

Pontifícia Universidade Católica de São Paulo  
PUC-SP

Thiago Cazelatto Breves

**Análise comparativa de requisitos e  
documentações de Jogos Digitais e de  
Softwares Tradicionais**

CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

São Paulo

2016



Thiago Cazellatto Breves

**Análise comparativa de requisitos e documentações de  
Jogos Digitais e de Softwares Tradicionais**

Trabalho Final apresentado à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de ESPECIALISTA em ENGENHARIA DE SOFTWARE, sob a orientação do Prof. Ms. Mario Madureira Fontes.

São Paulo

2016



# Sumário

<b>Sumário</b> . . . . .	<b>3</b>
<b>Lista de ilustrações</b> . . . . .	<b>5</b>
<b>Introdução</b> . . . . .	<b>15</b>
Organização do Trabalho . . . . .	15
<b>Capítulo 1</b> . . . . .	<b>17</b>
Definindo um <i>videogame</i> . . . . .	17
Desenvolvedores de <i>games</i> . . . . .	18
<b>Capítulo 2</b> . . . . .	<b>21</b>
Documento de <i>Game Design</i> . . . . .	21
Registro dos Requisitos . . . . .	22
<b>Capítulo 3</b> . . . . .	<b>25</b>
A documentação . . . . .	25
O método de trabalho . . . . .	26
<b>Conclusão</b> . . . . .	<b>29</b>
Considerações finais . . . . .	29
Trabalhos futuros . . . . .	29
<b>Anexos</b> . . . . .	<b>31</b>
Entrevista . . . . .	31
<b>Referências</b> . . . . .	<b>33</b>



# Lista de ilustrações

Figura 1 – Gráfico de estados mentais. Csikszentmihalyi (1991) . . . . .	19
Figura 2 – Exemplo de especificação visual. L. J. Souza, Mittelbach, and Neves (2010) . . . . .	22
Figura 3 – Imagem do jogo Oniken. Dias, Paiva, and Weiller (2012) . . . . .	26



# Abstract

This paper presents a comparison of the known practices of software engineering with management practices applied by game development teams in Brazil. Therefore are exposed definitions of games and some non-functional requirements that are of great importance for the design of this kind of software, as well as some of the peculiarities of the team that develop them and the most widely used method of documentation in the games area.

**Keywords:** documentation. games. digital games. non-functional requirements. software.



# Resumo

Este trabalho apresenta uma comparação das práticas conhecidas da área de engenharia de software com as práticas de gerenciamento aplicadas por equipes de desenvolvimento de jogos digitais no Brasil. Para tanto são expostas definições de jogos e de alguns requisitos não-funcionais que são de grande importância para a concepção deste tipo de software, assim como algumas das peculiaridades das equipes que os desenvolvem e o método de documentação mais utilizado na área de games.

**Palavras-chave:** documentação. games. jogos digitais. requisitos não-funcionais. software.



# Dedicatória

Dedico este trabalho a todos aqueles que se aventuram ou pensam em se aventurar no desenvolvimento de jogos digitais no Brasil.



# Agradecimentos

Agradeço ao Adrian Laubisch do Estúdio Aiyra, Vicente Filho da Manifesto Games e Thais Weiller da JoyMasher que colaboraram com a minha pesquisa e a Guilherme Sadao Ymanaka que me indicou as pessoas certas para me ajudar.



# Introdução

A indústria de *videogames*<sup>1</sup> gira um faturamento de cerca de 100 bilhões de dólares ao ano e segundo STATISTA (2015) esse faturamento é utilizado na produção de grandes jogos comerciais que possuem grandes equipes de desenvolvimento ou jogos independentes que possuem em sua essência uma equipe reduzida. As empresas dos jogos independentes tentam cada vez mais conquistar uma porção do mercado de jogos digitais.

Seja em grandes empresas ou em pequenos estúdios o jogo digital é um *software*, como apontado por Callele, Neufeld, and Schneider (2006), Callele, Neufeld, and Schneider (2005), Alves, Ramalho, and Damasceno (2007), L. Souza et al. (2009), L. J. Souza, Mittelbach, and Neves (2010) e vários outros escritores da área, seu processo de desenvolvimento difere bastante de um *software* tradicional, fazendo com que os formatos de documentação tradicionais não funcionem tão bem na construção de um *software* que é um jogo digital.

Portanto, para que os desenvolvedores realizem a documentação, é necessário fazer adaptações ou buscar alternativas. Por conseguinte, os novos estúdios que querem entrar neste mercado possuem dificuldade de acesso às informações dos processos e adaptações utilizados pelos mais experientes, tendo que passar por todo o processo de adaptação, que pode custar alguns anos de prática e fracassos comerciais.

