



PUC-SP

Pontifícia Universidade Católica de São Paulo

Eduardo Monclús Romanek

**Proposta de uma matriz de avaliação para *Quality Assurance* na
análise do processo de produção em jogos digitais**

**Mestrado em Desenvolvimento de
Jogos Digitais.**

São Paulo

2020



Pontifícia Universidade Católica de São Paulo

Eduardo Monclús Romanek

Proposta de uma matriz de avaliação para *Quality Assurance* na análise do processo de produção em jogos digitais

Relatório apresentado à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de Mestre em Desenvolvimento de Jogos Digitais, redigida sob a orientação do professor Dr. David de Oliveira Lemes.

São Paulo

2020

Romanek, Eduardo Monclús.

Proposta de uma matriz de avaliação para *Quality Assurance* na análise do processo de produção em jogos digitais

Registro: 2020

Orientador: David de Oliveira Lemes.

Dissertação de Conclusão de Mestrado – Pontifícia Universidade Católica de São Paulo

ROMANEK, Eduardo Monclús. **PROPOSTA DE UMA MATRIZ DE AVALIAÇÃO PARA *QUALITY ASSURANCE* NA ANÁLISE DO PROCESSO DE PRODUÇÃO EM JOGOS DIGITAIS.** Dissertação. Pontifícia Universidade Católica de São Paulo. Mestrado em Desenvolvimento de Jogos Digitais, redigida sob a orientação do professor Dr. David de Oliveira Lemes .

Banca Examinadora

Professor Dr. David de Oliveira Lemes
PUC-SP - Orientador

Professor Dr. Alexandre Santaella Braga
PUC-SP- Membro da Banca Examinadora - Interno

Professor Dr. Hermes Renato Hildebrand
PUC-SP – Membro da Banca Examinadora - Interno

Professor Dr. xxxxxx
Membro da Banca Examinadora - Externo

Professor Dr.
Membro da Banca Examinadora - Externo

Agradecimentos

Em primeiro lugar, agradeço a xxxxxxxxx.

Aos professores xxxxxxxxxx xxxxxxxxxx xxxxxxxxx

Ao meu orientador, xxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxx

Em especial, aos xxxxxxxxxx xxxxxxxxxx xxxxxxxxx

“A delayed game is eventually good, but a rushed game is forever bad.”

Shigeru Miyamoto

Resumo

O setor de QA no processo de produção de jogos tem um papel importante no que se refere a garantir que o jogo no mínimo seja funcional, com todas os componentes necessários para que o jogador tem a experiência do jogo. Deste forma pretende-se tecer a respeito deste setor e como o setor atua nessa questão.

Palavras-chave:. Garantia da Qualidade, QA, jogos digitais, produção de jogos digitais

Abstract

The QA sector in the game production process has an important role when it comes to ensuring that the game is at least functional, with all the necessary components so that the player has the experience of the game. This way, we intend to weave about this sector and how the sector works in this issue

Keywords: Quality Assurance, QA, game production

Índice de Figuras

FIGURA 1 – FATURAMENTO EM 2017 DAS DESENVOLVEDORAS DE JOGOS DIGITAIS..	ERRO! INDICADOR NÃO DEFINIDO.
FIGURA 2 – QUANTIDADE DE JOGOS PRODUZIDOS 2016/2017.....	7
FIGURA 3 – CLIMBING FIREMAN (1920)	27
FIGURA 4 – PINBALL HUMPTY DUMPTY GOTTLIEB,PUBLICAÇÃO DA INOVAÇÃO NA REVISTA BILLBOARD - 1947.....	29
FIGURA 5 - JOGO DE FUTEBOL	30
FIGURA 6 – PRODUÇÃO E CONSUMO DO JOGO	40
FIGURA 7 – DESIGNER E JOGADOR PERSPECTIVAS DIFERENTES	42
FIGURA 8 – SETOR DE QA ESTRUTURA ORGANIZACIONAL DESENVOLVEDORES E PUBLISHER’S	52
FIGURA 9 – ATIVIDADES DE UM GAME TESTE	56
FIGURA 10 – MODELO DE MATRIZ PUGH.....	97
FIGURA 11 – MODELO DA MATRIZ QFD.....	99
FIGURA 12 – PERSPECTIVA DO QA EM RELAÇÃO AO FRAMEWORK MDA	102
FIGURA 13 – GRÁFICO DMC X PGLD.....	106

Índice de Quadros

QUADRO 1 - SÍNTESE DAS PARTICULARIDADES DAS CIÊNCIAS NATURAIS SOCIAIS E DESIGN SCIENCE	12
QUADRO 2 - TIPOS DE ARTEFATOS	14
QUADRO 3 – RESUMO DOS CONCEITOS DA DESIGN SCIENCE	15
QUADRO 4 – FRAMEWORK DA DSR	17
QUADRO 5– ETAPAS NA CONDUÇÃO DA PESQUISA DSR	18
QUADRO 6 – MÉTODO DSR	21

Índice de Tabelas

TABELA 1- RESULTADO DE BUSCA EM LÍNGUA PORTUGUESA	23
TABELA 2 – RESULTADO DE BUSCA EM LÍNGUA INGLESA	23
TABELA 3- RESUMO DOS BUGS EM NÍVEIS DE GRAVIDADE	62
TABELA 4 – MATRIZ DE AVALIAÇÃO (DMC).....	103
TABELA 5 – VALORES DOS PESOS E NOTA A SEREM ATRIBUÍDOS NA MATRIZ DE AVALIAÇÃO	104
TABELA 6 – MATRIZ DE AVALIAÇÃO PRINCÍPIOS DO BOM LEVEL DESIGN (PGLD).....	105

Lista de Abreviaturas

.

SUMÁRIO

1. INTRODUÇÃO	4
1.2 PRINCIPAL QUESTÃO DA PESQUISA	4
1.3 HIPÓTESE	4
1.4 LOCALIZAÇÃO DA REVISÃO DE ESTUDOS ANTERIORES NO TEMPO E NO ESPAÇO	4
1.5 OBJETIVOS	4
1.5.1 <i>Objetivo geral</i>	5
1.5.2 <i>Objetivos específicos</i>	5
1.6 JUSTIFICATIVAS E RELEVÂNCIA DA PESQUISA	5
1.6.1 <i>Justificativa e relevância científica e comercial</i>	5
1.6.2 <i>Justificativa e relevância pessoal, profissional e acadêmica</i>	8
1.7 ORDENAÇÃO DOS CAPÍTULOS DE DISSERTAÇÃO	9
2. METODOLOGIA DA PESQUISA CIENTÍFICA	10
2.1 DEFINIÇÃO <i>DESIGN SCIENCE</i>	11
2.2 CONCEITOS DA <i>DESIGN SCIENCE</i>	13
2.3. <i>DESIGN SCIENCE RESEARCH</i>	15
2.3.1 MODELOS PARA CONDUÇÃO DA <i>DESIGN SCIENCE RESEARCH</i>	18
2.4 EMPREGO DO MÉTODO <i>DESIGN SCIENCE RESEARCH</i>	22
2.4.1 EMPREGO DO MÉTODO <i>DESIGN SCIENCE RESEARCH</i>	22
3. REVISÃO ESTUDOS ANTERIORES	22
3.1 PRODUÇÃO NACIONAL	22
3.2 PRODUÇÃO INTERNACIONAL	23
4. ABORDAGEM TEÓRICA	25
4.1 JOGO, O COMEÇO DE TUDO UM BREVE CONTEXTO HISTÓRICO	25
4.2 CRASH 1983 – CRISE NO MERCADO NORTE-AMERICANO DE JOGOS DIGITAIS E A QUALIDADE	30
4.3 NOVOS RUMOS SELO DE QUALIDADE NA INDÚSTRIA DO VÍDEO GAME	31
4.4 JOGOS ARCADE PADRÃO DE REFERÊNCIA DOS JOGOS	31
4.5 FUNDAMENTOS DO GAME DESIGN	31
4.5.1 <i>Qual é a definição de jogo</i>	31
4.5.2 <i>Sobre Tecnologia Mecânica, Dinâmica, Estética e Narrativa de jogos</i>	40

