes são encorajadas a rever e, se preciso for, mudar a visão do produto durante o desenvolvimento, desde que não fujam ao objetivo do projeto. Além disso, muitos recursos visuais são utilizados para facilitar a assimilação da visão do produto.

A tabela a seguir detalha mais essa comparação entre as abordagens ágil e tradicional na construção da visão:

<b>Categoria</b>	<b>Ágil</b>	<b>Tradicional</b>
<b>Nome do grupo de processos</b>	Fase Visão.	Grupos de processos Iniciação e Planejamento.
<b>Orientação (foco)</b>	Foco no problema. Descrever o problema central de maneira motivadora e motivar, explicitando os desafios e compromissos que precisarão ser “equacionados”.	Foco nas metas de desempenho. Descrever as “regras do jogo”, isto é, condições e metas que deverão ser alcançadas no final.
<b>Principal resultado</b>	Visão – descrição com ambiguidades, metáforas, analogias e conceitos integradores que desafiam mais do que indicam o resultado.	Termo de abertura do projeto. Declaração do escopo do projeto. EAP (atualização).
<b>Práticas de descrição</b>	“Caixa” para Visão do Produto. Declaração de Alto Nível. Arquitetura do produto. Cartões de características do produto. Cartões de requisitos de desempenho.	Elaboração de EAP. Declarações textuais em formato de documentos.
<b>Documentos gerados no início do projeto</b>	Folha de Dados do Projeto. Plano de Entregas.	Termo de Abertura do Projeto. Declaração de Trabalho. Declaração de Escopo. EAP.

Tabela 1: Diferenças da visão entre ágil e tradicional, segundo AMARAL [3].

A seguir será apresentado o método, que pretende contribuir para a construção da visão do produto, reunindo o melhor das abordagens ágil e tradicional.

#### V. PROPOSTA DE MÉTODO

O método será chamado de *Agile Game Vision Management Method*, principalmente pela sigla AGVMM. Seu objetivo é fazer com que a visão seja trabalhada de modo eficiente por todo o processo de produção do jogo, oferecendo um método eficiente para o máximo de tipos de projetos.

O método é dividido em cinco partes, conforme figura abaixo:



Figura 4: Método AGVMM, criado pelo autor. As setas laranjas mostram os pontos com maior uso de iterações.

Cada parte será explicada a seguir, com as devidas explicações de conceitos novos, como *User Stories* e *One Page Design*.

#### A. Termo de Abertura

A principal função deste termo é marcar o início do projeto e alinhar os pontos cruciais para que este início seja harmônico. O responsável por esta fase é o Produtor e para tal, precisará manter contato com todas as partes envolvidas para preencher o documento.

A intenção de iniciar o método com um documento mais próximo da abordagem tradicional é um sinal de que o Scrum e a abordagem ágil precisam de uma ferramenta mais eficaz para realizar o *kickoff* do projeto.

Ser conciso é essencial neste termo, já que o ideal é que este documento ocupe apenas uma página A4, mantendo assim o princípio ágil da simplicidade, podendo inclusive ser colado na parede da empresa para que todos envolvidos possam consultar caso surjam dúvidas.

Termo de Abertura			
<b>Título</b>	Nerds vs Monsters	<b>Data</b>	14/11/12
<b>Apelido</b>	NvM		
<b>Produtor</b>	Mario Bros		
1 - Principais Stakeholders			
<b>Cliente</b>	<b>Nomes</b>	<b>Iniciais</b>	<b>Função</b>
<i>Publisher Fictícia</i>	Ana Paula	AP	CEO
<b>Time</b>			
	Mario	MB	Produtor
	Carmem	CS	Diretora
	Camilla	CC	Game Designer
	Magnum	MM	Diretor de Arte
	Janaina	JV	Diretora Técnica
2 - High Concept			
Seu nome é Gomes e ao sair de casa depois de uma <i>nonstop</i> de 3 dias jogando CS, você percebe que o mundo não é mais o mesmo. Zumbis, alienígenas e outros monstros estão lutando pelo domínio da terra e somente você e seus amigos nerds podem salva-la.			
3 - Unique Selling Points			
<ul style="list-style-type: none"> <li>Batalha inovadora entre nerds e vários monstros;</li> <li>Armas divertidas e engraçadas do arsenal nerd;</li> <li><i>Multiplayer</i> viciante;</li> </ul>			
4 - Escopo negativo			
O jogo será desenvolvido apenas para a plataforma PC e <i>ports</i> para outras plataformas precisarão ser negociadas futuramente com o investidor.			
5 - Potenciais concorrentes			
<ul style="list-style-type: none"> <li>Jogo A;</li> <li>Jogo B;</li> <li>Jogo C;</li> </ul>			
5 - Fornecedores de ideias		6 - Responsável pelo <i>Product Backlog</i>	
Mario Carmem Camilla Magnum Janaina		Carmem	