Com o auxílio das pesquisas de Callele, Neufeld, and Schneider (2005), Callele, Neufeld, and Schneider (2006), Alves, Ramalho, and Damasceno (2007), Lemes (2015). A maioria das publicações tratam da documentação do processo desenvolvimento de *software* dos *games* como uma elicitación de requisitos não-funcionais. Este trabalho aponta algumas principais diferenças entre o processo de extração de requisitos tradicional de *software* com o que é praticado por alguns estúdios brasileiros. Um exemplo de caso dessa estudo é utilizado é mostrado com o processo com o jogo Oniken da Joymasher.

## Organização do Trabalho

O trabalho é dividido em 4 capítulos, sendo 3 capítulos para apresentação do estudo sobre documentação de software em jogo digitais e o último capítulo para as considerações finais e reflexões sobre o trabalho.

No primeiro capítulo é definido as competências de desenvolvimento em equipe de um jogo digital e quais são os principais elementos que compõe essas competências de desenvolvimento de um jogo digital.

---

<sup>1</sup> Utilizado como sinônimo para a expressão "jogos digitais".

No segundo capítulo é apresentado como é atualmente realizado a documentação de uma produção de um *videogame* comparando os pontos fortes e pontos fracos. Também é apresentado neste capítulo peculiaridades da extração de requisitos na área de documentação do software de um jogo digital.

No terceiro capítulo são apresentadas algumas práticas de documentação e os métodos de desenvolvimento de *software* de algumas das equipes de desenvolvimento de jogos atuantes no Brasil.

# Capítulo 1

A seguir serão expostas algumas definições e considerações dos jogos digitais e as equipes que os produzem, assim como comparações com os seus equivalentes na área de *software* tradicional, para que possamos entender melhor suas peculiaridades.

## Definindo um *videogame*

O *videogame* é definido por Callele, Neufeld, and Schneider (2006)., como “um produto de entretenimento multimídia que requer participação ativa do usuário” e como qualquer produto de entretenimento é supérfluo, ou nas palavras de Alves et al.: “ninguém tem obrigação de usar um *videogame*”, portanto o jogo deve ser pensado de forma a atrair público, ser pensado para agradar, como dito por Hassenzahl, Beu, and Burmester (2001), considerando elementos hedônicos como estilo, cores e música. Callele, Neufeld, and Schneider (2006) ainda diz:

O *videogame* possui uma parte software, com um conjunto de requisitos funcionais para sua implementação, uma jogo, com seus requisitos funcionais que criam o envolvimento cognitivo do jogador, e um conjunto de requisitos não funcionais que representam os estados emocionais induzidos ao jogador em cada estágio do jogo ou por cada elemento do jogo. Callele, Neufeld, and Schneider (2006)

A “parte *software*” é o que está por trás dos elementos de interação entre jogador e jogo, ou seja, a interface entre homem e máquina, que é definida por cálculos de comandos e retorno, seja este visual (exibição das imagens na tela), sonoro (músicas, falas, áudios), sensorial (vibrações, choques), ou qualquer outro meio que o equipamento, ou conjunto de equipamentos, onde o jogo está sendo executado permita.

A “parte jogo” é o ambiente em que o jogador está imerso, que inclui sua motivação, seu objetivo, elementos narrativos, entre outros, e as regras às quais ele está preso. Em jogos de mesa a maior parte desses elementos eram passados ao jogador por meio de manuais de instrução e referências visuais, deixando o desenvolver do jogo por conta de sua imaginação, uma vez que os elementos do jogo são processados por computador, é possível criar ambientes virtuais, narrativas que são contadas conforme a progressão do jogo, alteração de estado em tempo real e maior quantidade e complexidade de regras. Em jogos de mesa são comuns jogos por turno, jogos para vários jogadores e jogos onde o ambiente se mantém independente da atividade do jogador, barreiras que são facilmente

quebradas nos videogames (ou até mesmo invertidas, uma vez que é mais frequente vermos *games single player*<sup>2</sup> do que um *multiplayer*<sup>3</sup>).

A parte emocional é justamente aquilo que o *game designer* imagina que o jogador deve sentir durante seu jogo, Callele, Neufeld, and Schneider (2006) dá como exemplos: apreensão ao se aproximar de uma sala desconhecida e a adrenalina de estar dirigindo a alta velocidade e encontrar uma curva fechada a frente ou a felicidade de encontrar um tesouro no final da sua jornada. Um jogo que não produz estas variações emocionais se torna monótono e desinteressante. Fullerton (2008) afirma que o *game designer* deve produzir um jogo que seja rico emocionalmente, provoque empatia no jogador, e o impacte individual e culturalmente.

