

Requisitos não funcionais para Jogos Digitais

David de Oliveira Lemes*

Terceiro autor Thiago Cazelatto Breves

Mario Madureira Fontes

Pontifícia Universidade Católica de São Paulo, Departamento de Computação, Brasil

RESUMO

Este trabalho apresenta uma comparação de 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*

1 INTRODUÇÃO

A indústria de *videogames* (Utilizado como sinônimo para a expressão "jogos digitais") gira um faturamento de cerca de 100 bilhões de dólares ao ano [15] e 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 vários escritores da área [1, 2, 3, 13, 14], mas 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.

Diversas publicações [1, 2, 3, 10] 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.

2 CONCEITUAÇÕES

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.

*e-mail: dolemes@pucsp.br

2.1 Definindo um *videogame*

O *videogame* é definido por Callele et al. [3] 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 et al. [7], considerando elementos hedônicos como estilo, cores e música. Callele et al. [3] 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.

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 era passada 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 nos quais 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*¹ do que um *multiplayer*²).

A parte emocional é justamente aquilo que o *game designer* imagina que o jogador deve sentir durante seu jogo, Callele et al. [3] 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 [6] 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 [4] como um estado mental no qual o indivíduo está completamente envolvido

¹Jogo para apenas um jogador.

²Jogo para dois ou mais jogadores.

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*. Para manter esse equilíbrio e o jogador envolvido na atividade de jogar o *videogame*, deve-se considerar, por exemplo, que, conforme o jogador avança no jogo, ele irá se acostumando às mecânicas e melhorando suas habilidades, exigindo 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.

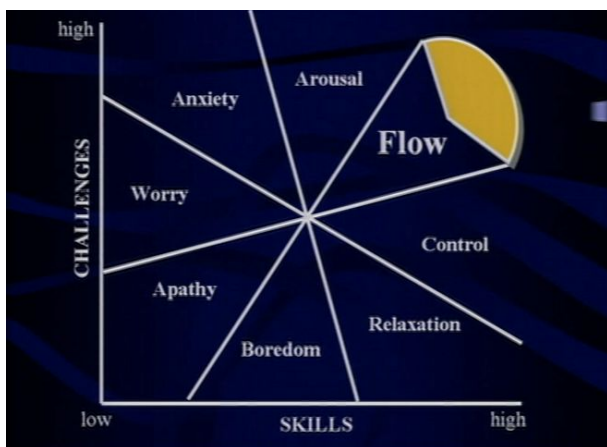


Figura 1: Gráfico de estados mentais. Csikszentmihalyi (1991) [4]

2.2 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 et al.[2] 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.

Uma equipe de desenvolvedores de *software* tradicional é geralmente composta inteira ou quase inteiramente por profissionais da área de Tecnologia da Informação. A vantagem dessa 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 na seção 3.

3 DOCUMENTANDO UM *game*

A documentação e seu conteúdo é um ponto importante da produção de um *software*, porém as peculiaridades dos jogos digitais abordadas na seção 2.2, como a grande quantidade de requisitos não-funcionais e multidisciplinaridade da equipe de produção, não estão previstas pelos modelos de documentação tradicionais. Na seção 3.1 será apresentado o GDD (*Game Design Document* ou Documento de *Game Design*) que foi concebido especificamente para *games* e serão tratadas as particularidades da elicitação dos requisitos no campo de desenvolvimento de *games*.

3.1 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 na seção 2.1, 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. Nos jogos digitais, as regras, como foi apontado por Lemes [9, 10], 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 Souza et al. [13], para documentar essas peculiaridades do jogo digital e permitir a comunicação e entendimento de uma equipe que, como foi exposta na seção 2.1, 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, esse documento acaba por ser extremamente abrangente e não possui um padrão fixo, abrangendo sessões sobre mecânica, roteiro, arte, fases, requisitos emocionais, como o Flow, citados na seção 2.1, e até mesmo códigos de trapaça. Tais sessões, durante a produção, podem ser preenchidas ou deixadas de lado conforme a necessidade de cada *game* a ser documentado.

Entretanto, Souza et al. [14] 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, discutindo a viabilidade de se fornecer especificações por meio de imagens e vídeos. Sua pesquisa apontou para a necessidade de mesclar os formatos para dar maior clareza aos conceitos registrados pelo documento.

Lemes [10] 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 idéas da concepção do *game* e ajude a gerar novas idéas, uma vez que ele torna real as abstrações do documento escrito.

3.2 Registro dos Requisitos

Para um *software* tradicional os requisitos são primariamente as funcionalidades do produto a ser desenvolvido e, de uma maneira secundária, requisitos não-funcionais como “fácil de usar” ou “bem colorido”. Esses geralmente não constituem uma parte fundamental das características do *software*, mas sim um diferencial. Quem define esses 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

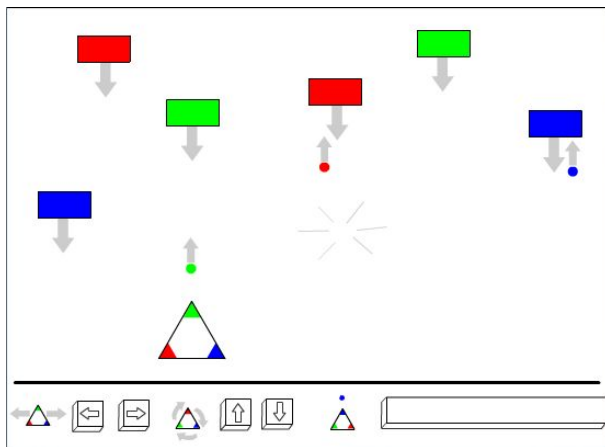


Figura 2: Exemplo de especificação visual. Souza et al. (2010) [14]

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 Bui*³ da *Manifesto Games*, ou ainda passar uma mensagem, no caso de *serious games* como *Darfur is Dying*⁴ que [6] cita em seu trabalho, como 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 [1], 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 na seção 4 serão apresentadas algumas práticas dos desenvolvedores brasileiros que já possuem experiência no mercado de desenvolvimento de *games*.