Tabela 2: Termo de Abertura criado pelo autor.

Abaixo segue o esclarecimento de alguns campos do termo de abertura:

- High Concept* é uma sentença curta que provê a visão do jogo. É usualmente a primeira etapa da definição do conceito do jogo e por conta disso precisa ser simples e fácil de entender. Se você não consegue explicar a visão do jogo em poucas palavras, provavelmente ela está complicada ou vaga demais para se tornar um jogo de sucesso [10];
- Unique Selling Points (USPs)*: um bom conceito de jogo deve ter características que atraiam clientes. Uma regra de ouro é que qualquer conceito de jogo deve ter pelo menos três pontos únicos de venda (*USPs*). Os

jogadores comumente buscam essas características na caixa do jogo antes de compra-los [10];

- Escopo negativo delimita o que com certeza não faz parte do escopo do projeto e precisa ser esclarecido, pois seu custo de produção excede o esperado pelo time de desenvolvimento;

Com o termo de abertura resolvido, é necessário entender melhor as necessidades e desejos dos jogadores.

### B. User Stories

O foco desta fase é o levantamento de necessidades e desejos do jogador, além de outros requisitos do jogo. O responsável por esta fase é o Diretor, principalmente porque esse levantamento servirá como base para o *Product Backlog*.

Comunicar abstrações como os desejos dos jogadores é algo complexo e para isso foi criada a técnica de *User Stories* por Kent Beck para expressar o valor de certas características para o consumidor [1]. A quebra de paradigma dessa técnica é fugir do foco na *feature* e focar na experiência do usuário, como HUNICKE [11] explica em seu prestigiado artigo sobre o *framework* MDA.

Histórias de usuários são construídas através da seguinte estrutura, recomendada por Mike Cohn:

Como <usuário>, quero <meta> porque <razão>.

- Usuário: consumidor do jogo ou membro do time de desenvolvimento do jogo;
- História: meta da história. É uma característica ou função do jogo, ferramenta ou processo de produção;
- Razão: benefício que o usuário terá caso essa história seja entregue;

Segue abaixo amostras de histórias de usuários:

Levantamento de User Stories					
Título	Nerds vs Monsters		Data	18/11/12	
Apelido	NvM				
Produtor	Mario Bros				
Diretora	Carmem Silva				
ID	Usuário	História	Razão	Critérios de Aceitação	Ref.
01	Como jogador	Preciso usar armas potentes	Para matar monstros	- variação interessante - progressão divertida - visual atrativo - SFX atrativos	
02	Como jogador	Gostaria de jogar contra meus amigos	Para me divertir jogando na internet	- mecânicas divertidas - diferentes modos de multiplayer	-
...	...	...	...	...	...

Tabela 3: Levantamento de User Stories.

Cada história tem sua lista de critérios de aceitação e referências. É crucial que esses critérios sejam bem escritos de acordo com os conceitos do INVEST [1]: independente, negociável, valiosa, estimável, com tamanho apropriado e testável. Além disso, a utilização de imagens para referências contribui para o entendimento das histórias, tornando a comunicação mais eficiente.

Essas histórias de usuário serão utilizadas em todo o método, sendo o elemento básico do *Product Backlog*, artefato-chave deste método. Por conta disso as histórias não precisam ser quebradas em menores partes possíveis neste momento, pois o time ainda não tem informações suficientes para isso. Nas fases seguintes o time descobrirá mais detalhes sobre o jogo e o *Backlog* será atualizado.