Um conceito fortemente ligado a esses estados emocionais é o conceito de Flow definido por Csikszentmihalyi (1991) como um estado mental onde o indivíduo está completamente envolvido com a atividade que está exercendo, apresentando características como concentração intensa, perda da autoconsciência, distorção da percepção temporal, etc. Estar no estado de Flow produz uma satisfação no indivíduo pela execução da atividade, porém para alcançá-lo deve haver um equilíbrio entre o nível de desafio que a tarefa representa e a habilidade que a pessoa tem para exercer esta tarefa. Naturalmente os níveis necessários para atingir a zona de Flow variam para cada indivíduo, mas se manter nela depende, invariavelmente do equilíbrio entre estes dois elementos (desafio e habilidade) e é aí que se encontra um dos desafios dos desenvolvedores de *games*, manter esse equilíbrio e o jogador envolvido na atividade de jogar o *videogame*, considerar, por exemplo, que conforme o jogador avança no jogo ele irá se acostumando às mecânicas e melhorando suas habilidades, fazendo com que o desafio que o jogo proporciona tenha que crescer de forma equivalente. Como mostrado no gráfico da figura a seguir, o desequilíbrio pode jogar o indivíduo no tédio ou na frustração, o que faria um jogador abandonar um jogo, o que resultaria em um fracasso comercial.

## Desenvolvedores de *games*

Para atender a essa demanda de considerações a serem feitas durante o desenvolvimento de um *game* são necessários profissionais de várias áreas do conhecimento, Callele, Neufeld, and Schneider (2005) aponta que uma equipe de desenvolvedores pode ser formada por cientistas da computação, engenheiros, artistas, músicos, designers e até mesmo psicólogos. Vale citar que desenvolvedores independentes não podem contar com uma equipe muito grande e frequentemente acabam por reunir várias dessas competências em um único profissional ou delegar algumas decisões ao consenso da equipe.

---

<sup>2</sup> Jogo para apenas um jogador.

<sup>3</sup> Jogo para dois ou mais jogadores.

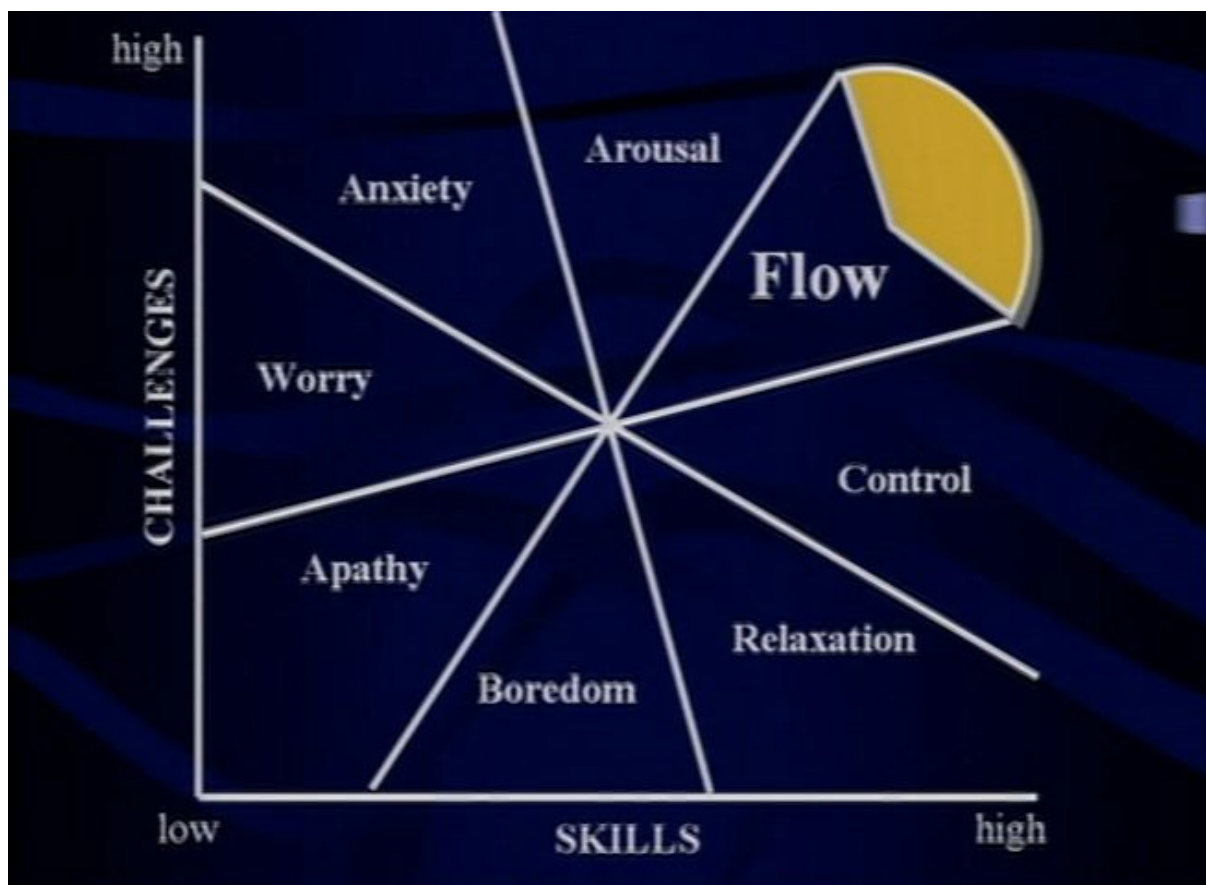


Figura 1 – Gráfico de estados mentais. Csikszentmihalyi (1991)