4.5.2.1 Framework MDA (Mechanics, Dynamics and Aesthetics)	40
4.5.2.2 Tétrade elementar	45
4.5.3 Interface e Experiência do Usuário	47
4.5.4 Teoria do Flow	47
4.6 FUNDAMENTOS PARA DESENVOLVIMENTO E PRODUÇÃO DOS JOGOS DIGITAIS	47
4.6.1 Projeto de um jogo digital	47
4.6.2 Etapas de desenvolvimento do jogo digital	48
4.6.3 Perfil do Jogador	48
4.6.4 Público-alvo	48
4.6.5 Plataformas de desenvolvimento do jogo digital	48
4.6.6 Métodos de prototipagem do jogo digital	49
4.6.7 Engenharia de Software e Hardware	49
5. QUALITY ASSURANCE (QA) A PLAYABILITY TESTING	50
5.1 DEFINIÇÃO DE QUALITY ASSURANCE (QA) NOS JOGOS DIGITAIS E SUA ESTRUTURA	50
5.2 ESSE UNIVERSO DOS TESTES, ETAPAS DO PROCESSO	55
5.2.1 A definição de bug e suas categorias	61
5.3.2 Realização de testes no ciclo de produção do jogo, uma visão geral	70
5.2.2.1 Teste Pré-produção	70
5.2.2.2 Teste Kickoffs	73
5.2.2.3 Teste Alfa	73
5.2.2.4 Teste Beta	76
5.2.2.5 Teste Ouro	81
5.2.2.6 Teste Pós-produção	82
5.2.3 OUTROS MÉTODOS DE TESTE UTILIZADOS NO DESENVOLVIMENTO DE UM JOGO	82
5.2.3.1 Teste Caixa Preta – "Black Box"	82
5.2.3.2 Teste Caixa Branca – "White Box"	83
5.2.3.3 Teste Ad Hoc	83
5.2.3.4 Teste Automatizado	85
5.2.3.5 Teste de Funcionalidade	86
5.2.3.6 Teste de Fumaça – "Smoke Testing"	87
5.2.3.7 Teste de Regressão	87
5.2.3.8 Teste de Progressão	88
5.2.3.9 Teste de Compatibilidade	88
5.2.3.10 Teste de Conformidade	88
5.2.3.11 Teste de Localização	88
5.3 FERRAMENTAS USADAS PELA EQUIPE DE QA	89

5.4 PLAYABILITY TESTING	90
5.5 ÁGILE QA??.....	95
6. DESENVOLVIMENTO DA MATRIZ DE AVALIAÇÃO	95
6.1 PROCESSO DE GESTÃO E QUALIDADE.....	95
5.2 FERRAMENTAS DE GESTÃO E QUALIDADE.....	96
5.2.1 MATRIZ DE PUGH.....	96
5.2.2 MATRIZ QUALITY FUNCTION DEPLOYMENT (QFD)	98
5.3 PROPOSTA DE FERRAMENTA DE AVALIAÇÃO	99
6. DISCUSSÃO DOS RESULTADOS	106
7. CONSIDERAÇÕES FINAIS.....	106
REFERÊNCIAS	107
GLOSSÁRIO.....	111
ANEXOS	112
ANEXO – 01 GAME DESIGN DOCUMENT	113
ANEXO – 02 RELAÇÃO DE ASSETS	115
ANEXO – 03 MODELAGEM UML.....	117
ANEXO – 04 MAPA DE ORIENTAÇÃO.....	119
ANEXO – 05 MAPEAMENTO BIBLIOGRÁFICO	121
ANEXO – 06 INFOGRÁFICOS.....	123
ANEXO – 07 MAPAS MENTAIS	125
ÍNDICE DE AUTORES.....	126
ÍNDICE REMISSIVO.....	127

1. INTRODUÇÃO

1.2 Principal questão da pesquisa

O objeto de estudo toma como referência a seguinte questão: *“Como o auxílio de uma matriz de avaliação para tomada de decisão pode contribuir com o Game Designer e Quality Assurance no projeto de desenvolvimento de um jogo digital?”*

1.3 Hipótese

Nesta pesquisa considera-se a hipótese da utilização de uma **matriz de avaliação**, onde esta possa ser uma nova ferramenta como auxílio no processo de desenvolvimento de um jogo digital.

1.4 Localização da revisão de estudos anteriores no tempo e no espaço

A revisão de estudos anteriores e trabalhos relacionados teve foco delimitado no período entre 2010 a 2020, o embasamento literário disponível em livros, palestras, artigos científicos, teses e dissertações.

A referida literatura restringiu-se às publicações nas línguas inglesa e portuguesa, considerando pesquisas disponíveis nas seguintes bases online:

- a) Banco de Teses e Dissertações da Pontifícia Universidade Católica de São Paulo (PUC-SP)
- b) Banco de Teses e Dissertações da Capes
- c) SBGames – Sociedade Brasileira de Games
- d) SciELO - *Scientific Eletronic Library Online*
- e) BDTD – Biblioteca Digital Brasileira de Teses e Dissertações

1.5 Objetivos

1.5.1 Objetivo geral

A presente pesquisa tem por objetivo geral esquadrihar sobre jogos digitais, contemplando principalmente os conceitos relativos a *Game Design e Quality Assurance* (QA), bem como, as descobertas e desafios pertinentes ao tema. Outrossim, levando-se em conta as possíveis aplicações de uma **matriz de avaliação** dos elementos que compõem o jogo e inseri-los nessa conjectura, como auxílio à evolução do projeto em desenvolvimento.

1.5.2 Objetivos específicos

A partir da proposta supracitada, a pesquisa tem os seguintes objetivos específicos:

- a) Identificar os estudos acadêmicos relacionados ao *Game Design e Quality Assurance*, tendo por referência o período de 2010 a 2020, valendo-se de palavras-chaves como: *Game Design, produção de jogos digitais, desenvolvimento jogos digitais, garantia de qualidade, controle de qualidade*;
- b) Contextualizar sobre *Game Design e Quality Assurance*, os princípios do desenvolvimento e produção dos jogos digitais, com o intuito de compreender e obter um arcabouço para proposição de uma ferramenta;
- c) Propor uma matriz de avaliação com os conceitos de *Game Design*;

1.6 Justificativas e relevância da pesquisa

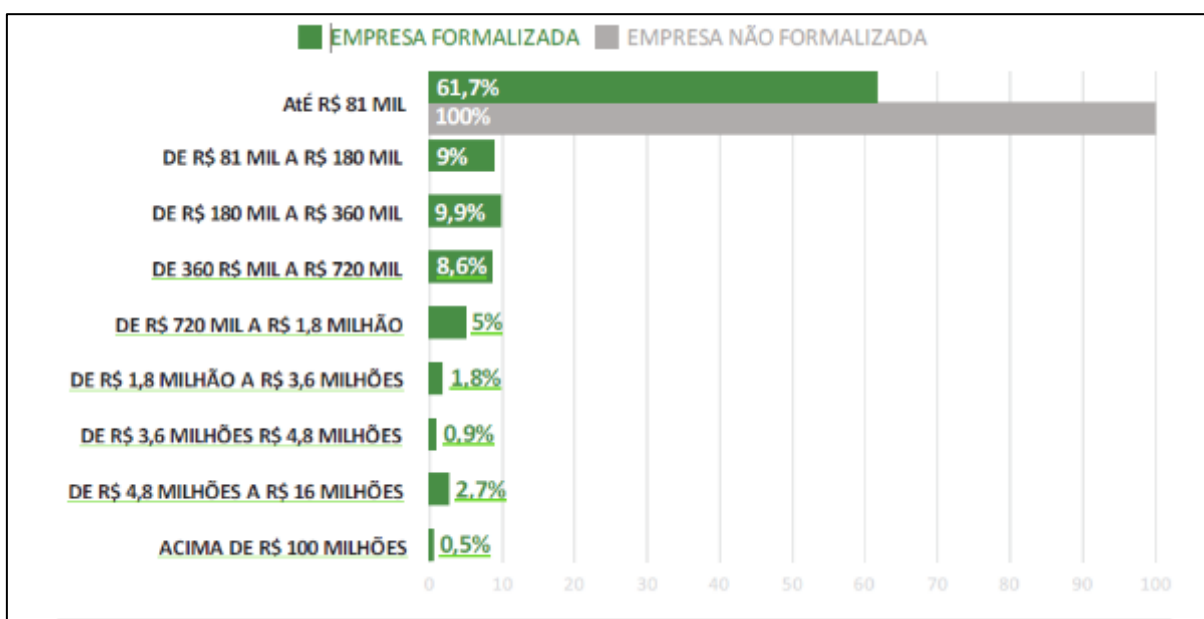
1.6.1 Justificativa e relevância científica e comercial

O cenário do desenvolvimento de jogos digitais vem se consolidando no Brasil, uma pesquisa realizada em 2018 pelo BNDES – II Censo da Indústria Brasileira de Jogos Digitais (II CIBJD) apontam um crescimento de 182% em relação ao primeiro

censo realizado em 2014 (FLEURY, SAKUDA e CORDEIRO, 2014), foram identificados 375 desenvolvedoras de jogos digitais até o início de 2018. Dessa amostra, 276 empresas possuem empresas abertas ou profissionais autônomos formalizados que emitem recibo de pagamento ou nota fiscal, e 99 destes foram categorizados como profissionais autônomos não formalizados não geram tributos para o Estado. (SAKUDA & FORTIM, 2018)

Vale destacar o levantamento do II CIBJD a respeito do faturamento dos desenvolvedoras de jogos em 2017 correspondente a trezentas e trinta e uma respostas contabilizadas, como segue:

Figura 1 – Faturamento em 2017 das desenvolvedoras de jogos digitais



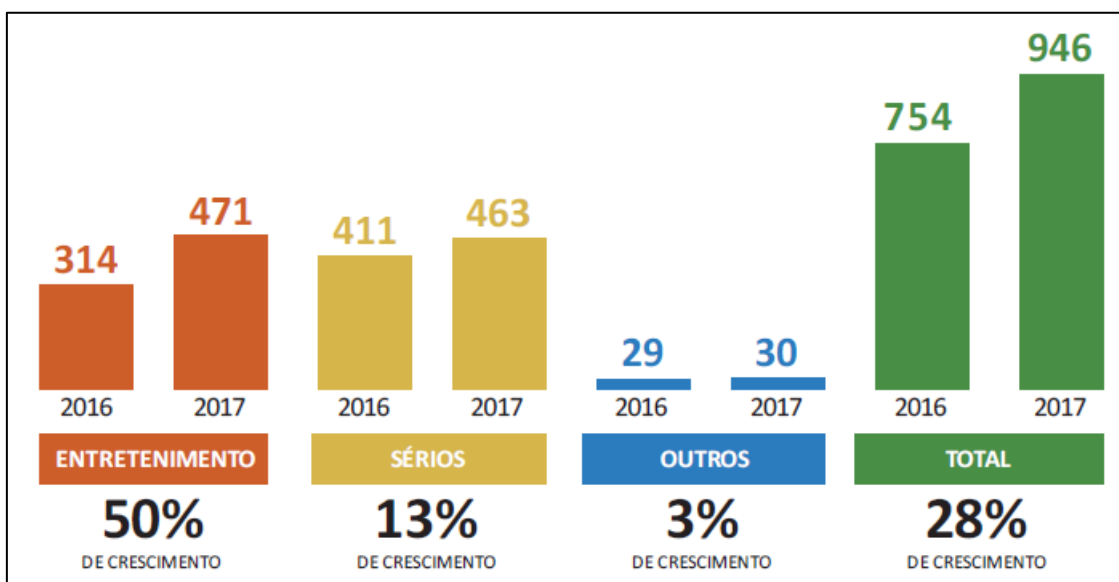
Fonte: (SAKUDA & FORTIM, 2018)

Os pesquisadores Sakuda & Fortim (2018), apontam que 165 das empresas, cerca de 49,8% a principal fonte de receita é oriunda no desenvolvimento de jogos digitais, a outra metade das empresas apesar de declararem com desenvolvedoras de jogos digitais, necessitam de outras atividades para constituir seu faturamento. Dentre essas outras atividades destacam-se o desenvolvimento de software e serviço de tecnologia de informação com 35 empresas (10,6%). Percebe-se pelo gráfico

apresentado que a principal faixa do faturamento anual tanto das empresas formalizadas (61,7%) como empresas não formalizadas (100%) é de até R\$ 81 mil.

Outro levantamento apresentado pelos pesquisadores, identifica a quantidade de jogos produzidos no biênio 2016/2017, sendo 1718 jogos desenvolvidos neste período pelas 227 empresas que responderam à pesquisa neste quesito. As empresas desenvolvedoras nacionais têm pautado o seu mercado de atuação para os jogos destinados para os segmentos de entretenimento com 785 jogos produzidos, jogos sérios (*serious game*) 874 jogos e com 59 para outros tipos de jogos.

Figura 1 – Quantidade de jogos produzidos 2016/2017



Fonte: (SAKUDA & FORTIM, 2018)

Outro aspecto proveniente da pesquisa, quando abordado sobre as atividades em parceria realizadas pelas desenvolvedoras com outros setores que atuam no mercado de jogos digitais, dentre as 242 respostas, os mais mencionados na lista são palestras e cursos com 54,5%, **play tests**¹ com 39,3% e as maratonas de

¹ *Play test* é o processo no qual a desenvolvedora testa um novo em busca de falhas antes de colocá-lo no mercado. Esse teste pode ser “aberto”, como por exemplo em eventos, “fechado” teste interno sem a participação do público e beta (teste final) quando o jogo está próximo de ser inserido no mercado.

desenvolvimento (Game Jams²) com 30,02%. A mesma pesquisa realizada para os profissionais autônomos, com 185 respostas, são os Game Jams com 45,4%, palestras e cursos (36,8%) e **play tests** (34,1%).

Assim, podemos notar a importância da realização de *play tests* tanto por parte das desenvolvedoras, como para os profissionais autônomos. Dessa forma, o teste do protótipo em locais ou eventos públicos, ajudam a refinar o andamento do projeto, além da troca de experiências com outros profissionais da área e aproximação com o público que irá consumir o produto em desenvolvimento.

Diante desse aumento do mercado de jogos supracitado, há o interesse em propor uma matriz de avaliação que possa auxiliar *Game Designer* e a equipe de *Quality Assurance* com o teste do jogo nessa trajetória de desenvolvimento e produção antes de inseri-lo no mercado.

1.6.2 Justificativa e relevância pessoal, profissional e acadêmica

Em se tratando de um Mestrado Profissional Desenvolvimento de Jogos Digitais, há a relevância de entender o cenário do mercado da indústria brasileira de jogos digitais, e com isso contribuir com o processo de gestão para o desenvolvimento do jogo. Outro ponto de interesse é aproximar a área acadêmica e as necessidades inerentes a indústria de jogos.

Com a formação profissional na área de Engenharia de Obras e Manutenção posso contribuir com a experiência nos critérios de gestão de processo e planejamento na produção de jogos digitais e adequar algumas ferramentas de gestão e qualidade utilizadas no setor Industrial e da Construção Civil e propor soluções deste viés, porte para Indústria dos Jogos Digitais.

² Game Jams é um evento onde ocorre o encontro de desenvolvedores de jogos. Em um curto espaço de tempo, cerca de vinte quatro ou setenta duas horas, a equipe de desenvolvimento tem que criar um jogo sobre um tema específico.

1.7 Ordenação dos capítulos de dissertação

Para compor o processo de pesquisa, a mesma foi organizada em seis capítulos, sendo:

O primeiro capítulo, origina-se o prelúdio desta dissertação onde são abordados a elaboração e delimitação do tema proposto, a contextualização do problema e a motivação para realização da pesquisa. Para o término deste primeiro capítulo são discorridos sobre os objetivos gerais e específicos, e as justificativas para o desenvolvimento da pesquisa.

No segundo capítulo aborda-se a metodologia da pesquisa científica, com a proposta da metodologia adotada para essa investigação, diante de seus objetivos teóricos e empíricos devidamente delineados neste trabalho. Optou-se pelos procedimentos metodológicos do *Design Science Research* utilizados para orientar essa pesquisa, preservando-se o rigor científico.

Para o terceiro capítulo verifica-se a revisão dos estudos anteriores, tendo como objetivo o levantamento bibliográfico realizado durante a investigação. Pretende-se apurar o estado da arte a respeito do tema proposto e evidenciar a análise dos dados auferidos.

O quarto capítulo é destinado a fundamentação da área tecnológica referente ao *Game Design* e desenvolvimento dos jogos digitais. Desta forma, aborda-se neste capítulo a concepção do *Game Design* com o embasamento teórico inicial e mapeamento do avanço deste gênero, assunto preponderante para esta investigação. Segue-se a discussão sobre os fundamentos no desenvolvimento e produção dos jogos digitais para o entendimento da concepção de um jogo.

No penúltimo capítulo, são relatados o conceito para elaboração da proposta da matriz de avaliação para auxílio ao *Game Designer* e *Quality Assurance*, onde é apresentado a proposição do tema discutindo o raciocínio para a implementação desta ferramenta.

O último capítulo, finda-se a presente pesquisa proposta, apresentando-se as considerações finais e sugestões para trabalhos futuros.

2. METODOLOGIA DA PESQUISA CIENTÍFICA

A permissa básica para o desenvolvimento do tema é o entendimento, a finalidade e a condução da pesquisa científica. Assim, para Gil (2002, p. 17), a pesquisa é:

[...] “o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos.

A pesquisa é desenvolvida mediante o concurso dos conhecimentos disponíveis e a utilização cuidadosa de métodos, técnicas e outros procedimentos científicos.”

Nesse mesmo pensamento, Dresch, Lacerda & Antunes Junior (2015, p. 15), definem a pesquisa como [...] “investigação sistemática, cujo objetivo central costuma ser o desenvolvimento ou refinamento de teorias e, em alguns casos, a resolução de problemas.”

Existem dois grupos quanto a natureza da pesquisa, sendo a básica ou a aplicada, conforme é apresentado por Prodanov & Freitas (2013, p. 51):

a) pesquisa básica: objetiva gerar conhecimentos novos úteis para o avanço da ciência sem aplicação prática prevista. Envolve verdades e interesses universais.

b) pesquisa aplicada: objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problema específicos. Envolve verdades e interesses locais.”

Na visão de Minayo (2002, p. 52 apud Zanella, 2013, p.32), a pesquisa científica básica [...] “permite articular conceitos e sistematizar a produção de uma determinada área do conhecimento”, enquanto a pesquisa científica aplicada na concepção de Trujillo Ferrari (1982, p. 171 apud Zanella, 2013, p.32), entende que [...] “não obstante a finalidade prática da pesquisa, ela pode contribuir teoricamente com novos fatos para o planejamento de novas pesquisas ou mesmo para a compreensão teórica de certos setores do conhecimento”.