Segundo KEITH [1] as histórias têm dois níveis de detalhe acima delas:

- Épico: histórias muito grandes para serem desenvolvidas em um Sprint, que precisarão ser quebradas em partes menores para entrar em produção;
- Saga: alguns projetos precisam de um nível ainda maior que um épico, chamado de saga, para *features* mais complexas;

É esperado que nesta fase o *Backlog* tenha um número significativo de épicos.

Outro recurso interessante das histórias de usuários são os Temas, que servem para relacionar as histórias em grupos que ajudam nas estimativas no *Sprint Planning*.

### C. Conceito

Recapitulando as fases do método, no termo de abertura entendemos o *high concept* do jogo e as *USPs*. No levantamento de *user stories* entendemos de forma mais detalhada as expectativas dos jogadores e agora é o momento de definir uma proposta de conceito do jogo. O responsável por esta fase é o time de desenvolvimento, em especial o Game Designer do projeto.

Realizar uma discussão profunda sobre técnicas vigentes de conceito de jogo foge do escopo desta pesquisa, dado seu foco no gerenciamento da visão, não no desenvolvimento do conceito em si. Por esta razão, foi escolhida uma técnica simples, validada e coerente com os princípios da abordagem ágil: *One Page Design*. A técnica propõe uma apresentação do game design mais visual e simples, fugindo dos complexos e extensos *Game Design Documents (GDDs)*, também chamados de *Game Bibles*. Vale reforçar que *GDDs* são associados com a abordagem tradicional no universo do design, assim como o *Waterfall* para a programação.

Segundo LIBRANDE [12], criador da técnica, ao utilizar a técnica o time inteiro se beneficia, principalmente porque o design é efetivamente lido por todos os membros do time, facilitando inclusive a oportunidade de contribuições de outros membros do time que não sejam designers. Já para o designer os benefícios são a mudança de paradigma para um design mais conciso e extremamente fácil de entender pelo seu time.

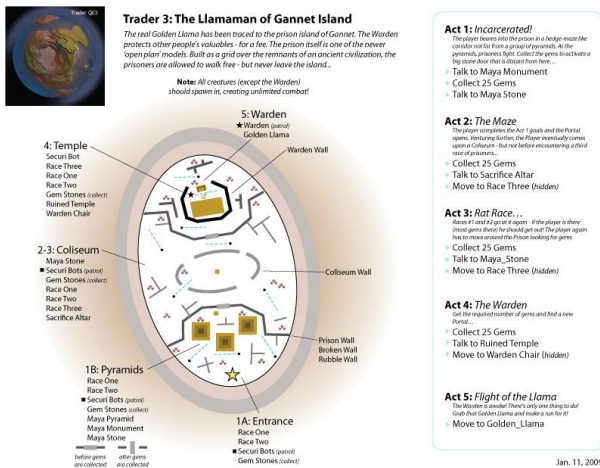


Figura 5: Exemplo de One Page Design.

Com a primeira proposta de conceito pronta, o designer apresenta ao Diretor e ao restante do time, que fazem suas devidas considerações. Nesse momento o processo precisa iterar quantas vezes seja necessário até que o time, e principalmente o Diretor, esteja satisfeito com o conceito do jogo.

Com o conceito resolvido, partimos para o *Product Backlog*.

**D. Product Backlog**

Para encerrar essa etapa inicial de criação da visão do jogo, é necessário retomar as *User Stories* e preencher o *Product Backlog* de acordo com as descobertas realizadas na fase de conceito. A principal diferença da 2ª fase para esta é o nível de incerteza que é bem menor. Sendo assim, o Diretor consegue escrever o *Backlog* do jogo com muito mais segurança quanto à coerência das histórias com as necessidades e desejos do jogador, além de alinhadas com as expectativas dos principais *stakeholders*.