Uma equipe de desenvolvedores de *software* tradicional é geralmente composta inteira ou quase inteiramente por profissionais da área de Tecnologia da Informação. A vantagem desta estrutura é que todos possuem uma linguagem comum e um certo entendimento do produto em que estão trabalhando. A equipe de desenvolvedores de *games* não tem essa praticidade, sendo necessário encontrar uma linguagem que alcance a todos os membros para que eles possam trabalhar em conjunto e confiar na capacidade de todos para executarem suas tarefas para que, no final, possam unir todos os elementos em um único produto.

Para alcançar um melhor entendimento entre os membros da equipe de desenvolvimento de jogos digitais, primeiramente deve-se possuir uma forma bem estabelecida de documentação e conhecer a função e importância dos requisitos que nela se encontram. Portanto estes temas serão abordados no próximo capítulo.



## Capítulo 2

A documentação e seu conteúdo é um ponto importante da produção de um *software*, porém as peculiaridades dos jogos digitais vistas no capítulo anterior, como a grande quantidade de requisitos não-funcionais e multidisciplinaridade da equipe de produção, não estão previstas pelos modelos de documentação tradicionais, portanto este capítulo apresentará o GDD (*Game Design Document* ou Documento de *Game Design*) que foi concebido especificamente para *games* e tratará sobre as particularidades da elicitação do requisitos nesta área.

### Documento de *Game Design*

Entre as peculiaridades dos *games* que possam justificar a adoção de um novo formato de documentação podemos destacar a existência das regras que compõe a parte jogo do *videogame* que foram mencionadas no capítulo 1, isso se dá pois enquanto um *software* tradicional e seus requisitos normalmente são projetados com o intuito de facilitar uma ou mais tarefas do usuário, para os jogos digitais as regras, como foi apontado por Lemes (2009) e Lemes (2015), existem para desafiar o jogador. Outra justificativa é de que não existe uma formalização da documentação dos requisitos não-funcionais que são tão importantes para a concepção do *game*, já que normalmente eles não possuem uma função tão vital para a concepção do software tradicional.

Como apontado por L. Souza et al. (2009), para documentar essas peculiaridades do jogo digital e permitir a comunicação e entendimento de uma equipe que, como foi exposta no primeiro capítulo, pode conter membros de diversas áreas do conhecimento e diversas experiências, surgiu o Documento de *Game Design*, um modelo de documentação concebido para atender às necessidades da concepção e desenvolvimento dos *games*, entretanto, para atender à grande variedade de estilos de jogos, este documento acaba por ser extremamente abrangente e não possui um padrão fixo, podendo possuir sessões para mecânica, roteiro, arte, fases, requisitos emocionais, como o Flow, citados no capítulo 1, e até mesmo códigos de trapaça. Sessões estas que, durante a produção, podem ser preenchidas ou deixadas de lado conforme a necessidade de cada *game* a ser documentado.

Entretanto, L. J. Souza, Mittelbach, and Neves (2010) questiona se o formato textual proposto pelo GDD é o mais indicado para o entendimento da informação por uma equipe tão multidisciplinar como a dos desenvolvedores de jogos, testando a viabilidade de se especificar com imagens e vídeos. Sua pesquisa apontou para a necessidade de mesclar os formatos para dar maior clareza aos conceitos registrados pelo documento.

Lemes (2015) ainda sugere que junto ao documento em texto, que ele chama de Documento de Projeto de Jogo [DPJ] se codifique um executável, que ele chama de Documento Executável de Jogo [DEJ], para que se possa validar as idéas escritas no DPJ, auxilie na decisão de fazer corte de funções e ideias da concepção do *game* e ajude a gerar novas ideias, uma vez que ele torna real as abstrações do documento escrito.