Dessa forma a pesquisa básica tem o foco em contribuir e aprofundar o conhecimento sobre as teorias e fenômenos sobre um determinado assunto. Enquanto na pesquisa aplicada temos o conhecimento gerado no estudo e como objetivo principal uma implementação prática dirigida para identificação e a solução de problemas.

Infere-se ao pesquisador na qualidade da pesquisa científica, a busca da estratégia e a esquematização do processo para orientação e desenvolvimento desta. Assim, opta-se nesta dissertação pelo emprego da pesquisa aplicada, pois objetiva-se estudar e analisar os dados coletados a partir da criação de um artefato que contribua para tomada de decisão nas etapas de desenvolvimento de produção de um jogo digital. Sob o ponto de vista da abordagem do tema proposto, o mesmo enquadra-se na perspectiva qualitativa. Dentre as metodologia apresentadas para que a pesquisa tenha reconhecimento e rigor científico acadêmico, decidimos pelo prisma da metodologia *Design Science Research*, lançando-se de objetivos teóricos e empíricos a serem alcançados no desenlace da pesquisa.

“Devem-se incorporar as melhores heurísticas de concepção, projeto e implementação disponíveis, e fazer avançar a fronteira do estudo da técnica e quando possível, da arte. Testar, aprender, incorporar, avançar.” (DRESCH, LACERDA & ANTUNES JUNIOR, 2015, p. xvi)

2.1 Definição *Design Science*

O termo *design science* cunhado por Herbert Alexander Simon em sua obra intitulada *The Sciences of the Artificial* de 1966, onde o mesmo faz a diferenciação entre o que é a ciência natural do que é ciência artificial. Para o autor a ciência do artificial é algo produzido ou inventado pelo homem ou que sofre intervenção deste, já a ciência natural tem como característica de como esses fenômenos naturais e sociais se comportam e interagem com o mundo. Nesse momento, origina-se o despertar acerca do *design science* para sua utilização na condução de determinadas investigações de pesquisa que não conseguem ser sustentadas somente pelas ciências naturais e sociais. (DRESCH, LACERDA & ANTUNES JUNIOR, 2015)

No quadro 1, apresenta-se de forma resumida as particularidades entre as ciências naturais, sociais e *design science*.

Quadro 1 - Síntese das particularidades das ciências naturais sociais e *design science*

Principais diferenças entre a ciência tradicional e a <i>design science</i>		
Categorias	Ciências tradicionais (sociais e naturais)	<i>Design science</i>
Propósito	Entender fenômenos organizacionais com base em uma objetividade consensual, desvendando os padrões gerais e as forças que explicam esses fenômenos.	Produzir sistemas que ainda não existem, isto é, mudar sistemas organizacionais e situações existentes para alcançar melhores resultados.
Modelo	Ciências naturais (física, p.ex.) e outras disciplinas que adotaram a abordagem científica (economia, p.ex.).	<i>Design</i> e engenharia (p.ex., arquitetura, engenharia aeronáutica, ciências da computação).
Visão do conhecimento	Representacional: nosso conhecimento representa o mundo como ele é; a natureza do pensamento é descritiva e analítica. Mais especificamente, a ciência é caracterizada pela busca por conhecimentos gerais e válidos, ajustes nas formulações de hipóteses e testes.	Pragmática: conhecimento a serviço da ação; a natureza do pensamento é normativa e sintética. Mais especificamente, o <i>design</i> assume que cada situação é única e se inspira em propostas e soluções ideais, pensamento sistêmico e informações limitadas. Além disso, enfatiza a participação, o discurso como um meio de intervenção e a experimentação pragmática.
Natureza dos objetos	Fenômenos organizacionais como objetos empíricos, com propriedades descritivas e bem definidas, que podem ser efetivamente estudados de uma posição externa.	Questões organizacionais e sistemas como objetos artificiais com propriedades mal definidas, tanto descritivas como imperativas, exigindo intervenções não rotineiras por parte de agentes com posições internas na organização. Propriedades imperativas também se desdobram de fins e de sistemas idealizados de maneira mais ampla.
Foco no desenvolvimento da teoria	Descoberta da relação causal geral entre variáveis (expressas em afirmações hipotéticas): a hipótese é válida? As conclusões permanecem dentro dos limites de análise.	Será que um dado conjunto integrado de proposições de projeto funciona em uma certa situação (problema) mal definida? O projeto e desenvolvimento de novos artefatos tendem a se mover para fora das fronteiras da definição inicial da situação.

Fonte: Adaptada de Romme (2003, p. 559).

Fonte: DRESCH, LACERDA & ANTUNES JUNIOR, 2015 p. 60

Pelo comparativo, temos as ciências tradicionais que são pautadas na descoberta de leis e teorias para o entendimento destes fenômenos. E no caso da *design science*, fica a cargo da concepção, elaboração e validação desse novo sistema a ser projetado.

Conforme descreve Dresch, Lacerda & Antunes Junior (2015, p. 57) a respeito o *desing science*:

“[...] é a ciência que procura desenvolver e projetar soluções para melhorar sistemas existentes, resolver problemas ou, ainda, criar novos artefatos que contribuem para uma melhor atuação humana, seja na sociedade, sejam nas organizações.”

Denota-se que a base que sustenta a metodologia é a criação de novos artefatos para resolução dos problemas que se apresenta, permitindo que se faça a experimentação e com isso a produção do conhecimento estabelecido durante o processo de investigação.

2.2 Conceitos da *Design Science*

Após apresentarmos a definição da ciência do artificial, buscamos entender os principais conceitos inerentes no emprego desse método. Nesse aspecto, o que podemos entender como artefato para a metodologia *design science*?

No teor da palavra artefato o seu significado descreve algo feito ou dado forma pelo homem, qualquer objeto ou produto que tenha um objetivo determinado. Nesse mesmo pensamento dito anteriormente a respeito da ciência do artificial por Simon (1996), aborda sua visão do artefato a luz do *design science*:

[...] objetos artificiais que podem ser caracterizados em termos de objetivos, funções e adaptações. São normalmente discutidos, particularmente durante a concepção, tanto em termos imperativos como descritivos [...]

[..]o artefato é a organização dos componentes do ambiente interno para atingir objetivos em um determinado ambiente externo [...]

(SIMON, 1996, p. 28 apud LACERDA ET AL., 2013, p. 748).

Os artefatos são projetados para um finalidade específica, visando atender a um determinado problema. Desta forma, para que se cumpra o seu objetivo, temos que considerar a viabilidade da solução, o método a ser empregado, elencar os possíveis artefatos gerados e suscitar o ponto de encontro (interface) entre o ambiente interno e um ambiente externo como preconiza Simon.

Para March & Smith (1995), o conhecimento gerado pelo *Design Research* pode assumir várias formas incluindo constructos, modelos, métodos, instanciações. (HEVNER & CHATTERJEE, 2010 p.15). Estes seriam os tipos de artefatos produzidos durante a investigação, no intuito de elucidar, são representadas cada tipo de artefato no Quadro 2, como segue:

Quadro 2 - Tipos de artefatos

Descrição		
Tipos de Artefatos	Constructos	Constructos ou conceitos formam o vocabulário de um domínio. Eles constituem uma conceituação utilizada para descrever os problemas dentro do domínio e para especificar as respectivas soluções. Conceituações são extremamente importantes em ambas as ciências (natural e de design). Eles definem os termos usados para descrever e pensar sobre as tarefas. Eles podem ser extremamente valiosos para designers e pesquisadores.
	Modelos	Um modelo é um conjunto de proposições ou declarações que expressam as relações entre os constructos. Em atividades de design, modelos representam situações como problema e solução. Ele pode ser visto como uma descrição, ou seja, como uma representação de como as coisas são. Cientistas naturais muitas vezes usam o termo 'modelo' como sinônimo de 'teoria', ou 'modelos' como as teorias ainda incipientes. Na <i>Design Science</i> , no entanto, a preocupação é a utilidade de modelos, não a aderência de sua representação à Verdade. Não obstante, embora tenda a ser impreciso sobre detalhes, um modelo precisa sempre capturar a estrutura da realidade para ser uma representação útil.
	Métodos	Um método é um conjunto de passos (um algoritmo ou orientação) usado para executar uma tarefa. Métodos baseiam-se em um conjunto de constructos subjacentes (linguagem) e uma representação (modelo) em um espaço de solução. Os métodos podem ser ligados aos modelos, nos quais as etapas do método podem utilizar partes do modelo como uma entrada que o compõe. Além disso, os métodos são, muitas vezes, utilizados para traduzir um modelo ou representação em um curso para resolução de um problema. Os métodos são criações típicas das pesquisas em <i>Design Science</i> .
	Instanciações	Uma instanciação é a concretização de um artefato em seu ambiente. Instanciações operacionalizam constructos, modelos e métodos. No entanto, uma instanciação pode, na prática, preceder a articulação completa de seus constructos, modelos e

	métodos. Instanciações demonstram a viabilidade e a eficácia dos modelos e métodos que elas contemplam
--	--

Fonte: Extraído de March e Smith (1995, p. 257-258) apud Lacerda et al. (2013 p. 749).