Nesse momento, o Diretor precisa se comunicar com os principais *stakeholders* do projeto para validar o *Backlog*. É importante que o investidor esteja sempre próximo do desenvolvimento, porém raramente esse executivo tem tempo hábil para acompanhar as iterações mínimas do projeto. Por conta disso é crucial que o conceito do jogo e o *Product Backlog* sejam validados pelo patrocinador antes de liberar o *greenlight* para a produção.

A estrutura do *Backlog* não deve se limitar a apresentada a seguir, cabendo ao Diretor e ao time adicionar as colunas que julgar relevantes, como Temas, Valor e Sprint.

Seguem algumas práticas que se mostraram eficientes ao autor através do empirismo, sem fundamentação teórica:

- Usar a coluna de valor para priorizar os itens do *Backlog* de acordo com os mais importantes para o próximo Sprint, ou seja, os itens que agregam mais valor. Ao ordenar os itens pelos valores mais altos, o *Backlog* estará pronto para os próximos planejamentos;

- Compartilhar a planilha do *Product Backlog* com todos os membros do time é uma boa prática, facilitando a retirada de dúvidas sobre as histórias e a verificação dos critérios de aceitação durante o desenvolvimento das *features* nos *Sprints*. É importante que apenas o Diretor tenha autorização para alterar a planilha, dando aos demais membros do time apenas o acesso para leitura;

Segue o exemplo de *Backlog*:

Product Backlog					
Título	Nerds vs Monsters		Data	15/01/13	
Apelido	NvM				
Produtor	Mario Bros				
Diretora	Carmem Silva				
ID	Usuário	História	Razão	Critérios de Aceitação	Ref.
01	Como jogador	Preciso usar armas potentes	Para matar monstros	- variação interessante - progressão divertida - visual atrativo - SFX atrativos	
02	Como jogador	Gostaria de jogar contra meus amigos	Para me divertir jogando na internet	- mecânicas divertidas - diferentes modos de multiplayer - latência aceitável da internet	-
...	...	...	...	...	...
98	Como artista	Preciso de uma ferramenta para testar animações no jogo	Para agilizar a pipeline de arte	- fácil de usar - rápida - estável	Plugin da Unity

Tabela 4: *Product Backlog*.

Com o *Backlog* montado, a produção pode começar.

**E. Desenvolvimento**

Com o fim da pré-produção, o foco do projeto é transferido da criação da visão para o desenvolvimento e manutenção da mesma. Ao implementar as *features*, é crucial verificar frequentemente seu alinhamento com a visão, para validar se o valor de negócio esperado está sendo entregue. Isso precisa ficar claro para todos os membros do time, pois caso não seja tomado esse cuidado, o time pode cair na armadilha do desenvolvimento em cascata e só desenvolver o que foi pensado na pré-produção.

Para isso é necessário que ao fim de cada Sprint o time se reúna e verifique se as *User Stories* desenvolvidas agregaram valor ao negócio, ou em outras palavras, tornou o jogo um produto mais interessante ao público-alvo. Mesmo tendo oportunidade de iterar no dia-a-dia, o principal momento de iteração da visão é na reunião de *Sprint Review* no final do

*sprint*, onde o time verifica se o jogo está alinhado com as expectativas criadas ao escrever as histórias de usuários.

Como o processo de game design iterativo é baseado na descoberta da diversão, e não apenas em sua criação prévia completa na pré-produção [13], o *Sprint Review* ganha uma importância enorme, por ser a principal ferramenta para manutenção da visão alinhada com as expectativas dos principais *stakeholders*.

Além disso, o Diretor precisa manter o *Backlog* atualizado constantemente, para que seja utilizado de maneira eficaz nas reuniões de *Sprint Planning*. Um bom *Product Owner* sempre tem o *Backlog* com histórias para três *sprints* à frente. Essa prática de antecipação do planejamento é chamada de *Release Planning* [1], onde o diretor planeja entregas maiores que normalmente são entregues para *stakeholders* que não podem participar de maneira próxima do desenvolvimento, como *publishers*, investidores, jogadores, etc.