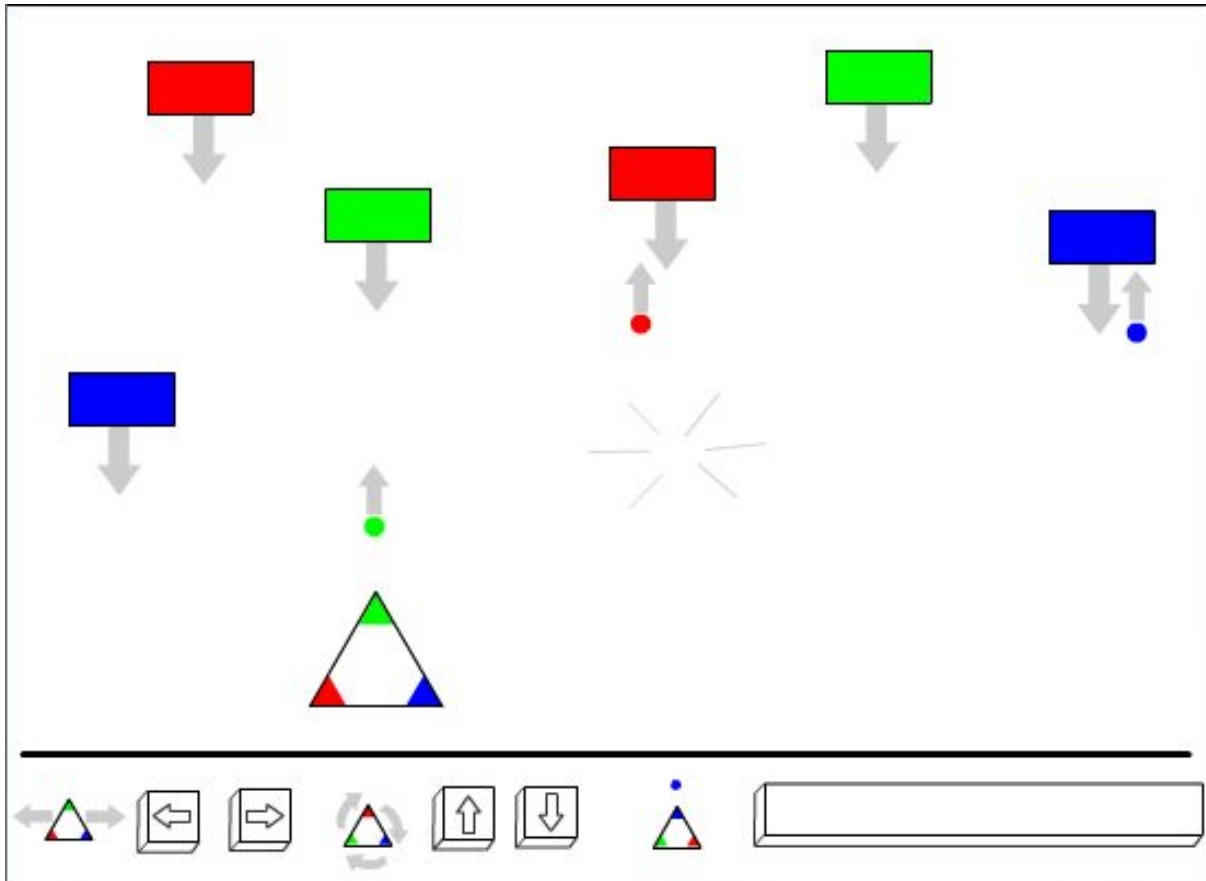


Figura 2 – Exemplo de especificação visual. L. J. Souza, Mittelbach, and Neves (2010)

## Registro dos Requisitos

Para um *software* tradicional os requisitos são primariamente as funcionalidades do produto a ser desenvolvido, podendo possuir de forma secundária requisitos não-funcionais como “fácil de usar” ou “bem colorido”. Estes geralmente não constituem uma parte fundamental das características do software, mas sim um diferencial. Quem define estes requisitos geralmente é um cliente que está contratando a equipe de desenvolvimento ou um representante deste, e eles são obtidos pela equipe através de entrevistas, documentos descritivos, contratos, acordos, entre outros, ou seja, a equipe tem a quem recorrer para elicitar os requisitos e tomar decisões de projeto.

Já para os *videogames*, além das funcionalidades do *software*, como dito anteriormente, temos as mecânicas do jogo para definir e requisitos não-funcionais de grande

importância para o sucesso do *game* como produto comercial. Um produto de *game* tem de ter uma identidade visual bem definida, divertir e entreter o jogador e até de ensinar e treinar, no caso de jogos educativos como Mercadinho do Seu Biu<sup>4</sup> da Manifesto Games, ou passar uma mensagem, no caso de *serious games* como Darfur is Dying<sup>5</sup> que Fullerton (2008) cita em seu trabalho, que são boas alternativas de mercado, principalmente para desenvolvedores independentes.

Para as equipes de desenvolvimento de *games* o cliente é geralmente o jogador, o público-alvo, então, como apontado por Alves, Ramalho, and Damasceno (2007)., seu desenvolvimento deve ser voltado para o mercado consumidor. Graças a isto os desenvolvedores, não têm acesso ao cliente, não têm de quem extrair os requisitos, tornando a elicitação um processo altamente criativo. Desde o momento da concepção, decisões de projeto, funcionalidades, regras, restrições, tudo está nas mãos da equipe e suas decisões impactarão no sucesso e retorno financeiro que o game proporcionará, por esta razão pesquisas de mercado, decisões de marketing e o apelo dos requisitos não-funcionais são tão importantes.

Para tomar decisões de documentação e extração de requisitos experiência é um fator que facilita o processo e aumenta as chances de sucesso do produto. Portanto no próximo capítulo serão apresentadas algumas práticas dos desenvolvedores brasileiros que já possuem experiência no mercado de desenvolvimento de *games*.