Outros conceitos abordados por Dresch, Lacerda & Antunes Junior (2015), é que as soluções que atendam o problema em questão devem ser viáveis, não necessariamente ótimas. E ainda, destaca-se a divisão em classe de problemas, que podem ser de ordem prática ou teórica, no qual ajuda na organização e serve de orientação para o desenvolvimento da pesquisa com os princípios da *Design Science*. Os autores buscam ressaltar a importância do conceito de validade pragmática, onde o interesse além de atender a validação científica, objetiva-se assegurar a utilidade da solução proposta para o problema, atenda às necessidades dos interessados, além do seu funcionamento.

A Design Science é interativa e o artefato inicial, produto da solução proposta para um problema, precisa ser avaliado e redesenhado até que seja encontrada uma solução satisfatória (BALOH; DESOUSA; HACKNEY, 2012)

Quadro 3 – Resumo dos conceitos da Design Science

Conceito de Design Science	Ciência que procura projetar e desenvolver soluções para melhorar sistemas existentes, resolver problemas, criar novos artefatos.
Artefato	Algo que é construído pelo homem; interface entre o ambiente interno e o ambiente externo de um determinado sistema.
Soluções satisfatórias	Soluções suficientemente adequadas para o contexto em questão. As soluções devem ser viáveis para a realidade. Não precisa ser uma solução ótima.
Classes de Problemas	Organização que orienta a trajetória e o desenvolvimento do conhecimento no âmbito da <i>Design Science</i> .
Validade Pragmática	Busca assegurar a utilidade da solução proposta para o problema. Considera: custo/benefício da solução, particularidades do ambiente em que será aplicada e as reais necessidades dos interessados na solução.

Fonte: DRESCH, 2013

2.3. Design Science Research

Para Dresch, Lacerda & Antunes Junior (2015, p. 67), a *Design Science* é a base epistemológica do estudo do artificial. Por sua vez, a metodologia *Design Science Research* fundamenta e operacionaliza a condução da pesquisa quando o propósito é um artefato ou uma prescrição. Em complemento a esse raciocínio os autores afirmam:

“É uma forma de produção de conhecimento científico que envolve o desenvolvimento de uma inovação, com a intenção resolver problemas do mundo real e, ao mesmo tempo, fazer uma contribuição científica de caráter prescritivo. Este tipo de pesquisa produz como resultado um artefato que representa uma solução para uma ampla gama de problemas, também denominado conceito de solução, que deve ser avaliado em função de critérios relacionados à geração de valor ou utilidade.” (DRESCH, LACERDA & ANTUNES JUNIOR, 2015, p. xiii)

Para que possamos atender os requisitos que permeiam a produção técnica e a produção científica, é importante balizarmos os dois interesses para que tenhamos o objetivo principal o desenvolvimento de um artefato. Nesse sentido os autores Dresch, Lacerda & Antunes Junior (2015) com base nas orientações de Hevner et al. (2004), descrevem a necessidade de termos o rigor e relevância para êxito na pesquisa.

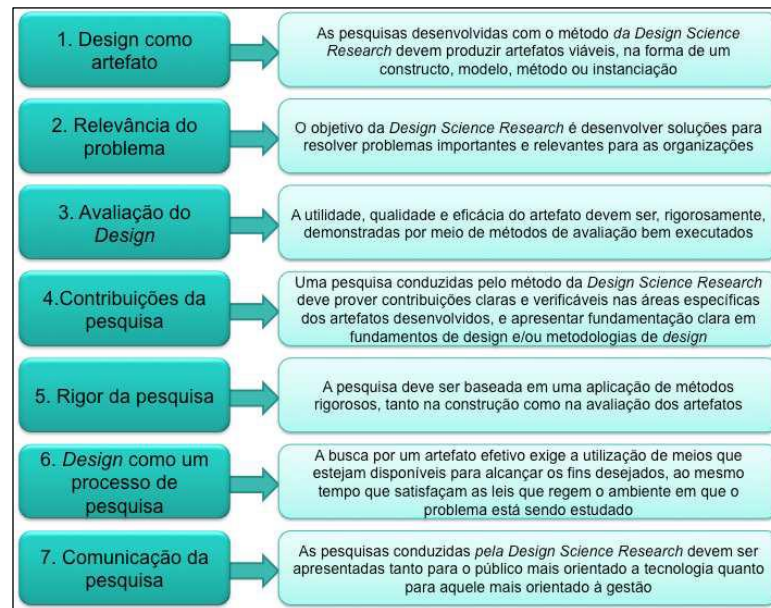
Dessa forma temos a relevância da pesquisa no ambiente onde serão retiradas as informações pertinentes na busca da solução e o local onde serão aplicadas na prática. Com isso, infere-se as observações do fenômeno e adequações para a construção do artefato. E de outro lado, temos o rigor para que a pesquisa seja considerada válida, embasada na coleta do conhecimento teórico para formarmos nosso arcabouço e aumentarmos a base de conhecimento naquela área. Assim, a base de conhecimento é agrupada em fundamentos e métodos reconhecidos pela academia.

Quadro 4 – Framework da DSR



Fonte: DRESCH, LACERDA & ANTUNES JUNIOR, 2015 p. 68

Para a condução da pesquisa metodológica sob égide *Design Science Research* o pesquisador Alan Hevener et al (2004), definiu sete etapas a serem seguidas para o desenvolvimento da pesquisa são elencadas da seguinte forma:

Quadro 5– Etapas na condução da pesquisa DSR

Fonte: DRESCH, 2013

Em suma, a metodologia científica *Design Science Research* é baseada na criação de artefatos para entendimento e orientada a solução do problema de pesquisa, de forma pragmática em busca do aprimoramento de teorias e soluções estudadas durante a investigação. Isso implica na validação do artefato pautados no rigor e relevância para que obtenha funcionalidade desejada no ambiente onde foi realizado o estudo e que contribua com área acadêmica com os resultados obtidos no decorrer do estudo.

2.3.1 Modelos para condução da *Design Science Research*

Há diversos modelos para condução da pesquisa valendo-se da metodologia *Design Science Research*, com a proposição de desenvolver artefatos e conceber conhecimento teórico. Os pesquisadores utilizaram o conceito *Design Science* e foram desenvolvendo seus modelos, em busca do método que melhor atendesse aos requisitos da *Design Science*. Essas propostas contribuíram para evidenciar a metodologia *Design Science Research* e fomentar a sua utilização em novas pesquisas e artigos na academia, em especial as áreas de engenharia e tecnologia como o Sistema de Informação. Para Larsen et al. (2020), *Design Science Research*

se tornou importante na última década para a comunidade de Sistema de Informação, onde houve o aumento do número de publicações em congressos a respeito da metodologia.

Nessa dissertação não há pretensão de explicitar cada um dos modelos propostos, ao invés disso, será apresentado o modelo escolhido para desenvolvimento da metodologia e condução da pesquisa científica. O modelo apresentado e proposto por Dresch, Lacerda & Antunes Junior (2015, p. 124-133), é composto por doze etapas principais como segue:

- a) **Identificação do problema** – investigar sobre o olhar da metodologia *Design Science Research* o problema de interesse do pesquisador, encontrado resposta para uma questão importante ou solução para um problema prático ou uma classe de problemas. Pede-se ao pesquisador justificar a importância e relevância do objeto de estudo de forma clara e objetiva.
- b) **Conscientização do problema** – o pesquisador busca nessa etapa o máximo de informações possíveis a respeito do problema, assegurando-se da completa compreensão das suas especificidades, causas e contextos. Considera-se também as funcionalidades do artefato, performance esperada e seus requisitos de funcionamento.
- c) **Revisão sistemática da literatura** – destina-se na busca da base de conhecimento, tanto o conhecimento gerado a partir das ciências tradicionais, quanto aquele fundamentado na *Design Science*.
- d) **Identificação dos artefatos e configuração das classes de problemas** – realizado a etapa anterior, apoiará o pesquisador a evidenciar, caso existam, as Classes de Problema e artefatos relacionados ao que ele está tentando resolver.
- e) **Proposição de artefatos para resolução do problema** – identificados os artefatos e Classe de Problemas, quando houver, bem como as soluções satisfatórias formalizadas. Se faz necessário essa etapa, pois na etapa

anterior tratava-se da visualização de possíveis artefatos genéricos para resolver um problema genérico. Sendo assim, o pesquisador propõe artefatos viáveis à realidade em estudo, o seu contexto de atuação, etc. Desenvolve-se raciocínio abduutivo, sendo uma etapa essencialmente criativa, aliado aos conhecimentos prévios, com o intuito de propor soluções robustas para melhoria da situação atual.