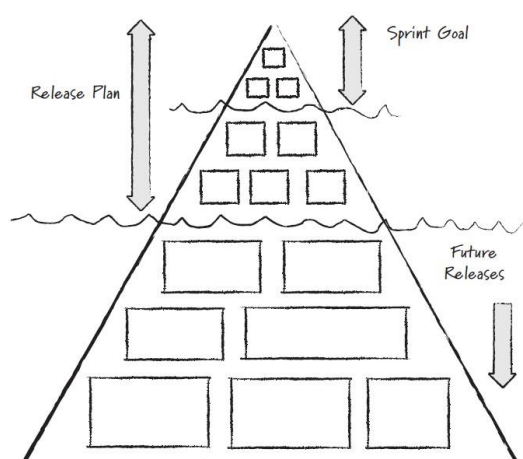


Figura 6: *Release Planning*.

## VI. CONCLUSÃO

O tema tratado neste artigo é extremamente rico e precisa de mais publicações com adaptações para as vastas peculiaridades da indústria de desenvolvimento de jogos digitais. Com certeza o método AGVMM proposto neste artigo precisa ser testado para ser validado. Contudo o *framework* se mostrou como uma possibilidade interessante para melhorar a fase inicial dos projetos de desenvolvimento de *games*, onde existem altos níveis de incerteza e novas abordagens precisam ser criadas para minimizar os altos riscos de fracasso nos projetos presentes na indústria.

Este artigo é apenas o começo de uma pesquisa que o autor pretende levar adiante para mestrado e doutorado, visando à criação de um modelo referencial completo para gerenciamento ágil de desenvolvimento de jogos digitais, da criação da visão inicial até o lançamento do jogo e manutenção do mesmo.

Existem algumas oportunidades interessantes para trabalhos futuros nesta pesquisa. A implementação do *framework* em um número significativo de empresas para criar cases de uso sem dúvidas é a mais importante, apresentando assim resultados reais na indústria. Além disso, é possível inserir algum

processo de prototipagem na 3ª fase (conceito), o que pode aumentar a eficiência do método, por ter mais chances de reduzir as incertezas. Também é possível adaptar o *framework* para um vasto leque de possibilidades, como empresas e projetos de tipos e tamanhos distintos, até mesmo com o uso em outras indústrias criativas como software, cinema, televisão, publicidade, etc.

## VII. REFERÊNCIAS

- [1] KEITH, Clinton. Agile Game Development with Scrum. Boston: Addison Wesley, 2010.
- [2] The Standish Group: CHAOS Summary for 2010. Disponível em <<http://insyght.com.au/special/2010CHAOSSummary.pdf>>. Acesso em 10 jul. 2013.
- [3] AMARAL, Daniel Capado; CONFORTO, Edivandro Carlos; BENASSI, João Luís Guilherme; ARAÚJO, Camila de. Gerenciamento ágil de projetos: aplicação em produtos inovadores. São Paulo: Saraiva, 2011.
- [4] KERZNER, H. Project Management: a system approach to planning. Scheduling and controlling. Nova Iorque: Van Nostrand Reinhold Company, 1984.
- [5] MAYOR, H. Beyond the Gantt chart: Project management moving on. European Management Journal, v. 19, n. 1, p. 92-100, 2001.
- [6] BECK, Kent et al. Manifesto for Agile Software Development. 2001. Disponível em: <http://www.agilemanifesto.org/>.
- [7] Agile Atlas. Disponível em <[www.agileatlas.org](http://www.agileatlas.org)>. Acesso em 02 jul. 2013.
- [8] KOTTER, J.P. Leading change: Why transformation efforts fail. Harvard business review, v. 73, p. 59-67, 1995.
- [9] ---. Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK). 4ª edição. Pennsylvania, EUA: PMI, 2009.
- [10] HIGHT, John; NOVAK, Jeannie. Game Development Essentials: Game Project Management. Nova Iorque: Delmar Cengage Learning, 2008.
- [11] HUNICKE, R.; LE BLANC, M.; ZUBEK, R., 2004. MDA: a formal approach to game design and game research. Proc. GDC, San Jose.
- [12] LIBRANDE, S., 2010. One-page Designs. Proc. GDC, San Francisco.
- [13] PERRY, L., 2010. Prototype based design: a better, faster way to design your games. Proc. GDC, San Francisco.