---

<sup>4</sup> Jogo sobre empreendedorismo, vencedor do Desafio Universtiário Empreendedor de 2014 do Sebrae e publicado pelo mesmo no site <http://www.desafio.sebrae.com.br> (acessado 10/11/2015)

<sup>5</sup> Jogo sobre a crise e terrorismo em Darfur no Sudão, pode ser jogado em <http://www.darfurisdying.com/index.html> (acessado 12/01/2016)



## Capítulo 3

Neste capítulo apresenta-se as ideias coletadas das entrevistas com os estúdios Aiyra, Manifesto Games e Joymasher. Apresenta-se também uma reflexão e comparação dos principais métodos de documentação e de trabalho de desenvolvimento de *software* realizado pelos estúdios. A transcrição das entrevistas estão no capítulo Anexos.

### A documentação

Todos os estúdios questionados mantêm alguma forma de documentação do jogo e salientam a importância de mantê-la para manter a ordem no projeto e para manter uma visão comum do jogo por toda a equipe. Adrian Laubisch do Estúdio Aiyra cita que o projeto pode facilmente virar um caos se não estiver documentado.

No entanto, existem grandes divergências quanto à forma escolhida para realizar a documentação. Por exemplo, os membros da JoyMasher escrevem textos, sejam físicos ou digitais com rabiscos em uma folha A3. A Manifesto Games usa quadros de ideias, fluxogramas e apresentações em PowerPoint. O estúdio Aiyra usa Wikis com textos e fotografias que são um misto de formatos como apontado por L. J. Souza, Mittelbach, and Neves (2010) conforme abordado no capítulo 2.

A documentação depende da necessidade e do escopo dos projetos e da forma que a equipe trabalha.

Por outro lado, é importante citar que todos os estúdios acreditam em plataformas colaborativas<sup>6</sup>, e toda a equipe pode modificar o conteúdo da documentação seja qual for o formato.

Adrian, do estúdio Aiyra, ressaltou que devido ao fato dos estúdios brasileiros trabalharem com projetos de escopo reduzido e desenvolvimento com prazos apertados, o formato original do GDD, abordado no segundo capítulo, foi deixado de lado em prol de plataformas mais dinâmicas como Wikis.

Afinal, como apontando não apenas por Adrian, mas por L. J. Souza, Mittelbach, and Neves (2010) e Salazar et al. (2012) em seus trabalhos, produzir um GDD completo costuma levar muito tempo e gerar um texto muito grande, difícil de ser lido e de ter os requisitos rastreados.

---

<sup>6</sup> Ferramentas através das quais é possível produzir um documento em que várias pessoas podem contribuir e editar (colaborar com) o conteúdo



Figura 3 – Imagem do jogo Oniken. Dias, Paiva, and Weiller (2012)

## O método de trabalho

Quanto ao método de trabalho todos os estúdios de *games* questionados pelo autor afirmaram usar adaptações de metodologias ágeis, principalmente Scrum<sup>7</sup>, uma escolha feita levando em conta a proposta de curtos ciclos iterativos, repetindo as várias etapas do processo de desenvolvimento, desde a concepção até o teste, para cada porção desenvolvida em cada iteração. Vicente Filho, da Manifesto Games, ressalta que esta característica vai ao encontro dos interesses do desenvolvimento de *game*, que está constantemente necessitando de ajustes e testes de qualidade, que, de acordo com esta metodologia, vão acontecer a cada iteração.

No entanto a metodologia como foi originalmente proposta apresenta algumas dificuldades para os desenvolvedores que acontecem, grande parte, pelo fato de que ela foi pensada para o desenvolvimento de um *software* tradicional, o maior exemplo disto são as *User Stories*, que se tornam difíceis de se escrever quando se leva em consideração as mecânicas e os requisitos não-funcionais dos jogos, que, como apontado no primeiro capítulo, são elementos essenciais para a concepção de um *videogame*, e mesmo os requisitos funcionais

<sup>7</sup> Metodologia ágil para desenvolvimento de software, pode ser lido mais sobre a metodologia em <http://www.desenvolvimentoagil.com.br/scrum/> (acessado em 11/01/2016)

sofrem com o fato de terem de ser decididos unicamente pela equipe de desenvolvimento. Miller (2008) e Keith (2008) abordam o uso do Scrum no desenvolvimento de *games* assim como seus pontos fortes e pontos fracos.

Como visto pelos relatos dos estúdios apontados neste capítulo e que podem ser lidos nos Anexos, é importante manter o hábito de documentar o desenvolvimento do projeto de *game*, em prol da comunicação da equipe (como apontado nos capítulos 1 e 2) e da organização do trabalho. Manter um método de trabalho ágil e iterativo, sendo Scrum o mais praticado, é importante para facilitar o desenvolvimento, uma vez que suas práticas convergem com as necessidades do processo de produção de um *videogame*. No entanto, vale ressaltar que cada equipe deve procurar os formatos de documentação e métodos de trabalho que melhor se adequem às necessidades de seus membros e projetos e nunca se esquecer de que eles são elementos que contribuem para o sucesso comercial.