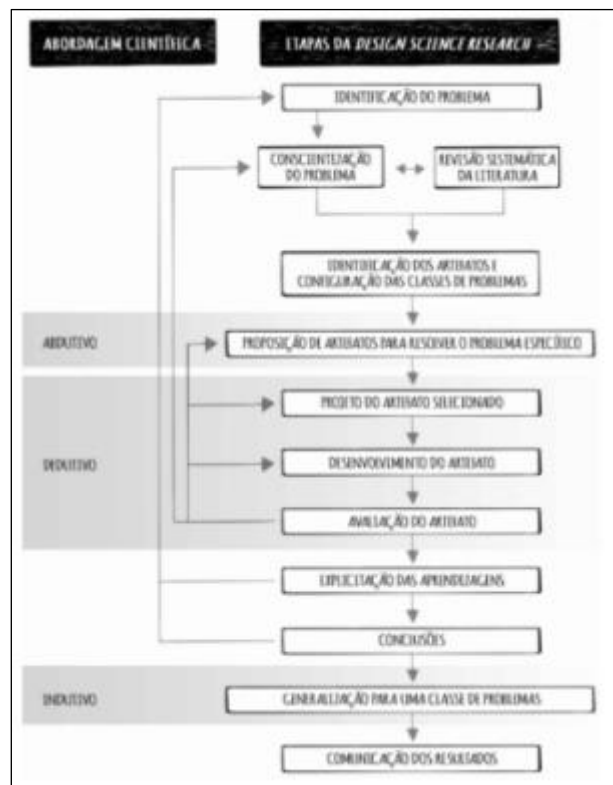
- f) **Projeto do artefato** – ocorre a seleção de um dos artefatos propostos anteriormente. Selecionado o artefato o projeto deve considerar as características internas e externas desse artefato (componentes, relações internas de funcionamento, limites, contexto onde será operado). É importante que o pesquisador descreva todos os procedimentos de construção e avaliação do artefato. E cabe também informar o desempenho esperado, que vai garantir uma solução satisfatória para o problema.
- g) **Desenvolvimento do artefato** – pode ser utilizadas diferentes abordagens para construção do artefato, sendo: algoritmos computacionais, representações gráficas, protótipos, maquetes, etc. Objetiva-se gerar conhecimento que seja aplicável e útil para a solução do problemas. Ao final temos o artefato em seu estado funcional e a heurística da construção, o desenvolvimento do artefato.
- h) **Avaliação do artefato** – o pesquisador observa e mensura o comportamento do artefato na solução do problema. A avaliação deve ser conduzida em um ambiente experimental ou em um contexto real, No caso, o artefato for do tipo instanciãção, precisa obrigatoriamente ser aplicado e analisado no ambiente real. Neste caso, outros métodos de pesquisa, como pesquisa-ação, por exemplo, poderão ser utilizados, pois muito provavelmente haverá a necessidade de interação entre o pesquisador, usuários e as pessoas da organização na qual o artefato está instanciado.
- i) **Explicitação das aprendizagens e conclusão** – considerando-se que o artefato atingiu os resultados esperados após a etapa de avaliação, se faz necessário os apontamentos obtidos durante o processo de pesquisa,

apresentando os pontos de sucesso e insucesso. Assegura-se assim que a pesquisa realizada possa servir de referência e como subsídio para a geração de conhecimento na teoria e na prática.

- j) **Generalização para uma classe de problemas e comunicação dos resultados** – após a conclusão da pesquisa, o artefato desenvolvido e estabelecido o processo de construção do mesmo, o importante que haja uma generalização para uma classe de problemas. Assim, o conhecimento gerado em uma situação específica possa, posteriormente ser aplicado a outras situações similares e que são enfrentadas por diversas organizações. Por fim, é essencial que se tenha a comunicação dos resultados, em diversos veículos acadêmicos e organizações. A veiculação do conhecimento gerado contribui para o avanço do conhecimento geral.

Quadro representativo das etapas do método *Design Science Research* proposto:

Quadro 6 – Método DSR



Fonte: DRESCH, LACERDA & ANTUNES JUNIOR, 2015 p. 124

2.4 Emprego do método *Design Science Research*

Feitas todas as premissas para o desenvolvimento da metodologia *Design Science Research*, vamos aplicá-la na condução da nossa pesquisa.

2.4.1 Emprego do método *Design Science Research*

3. REVISÃO ESTUDOS ANTERIORES

Nesse capítulo aborda-se o estado da arte com a referência de estudos realizados no tempo e espaço de 2010 a 2020, com o intuito do levantamento de trabalhos realizados por outros pesquisadores relativos ou que se aproximam ao tema proposto. A investigação abrange dois cenários: 1) A gestão na produção e desenvolvimento de jogos digitais. 2) A utilização de ferramentas de qualidade no processo de desenvolvimento dos jogos digitais.

A base de dados utilizada para a pesquisa acadêmica foram abordados anteriormente, sendo: a) Banco de Teses e Dissertações da Pontifícia Universidade Católica de São Paulo (PUC-SP); b) Banco de Teses e Dissertações da Capes; c) Sociedade Brasileira de Games (SBGames); d) *Scientific Eletronic Library Online* (SciELO); e) Biblioteca Digital Brasileira de Teses e Dissertações (BDTD)

3.1 Produção nacional

Mapeamento bibliográfico da produção nacional, com os apontamentos das produções que se assemelham ao tema. As palavras-chaves para busca foram: “*produção de jogos digitais; desenvolvimento de jogos digitais; garantia de qualidade; controle de qualidade; avaliação de qualidade*”. Com base definida, encontramos o seguintes resultados para as palavras definidas entre as configurações possíveis para o objeto de busca “*produção de jogos digitais and garantia de qualidade*”, “*produção*

de jogos digitais **and** controle de qualidade” , “produção de jogos digitais **and** avaliação de qualidade”. O mesmo foi feito para o segundo objeto de busca “desenvolvimento de jogos digitais **and** garantia de qualidade” , “desenvolvimento de jogos digitais **and** controle de qualidade” , “desenvolvimento de jogos digitais **and** avaliação de qualidade”.

Tabela 1- Resultado de busca em Língua Portuguesa

BANCO DE BUSCA	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	TOTAL PERÍODO
PUC-SP	0	0	0	0	0	0	0	0	0	0	0	0
CAPES	12	25	25	28	21	19	22	34	31	30	0	247
SBGames	0	0	0	0	0	0	0	0	0	0	0	0
BDTD	1	1	1	1	2	2	5	6	3	2		24
SciELO	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL GERAL												271

Fonte: Produzido pelo autor

Após essa primeira etapa, realizamos uma segunda avaliação encontrando artigos, dissertações e teses que se assemelham ao tema proposto. Encontramos a dissertação de 2017, “*Playtest e Design Research: o teste do jogo sob perspectiva da pesquisa de Design Science*”, do autor Marco Akira Miura, aborda-se a aplicação do DSR como procedimento de *playtest* para o estudo de caso da análise do protótipo do jogo Apsys proposto.

3.2 Produção internacional

Mapeamento bibliográfico da produção internacional, com os apontamentos das produções que se assemelham ao tema. Na busca foram utilizados os termos: “*game development process, game production, quality assurance, control quality, quality assessment*”. Na busca utilizando os filtros chegamos ao seguinte resultado:

Tabela 2 – Resultado de busca em Língua Inglesa

BANCO DE BUSCA	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	TOTAL PERÍODO
PUC-SP	0	0	0	0	0	0	0	0	0	0	0	0
CAPES	2	2	1	34	43	42	36	54	45	34	0	293
SBGames	0	0	0	0	0	0	0	0	0	0	0	0
BDTD	0	0	0	0	0	0	1	0	3	0	0	4
SciELO	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL GERAL												297

Fonte: Produzido pelo autor

Após o critério de exclusão, foram encontrados artigos, dissertações relevantes ao tema proposto, sendo:

De 2015, a dissertação “*Quality Assurance Testing in Video Games – The importance and impact of misunderstood industry*”, o autor Essi Ruuska explana sobre a definição de garantia de qualidade nos videogames e qual seu impacto no produto final. O autor conclui a importância da garantia da qualidade no desenvolvimento dos jogos, pois ajuda na detecção de falhas, evitar perdas financeiras e comerciais.

Em 2019, temos o artigo “*Quality of Video Games: Introduction to a complex Issue*”, do autor Maciej D. Sobocjnki, este apresenta uma visão da indústria do vídeo game e seu rápido crescimento, a competição acirrada do mercado, tornando um negócio de alto risco. Outro ponto abordado é a dificuldade de se empregar métodos, frameworks da área de qualidade na indústria do vídeo game

4. ABORDAGEM TEÓRICA

Jogos acompanham a história da humanidade, desde a época das antigas civilizações, em cada período temos algum ou outro jogo que transcende a existência daquela civilização, e ainda permanecem em nosso cotidiano e são jogados de geração em geração.

Mas afinal, o que é um jogo eletrônico? Quais são as particularidades que envolvem cada tipo de um jogo eletrônico? Por que alguns tipos de jogos eletrônicos são tão consagrados e outros caem no esquecimento, quais são os fatores que influenciam essa questão? É isso que pretendemos abordar neste capítulo e entender as conexões que estabelecem entre o jogo e o jogador.