4 ESTUDOS DE CASO

Nesta seção apresentam-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 pode ser acessada no link: <http://bit.ly/2917KRj>.

4.1 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.

³Jogo sobre empreendedorismo, vencedor do Desafio Universitário Empreendedor de 2014 do Sebrae e publicado pelo mesmo no site <http://www.desafio.sebrae.com.br> (acessado 10/11/2015)

⁴Jogo sobre a crise e terrorismo em Darfur no Sudão, pode ser jogado em <http://www.darfurisdying.com/index.html> (acessado 12/01/2016)

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 Souza et al. [14] conforme abordado na seção 3.

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 colaborativas⁵, 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 na seção 3, foi deixado de lado em prol de plataformas mais dinâmicas como Wikis.

Final, como apontando não apenas por Adrian, mas por Souza et al. [14] e Salazar et al. [12] 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.



Figura 3: Imagem do jogo Oniken. Dias et al. (2012) [5]

4.2 O método de trabalho

Quanto ao método de trabalho todos os estúdios de *games* questionados pelo autor afirmaram usar adaptações de metodologias ágeis, principalmente SCRUM⁶, 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*, resalta 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 e são devidas, grande parte, ao fato de que ela foi pensada para o desenvolvimento de um *software* tradicional. O maior exemplo disso são as *User Stories*, que se tornam difíceis de se escrever quando se leva em consideração as mecânicas e os requisitos não-funcionais dos

⁵Ferramentas através das quais é possível produzir um documento em que várias pessoas podem contribuir e editar (colaborar com) o conteúdo

⁶Metodologia ágil para desenvolvimento de *software*, pode ser lido mais sobre a metodologia em <http://www.desenvolvimentoagil.com.br/scrum/> (acessado em 11/01/2016)

jogos, que, como apontado na seção 2, são elementos essenciais para a concepção de uma *videogame*, e mesmo os requisitos funcionais sofrem com o fato de terem de ser decididos unicamente pela equipe de desenvolvimento. Miller [11] e Keith [8] abordam o uso do SCRUM no desenvolvimento de *games* assim como seus pontos fortes e fracos.

Como pode ser observado pelos relatos dos estúdios apontados nesta seção é importante manter o hábito de documentar o desenvolvimento do projeto de *game*, em prol da comunicação da equipe (como apontado nos seções 2 e 3) 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 as necessidades de seus membros e projetos e nunca se esquecer de que eles são elementos que contribuem para o sucesso comercial.

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 a essas 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 as 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 convier 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 esses 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 [10]), 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.

REFERÊNCIAS

- [1] C. Alves, G. Ramalho, and A. Damasceno. Challenges in requirements engineering for mobile games development: The meantime case study. In *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*, pages 275–280. IEEE, 2007.
- [2] D. Callele, E. Neufeld, and K. Schneider. Requirements engineering and the creative process in the video game industry. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 240–250. IEEE, 2005.
- [3] D. Callele, E. Neufeld, and K. Schneider. Emotional requirements in video games. In *Requirements Engineering, 14th IEEE International Conference*, pages 299–302. IEEE, 2006.
- [4] M. Csikszentmihalyi. *Flow: The psychology of optimal experience*, volume 41. HarperPerennial New York, 1991.
- [5] D. Dias, P. Paiva, and T. Weiller. Oniken. digital game, june 2012. Acessado em 25 de dezembro 2015.
- [6] T. Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Elsevier Inc., second edition, 2008.
- [7] M. Hassenzahl, A. Beu, and M. Burmester. Engineering joy. *Ieee Software*, (1):70–76, 2001.
- [8] C. Keith. Feedback: “top 10 pitfalls using scrum methodology for video game development”. website, september 2008. Acessado em 25 de dezembro 2015.
- [9] D. d. O. Lemes. Games independentes - fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais. Master’s thesis, 2009.
- [10] D. d. O. Lemes. *A técnica de programação exploratória (PXP) : projetos de criação e desenvolvimento de jogos digitais*. PhD thesis, PUCSP, abril 2015.
- [11] P. Miller. Top 10 pitfalls using scrum methodology for video game development. webpage, july 2008. Acessado em 25 de dezembro 2015.
- [12] M. G. Salazar, H. Mitre, C. L. Olalde, J. L. G. Sánchez, et al. Proposal of game design document from software engineering requirements perspective. In *Computer Games (CGAMES), 2012 17th International Conference on*, pages 81–85. IEEE, 2012.
- [13] L. Souza, C. Albuquerque, S. Fontes, M. Câmara, S. Coutinho, and A. Neves. Análise de documento de game design: Interpretação e resultados gerados. In *VIII Brazilian Symposium on Games and Digital Entertainment*, 2009.
- [14] L. J. Souza, A. F. Mittelbach, and A. M. Neves. Estudo de formatos alternativos para documentação de game design. In *IX Brazilian Symposium on Games and Digital Entertainment*, 2010.
- [15] STATISTA. Video games revenue worldwide from 2012 to 2015, by source (in billion u.s. dollars). website, 2015. Acessado em 25 de dezembro 2015.