# Conclusão

## Considerações finais

Com base nos conhecimentos expostos no texto acerca de *games*, seus requisitos, suas documentações e o relato de desenvolvedores atuando no mercado brasileiro, podemos tirar algumas conclusões quanto à concepção e desenvolvimento de um jogo digital.

Ao criar um jogo, o mercado consumidor deve ser seu foco, a partir dele o desenvolvedor deve pensar quais as características e restrições que o produto deve ter, como ele deve progredir, e como ele deveria afetar o jogador emocionalmente. Para chegar nessas conclusões, a equipe de desenvolvimento deve ser multidisciplinar e usar de empatia, se colocar no lugar do seu futuro usuário e imaginar como cada requisito, funcional ou não, influenciará sua experiência, pensando em apelos visual, complexidade das mecânicas e até mesmo que cores seriam mais adequadas para atingir o público-alvo.

Documentar estas decisões é importante, pois a documentação mantém a ordem e o entendimento da equipe acerca do *software* a ser desenvolvido. Embora existam alguns modelos para documentação de *softwares* tradicionais e até mesmo modelos específicos para jogos, como o GDD, os desenvolvedores não precisam se apegar a um *template*, podem fazer aquilo que melhor convir à equipe, desde que possua uma linguagem que atinja a todos os membros, principalmente em uma equipe tão multidisciplinar como a de desenvolvedores de jogos, usar imagens, mapas mentais, entre outros recursos, podem ser mais eficientes que um texto formal, assim como combinar estes recursos pode ser uma boa prática.

Trabalhar de forma iterativa e incremental, sempre atualizando e mantendo uma relação de complementaridade entre documentação e código (assim como nos documentos DPJ e DEJ citados por Lemes (2015)), testando e modificando o que for necessário, mesmo que seja um conceito já estabelecido, é a forma mais indicada de se trabalhar em um projeto que está tão vivo e sujeito a modificações e prazos apertados como um *game*.

## Trabalhos futuros

Persiste a carência de uma forma melhor definida de se especificar requisitos não funcionais, para que seja possível padronizar este processo no futuro da documentação dos jogos digitais, assim como ainda se faz necessária uma metodologia de desenvolvimento focada em *games*, para que as futuras equipes de desenvolvedores não precisem perder tempo de produção com a adaptação de uma metodologia que foi concebida para o desenvolvimento

de um *software* tradicional e que não funciona tão bem com o desenvolvimento de um jogo.

# Anexos

## Entrevista

Foi feita uma breve pesquisa através de troca de e-mails com membros de estúdios de games brasileiros. Responderam às perguntas Adrian Laubisch do Estúdio Aiyra, Vicente Filho da Manifesto Games e Thais Weiller da JoyMasher e as perguntas e suas respostas estão transcritas a seguir.

### **Vocês documentam o processo de criação dos seus jogos? Como?**

Aiyra - Sim, sem dúvida. Acreditamos que a agilidade e colaboração nesse ponto são muito importantes. Por isso a nossa documentação geralmente é um misto de brainstorms com quadros brancos e fotografias com formalizações das documentações em Wikis que podem ser editadas e consultadas por todos periodicamente, seguindo regras.

Manifesto - Sim. O processo de criação de jogos é documentado através de fluxograma. E cada um dos jogos possui uma game concept document (apresentação em PPT) para descrever detalhes gerais como mecânica básica, estilo artístico e referências.

JoyMasher - Depende do jogo. Alguns jogos são documentados no google drive, com um documento de texto formal em que todos editam. Outros são só uma folha A3 com um monte de coisa rabiscada. Depende do jogo.

### **Que decisões foram importantes para a concepção e desenvolvimento dos jogos? Em que momentos do processo vocês tomaram essas decisões?**

Aiyra - Se a pergunta está relacionada à decisão para a documentação, a decisão foi tomada logo no início, quando notou-se o caos que é gerado quando um projeto de game (por ser tão vivo e tão multidisciplinar) fica sem controle de documentação.

JoyMasher - A maior parte das decisões importantes ficam para o começo, como o tipo do jogo, o tema, os elementos de gameplay, etc. Mas algumas decisões muito difíceis e importantes para a entrega acabam ficando para o fim, como cortar partes e mecânicas do jogo.

### **Vocês conhecem e/ou utilizam algum processo de desenvolvimento de software (ou parte dele) no desenvolvimento dos seus jogos?**

Aiyra - Conhecemos diversos. Utilizamos no momento uma versão adaptada do SCRUM, baseada em um processo desenvolvido pelo nosso CEO.

Manifesto - Nós adotamos as metodologias ágeis, especialmente SCRUM, para gestão dos projetos.