4.1 Jogo, o começo de tudo um breve contexto histórico

Os jogos remontam à época das antigas civilizações, alguns deles como o *Senet* jogo da mitologia egípcia, os jogos de estratégia *Wei-qi* e *Xiang Qi* de origem chinesa, o *Shogi* oriundo do Japão ou o predecessor do Xadrez o jogo indiano *Chaturanga*.

“A humanidade jogou ao longo da sua história. Se olharmos para os dados e o tabuleiro dos primitivos jogos da tumba do Rei Tutankamon ou o grafite representando os tabuleiros usados por patrícios aguardando o Fórum Romano, as pessoas deixaram artefatos indicando o jogo como parte do seu legado.” (DEMARIA, 2019, p. 2, tradução nossa)³.

Interessante é observar que desde aquele período a humanidade teve momentos de abstração e criou modelos ou por alguma representação de elementos gráficos que refletisse sua cultura, aspectos sociais ou de ordem militar. Podemos dizer que os jogos também entram nesse contexto, no entanto para Burgun (2013), a construção dos jogos nunca foi encarada como um ofício como a música, a escrita e

³ “Humankind has gamed throughout its history. Whether we look at the dice and primitive board games from King Tut’s tomb or the graffiti representing game boards used by waiting patricians in the Roman Forum, people have left artifacts indicating play as part of their legacy.”

as artes visuais. Apesar da criação dos jogos, nunca houve até então uma classe específica chamada *game designer*.

Um modelo de jogo que sofreu várias modificações com decorrer do tempo foi o jogo de Xadrez, conforme houve a disseminação do jogo em algumas regiões como o Oriente Médio, Ásia e posteriormente a Europa. Diante da sua popularização, o Xadrez foi adquirindo elementos culturais daquela região e também sofreu mudanças nas mecânicas, *design* e regras sendo incorporadas ao jogo, até o formato atual padronizado e reconhecido mundialmente desde 1924 pela FIDE⁴.

Podemos contextualizar algumas dessas mudanças como a introdução do “xeque” e “xeque-mate” pelos persas, com os europeus houve a alteração na movimentação das peças como o peão, no primeiro lance há a oportunidade de avançar duas casas. E a possibilidade de se tornar uma peça mais poderosa ao atravessar o tabuleiro na chegada da primeira linha do adversário, pode-se dizer um dos primeiros “*level up*” conhecidos, característicos e presentes nos jogos de vídeo game. Uma das modificações mais significativas foi o advento da dama por volta do séc. XV, além da presença da figura feminina em substituição a figura do vizir para os árabes, trouxe uma inovação na mecânica da peça, podendo transitar pelo tabuleiro em todas as direções em vista ao do vizir que movia-se uma casa por vez na diagonal. Outras peças sofreram suas alterações na mecânica e no *design*, como o elefante e o carro de guerra, antecessores do bispo e a torre respectivamente.

Assim, estava presente as características da cultura medieval europeia e com as implementações que ocorreram no jogo houve uma maior dinâmica, aumento da complexidade e estratégia durante a partida. Nesse processo percebe-se em suas devidas proporções, o início dos atributos iguais a de um *game designer* no desenvolvimento do seu jogo.

Em outro período, no fim do séc. XIX e início do séc. XX na Inglaterra, foram criadas máquinas automáticas baseadas em esportes da época como o críquete, essa

⁴ Federação Internacional de Xadrez (*Fédération Internationale des Échecs*), fundada em 20 de julho de 1924, em Paris.

máquina era operada por moeda e simulava um recorte da partida de críquete, que neste caso, focava nas rebatidas e lançamentos do jogo. Outro modelo desse tipo de máquina foi o jogo “*Climbing Fireman*”, após cada jogador inserir a moeda para o início da partida, os jogadores tinham que girar rapidamente a manivela para que o bombeiro atingisse o mais rápido possível o topo da escada, definindo o vencedor e a devolução da moeda inserida na máquina. (WILLIAMS, 2017)

Figura 2 - Climbing Fireman (1920)



Fonte: WILLIAMS, 2017 p.11

E ainda, este tipo de máquina ficou popular no Estados Unidos, principalmente nos locais de centro de entretenimento, conhecidos como “*Arcade*”. Logo alavancou

cada vez mais os negócios desse tipo de diversão com a produção local e o surgimento de novas máquinas mais sofisticadas melhorando o sistema de interação entre a máquina e o jogador. Em alguns casos, como por exemplo a máquina de 1929, “*Play the Derby*” que simulava o Turfe, cada jogador ao girar a manivela tinha que conduzir seu respectivo cavalo na pista por duas volta, os engenheiros mecânicos desenvolveram um sistema com uma embreagem que escorregava quando o jogador girasse a manivela muito rápido, fazendo com que o cavalo parasse de forma abrupta. Dessa forma, além de deixar a partida mais interessante o mecanismo do jogo seria menos danificado. (WILLIAMS, 2017)

Há um novo elemento como incremento na interação, permitindo um maior desafio durante a competição. Outro ponto interessante a ser observado, é que o dispositivo foi criado com a intenção de reduzir a manutenção da máquina, na busca da solução de um problema foi incorporado ao elemento do jogo.

Nesse aspecto de inovação, inspirado no jogo francês “*Batelle*”, as conhecidas máquinas de *pinball* sofreram inúmeros processos de melhoria até chegar ao modelo de *pinball* que conhecemos atualmente. Segundo Williams (2017), nos Estados Unidos em 1930, tiveram as primeiras máquinas operadas por moedas, uma delas a “*Baffle Ball*” criada por David Gottlieb, devido ao baixo custo da partida a um centavo de dólar, se tornou extremamente popular nesta época vendendo cerca de cinquenta mil unidades neste ano. O objetivo era fazer com que a esfera de metal impulsionada pelo êmbolo atingisse a maior pontuação entre as “caçapas” da mesa inclinada. A esfera após impulsionada poderia ser desviada do seu trajeto devido aos pinos colocados na mesa, a cada partida eram liberadas sete esferas, sendo uma por vez.

Devido ao sucesso no Estados Unidos, outras empresas decidiram fabricar suas máquinas e lançar no mercado novos modelos e incorporando novas tecnologias com elementos eletromecânicos e sinópticos no equipamento. Entre 1930 a 1940, houve uma forte repressão no Estados Unidos nas diversões consideradas como jogos de azar e várias máquinas foram apreendidas, como as máquinas caça-níqueis e também estava incluso nesse questionamento as máquinas de *pinball*. No entender da Justiça americana a dinâmica que constituía o jogo de *pinball*, o desfecho da trajetória da esfera era aleatório não dependia totalmente da habilidade do jogador

sobre o jogo, outro ponto foi a possibilidade de ganhar bonificação de jogo grátis por atingir determinada pontuação ou apostas realizadas com pagamento para pontuação mais alta. Diante desse impasse, em 1947 foi introduzido na máquina “*Humpty Dumpty*” aletas movidas por solenoide ou flippers⁵, no qual permitiu para que o jogador obtivesse maior controle sobre a esfera na mesa.

Esta foi uma forma inovadora na tentativa para que o *pinball* não fosse caracterizado como jogo de azar, nesse exemplo temos outro artefato que foi criado na resolução de um problema. Essa implementação fez um grande sucesso, fazendo com que as demais empresas adequassem suas máquinas com esse novo dispositivo, um marco que até hoje são utilizados na máquinas atuais ou virtuais.

Figura 3 – Pinball Humpty Dumpty Gottlieb, publicação da inovação na revista Billboard - 1947



Fonte: CROCI, 2015

⁵ *Flippers Bumpers* – Termo designado para as aletas que rebatiam a esfera modificando sua trajetória, quando acionadas pelo jogador por botões nas laterais da máquina. Principal meio do jogador controlar a esfera, ajudando-o no aumento da marcação de pontos durante a partida. No Brasil, o termo *flíper* é nome utilizado para as máquinas de *pinball* e fliperama para o centro de diversões

Figura 4 - Jogo de futebol



Conforme apresentado na figura 1, em cada época temos diferentes formas de abstração para a simulação de uma partida de futebol e o mesmo conceito de interação de jogar futebol. A primeira figura, da esquerda para a direita, temos a máquina mecânica *Play Football*⁶ construída em 1926 pela a empresa *Chester-Pollard Amusement Company*, a mecânica do jogo é semelhante ao pebolim com alavanca para o chute da esfera de aço com o objetivo de marcar gol em seu adversário. Já na figura ao centro, temos o cartucho *International Soccer* lançado pela Atari para seu videogame VCS 2600 em 1977, através da alavanca do controle (*joystick*) para movimentação do jogador na tela e um botão de ação para passe e chute. Na terceira figura temos o renomado FIFA 20 lançado em 2019 para diversas plataformas, o jogador passa a comandar o personagem pelo controle com duas alavancas analógicas e vários botões de acionamento para tomada de decisão de passes, dribles e movimentações dentro do campo.