JoyMasher - Usamos um pouco de AGILE e um pouco de Scrum, mas sempre com adaptações.

**O que vocês acreditam que diferencia o desenvolvimento de um jogo do desenvolvimento de um software comum?**

Aiyra - A multidisciplinaridade das equipes. Projetos de games tem equipes de arte, som, código, roteiro, game design, negócios e todas elas são igualmente importantes para o resultado final. Coordenar a comunicação entre todas estas é o verdadeiro desafio.

Manifesto - A criação de jogos envolve inúmeros elementos não-tangíveis que são essenciais para o sucesso do projeto. Estética, usabilidade, diversão, engajamento são alguns desses elementos.

Essa característica torna a produção de jogos menos linear, menos *waterfall*. A simples produção dos requisitos listados em um documento não produzem um jogo divertido. É essencial, portanto, testar e avaliar constantemente o jogo para ajustar esses elementos pouco tangíveis. E é justamente aí que os processos ágeis auxiliam na produção por incentivarem a iteração constante sobre o projeto.

JoyMasher - Tudo, jogo e software só tem em comum sua composição em programação. Um software pode ser planejado desde o começo e pode até ser feito na metodologia *waterfall*, um jogo é algo tão elástico e variável que até Scrum pode não se adaptar tão bem ao processo criativo. É quase como se software fosse música erudita e jogos fossem jazz.

# Referências

- Alves, Carina, Geber Ramalho, and Alexandre Damasceno. 2007. “Challenges in Requirements Engineering for Mobile Games Development: The Meantime Case Study.” In *Requirements Engineering Conference, 2007. RE’07. 15th IEEE International*, 275–80. IEEE.
- Callele, David, Eric Neufeld, and Kevin Schneider. 2005. “Requirements Engineering and the Creative Process in the Video Game Industry.” In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, 240–50. IEEE.
- . 2006. “Emotional Requirements in Video Games.” In *Requirements Engineering, 14th IEEE International Conference*, 299–302. IEEE.
- Csikszentmihalyi, Mihaly. 1991. *Flow: The Psychology of Optimal Experience*. Vol. 41. HarperPerennial New York.
- Dias, Danilo, Pedro Paiva, and Thais Weiller. 2012. “Oniken.” Joysmasher; digital game. june. <http://joymasher.com/oniken/>.
- Fullerton, Tracy. 2008. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Second Edition. Elsevier Inc.
- Hassenzahl, Marc, Andreas Beu, and Michael Burmester. 2001. “Engineering Joy.” *Ieee Software*, no. 1. IEEE: 70–76.
- Keith, Clinton. 2008. “Feedback: ‘Top 10 Pitfalls Using Scrum Methodology for Video Game Development’” website. september. <http://blog.agilegamedevelopment.com/2008/07/feedback-top-10-pitfalls-using-scrum.html>.
- Lemes, David de Oliveira. 2009. “Games Independentes - Fundamentos Metodológicos Para Criação, Planejamento E Desenvolvimento de Jogos Digitais.” *Programa de Pós-Graduação Em Tecnologias Da Inteligência E Design Digital. Dissertação de Mestrado. Orientação: Luís Carlos Petry. Pontifícia Universidade Católica de São Paulo-PUC: SP*. Master’s thesis.
- . 2015. “A Técnica de Programação Exploratória (PXP) : Projetos de Criação E Desenvolvimento de Jogos Digitais.” PhD thesis, PUCSP.
- Miller, Paul. 2008. “Top 10 Pitfalls Using Scrum Methodology for Video Game Development.” Gamasutra; webpage. july. [http://www.gamasutra.com/view/feature/132121/top\\_10\\_pitfalls\\_using\\_scrum\\_.php](http://www.gamasutra.com/view/feature/132121/top_10_pitfalls_using_scrum_.php).
- Salazar, Mario Gonzalez, Hugo Mitre, Cuauhtémoc Lemus Olalde, José Luis González Sánchez, and others. 2012. “Proposal of Game Design Document from Software

Engineering Requirements Perspective.” In *Computer Games (CGAMES), 2012 17th International Conference on*, 81–85. IEEE.

Souza, LJ, CS Albuquerque, SM Fontes, MM Câmara, SG Countinho, and AM Neves. 2009. “Análise de Documento de Game Design: Interpretação E Resultados Gerados.” In *VIII Brazilian Symposium on Games and Digital Entertainment*.

Souza, Luiz J, Artur F Mittelbach, and André M Neves. 2010. “Estudo de Formatos Alternativos Para Documentação de Game Design.” In *IX Brazilian Symposium on Games and Digital Entertainment*.

STATISTA. 2015. “Video Games Revenue Worldwide from 2012 to 2015, by Source (in Billion U.S. Dollars).” website. <http://www.statista.com/statistics/278181/video-games-revenue-worldwide-from-2012-to-2015-by-source/>.