4.2 Crash 1983 – Crise no mercado norte-americano de jogos digitais e a Qualidade

Nesse item aborda-se a recessão na indústria do vídeo game que ocorreu entre 1983 a 1985. Quais foram os aspectos e fatores envolvidos que ocasionaram esta crise, saturação do mercado, outros fatores? Qual era o controle de qualidade e como era feito neste período.

⁶ Apresentação da máquina *Play Football* <https://www.youtube.com/watch?v=fiPtW5Ge7Hk>

4.3 Novos rumos selo de qualidade na indústria do vídeo game

Nessa seção pretende-se abordar o surgimento do selo de qualidade da empresa Nintendo, quais foram os motivos para implantação e o que acarretou ao mercado da indústria dos vídeo games após a sua implantação.

4.4 Jogos *Arcade* padrão de referência dos jogos

Nessa seção pretende-se discutir a respeito da era de “ouro” dos jogos de arcade nas décadas de 1980 e 1990, onde os jogos de consoles de videogame buscavam o padrão dos jogos *Arcade* como referência em qualidade.

4.5 Fundamentos do Game Design

4.5.1 Qual é a definição de jogo

Segundo Sales e Zimmerman (2004, p. 83), estudos acerca da definição de jogo é pauta de diversos autores, essa indagação estende-se ao longo do tempo. O historiador de jogos David Parlett, comenta que a palavra jogo é usada para tantas atividades diferentes que não vale a pena insistir em nenhuma definição proposta proposição. Entretanto, os autores argumentam por ser objeto central do estudo do *game design* é fundamental tentar defini-la nesse domínio.

Para Sales e Zimmerman (2004, p. 97), não existe distinção na definição “jogo” entre ele ser um jogo digital e não digital, as qualidades que definem um jogo em uma mídia também definem em outra.

Em busca de um melhor entendimento, Sales e Zimmerman (2004, p. 86-91) discorrem sobre a definição de jogo sob ótica de oito autores, sendo:

- **David Parlett:** distingue em jogos informais (brincar, brincadeira não dirigida) e jogos formais. O jogo formal tem uma estrutura dupla baseada em fins e meios.

Fins, é uma competição para atingir um objetivo, esse objetivo é vencer. O jogo é encerrado somente quando um dos competidores, sejam disputas individuais ou em equipes, vence. Logo, vencer é “fim” do jogo em ambos os sentidos da palavra, com término e como objetivo.

Meios, o jogo tem um conjunto acordado de equipamentos e de “regras” procedimentais por meio das quais o equipamento é manipulado para produzir uma situação vencedora.

Essas duas ideias, vencer e por meio de regras definem e distinguem dos jogos menos “formais”

- **Clark C. Abt:** reduzido à sua essência formal, um jogo é uma atividade entre dois ou mais tomadores de decisão independentes que buscam atingir seus objetivos em algum contexto limitador. Uma definição mais convencional diria que um jogo é um contexto com regras entre adversários que tentam vencer objetivos.

Temos quatro termos-chaves que se destacam para compreensão de jogo sendo uma atividade, um processo, um evento e tomada de decisão.

- **Johann Huizinga:** [brincar é] uma atividade livre, ficando conscientemente tomada como “não séria” e fora da vida comum, mas ao mesmo tempo absorvendo o jogador intensa e totalmente. É uma atividade desprovida de interesse material e nenhuma obtenção de lucro. É praticada dentro de seus próprios limites de tempo e espaço consoante com regras fixas e de forma ordenada. Promove a formação de agrupamentos sociais, que tendem a se cercar de sigilo e enfatizar sua diferença em relação ao mundo comum, por meio de disfarces ou outros meios.

- **Roger Caillois:** Livre, é uma atividade não obrigatória, se fosse perderia imediatamente a atraente e alegre qualidade de diversão.

Separada, circunscrita em limites de espaço e tempo, definidas e fixadas previamente.

Incerta, o curso do qual não pode ser determinado, nem o resultado alcançado antecipadamente, e alguma margem para inovações deixadas para a iniciativa do jogador.

Improdutiva, não cria bens, nem riqueza, nem novos elementos de qualquer tipo. Exceto para troca de bens entre os jogadores, culminando em uma situação idêntica à que prevalecia no início do jogo.

Regido por regras, sob convenções que suspendem as leis ordinárias e, por enquanto, estabelecem uma nova legislação, que conta sozinha

Faz de conta, acompanhado de uma consciência especial de uma segunda realidade ou fantasia livre, em relação à vida real.

- **Bernard Suits:** a interação em um jogo é engajar-se em uma atividade focada na realização de um determinado estado de coisas, usando apenas os meios permitidos pelas regras, onde as regras proíbem os mais eficientes em prol dos meios menos eficientes, e onde tais regras são aceitas apenas porque possibilitam essa atividade.

Em suma, jogar é um esforço voluntário para superar obstáculos desnecessários.

- **Chris Crawford:** é a representação de quatro qualidade principais que definem essa categoria de jogos.

Representação, um jogo é um sistema formal fechado que representa subjetivamente um subconjunto da realidade.

Entende-se por “fechado” que o jogo é completo e autossuficiente como estrutura. Esse modelo de mundo criado é internamente completo, nenhuma referência precisa ser feita a agentes fora do

jogo. E por “formal”, apenas o jogo tem regras explícitas. Um jogo é uma coleção de partes que interagem umas com as outras, geralmente de forma complexa. É um sistema. Um jogo cria uma representação subjetiva e deliberadamente simplificada da realidade emocional.

Interação, a coisa mais interessante sobre a realidade não é o que é, ou mesmo que muda, mas como ela muda, uma emaranhada rede de causa e efeito pela qual todas as coisas estão interligadas. A única forma de representar adequadamente essa rede é permitir que o público explore para deixá-los gerar causa e observar seus efeitos. Os jogos fornecem esse elemento interativo, é um fator primordial em seu apelo.

Conflito, um terceiro elemento que aparece em todos os jogos. O conflito surge naturalmente da interação em um jogo. O jogador está ativamente perseguindo algum objetivo, obstáculos o impedem de atingir facilmente esse objetivo. O conflito é um elemento intrínseco a todos os jogos. Pode ser direto ou indiretamente, violento ou não-violento, mas sempre está presente em cada jogo.

Segurança, o conflito implica perigo, perigo significa risco de dano, dano é indesejável. Portanto, um jogo é um artifício para proporcionar experiências psicológicas de conflito e perigo, ao mesmo tempo que exclui suas realizações físicas. Em resumo, um jogo é uma forma segura de experimentar a realidade. Os resultados de um jogo são sempre menos severos do que as situações que o jogo modela.

- **Greg Costikyan:** um jogo é uma forma de arte na qual os participantes, denominados jogadores, tomam decisões a fim de gerenciar recursos por meio das fichas do jogo na busca de um objetivo.
- **Elliot Avedon e Brian Sutton-Smith:** jogos são um exercício de sistemas de controle voluntário, no qual ocorre uma disputa entre

forças, limitadas por regras, a fim de produzir um resultado que cause um desequilíbrio.

Na análise de Sales e Zimmerman (2004, p. 92-93), constata que não há um consenso claro na definição de jogo e nenhum dos autores fazem a reflexão no domínio do *game design*. Na perspectiva dos autores, “um jogo é um sistema no qual os jogadores se envolvem em um conflito artificial, definido por regras que implica em um resultado quantificável”. As principais ideias relacionadas a este pensamento são apresentadas a seguir:

- **Sistema:** são fundamentais para os jogos.
- **Jogadores:** os jogadores interagem com o sistema de um jogo para experimentar o jogo. É formado por um mais participantes jogam ativamente.
- **Artificial:** os jogos mantêm uma fronteira da chamada “vida real” no tempo e espaço. Embora os jogos obviamente ocorrem no mundo real, a artificialidade é uma das suas características que a definem.
- **Conflito:** todos os jogos incorporam uma competição de forças. A disputa pode assumir várias formas, da cooperação à competição, do conflito solo com o sistema do jogo a um conflito social multijogador. O conflito também é fundamental para os jogos.
- **Regras:** as regras são fundamentais e fornecem a estrutura da qual o jogo emerge, delimitando o que o jogador pode ou não executar.
- **Resultado quantificável:** os jogos têm um objetivo ou resultado quantificável. Na conclusão de um jogo, o jogador ganha, perde ou recebe algum tipo de pontuação numérica. Um resultado quantificável é o que geralmente distingue um jogo de atividades lúdicas menos formais.

Juul (2005, p. 74-75 e 262), em seu estudo propõe uma estrutura com os requisitos necessários para que um modelo seja considerado como jogo. O autor teve

como desígnio de explicar a relação dos jogos de computador a outros tipos de jogos e o que acontece nas fronteiras do campo dos jogos. O videogame é combinação de regras e ficção. As regras definem o que pode ou não ser feito e oferece desafios que o jogador deve gradualmente aprender a superar. A estudo é norteado por três premissas para uma boa definição de um jogo, como: os tipos de sistemas definidos pelas regras de um jogo, a relação entre jogo e jogador e relação do jogador com o resto do mundo. Após o processo de análise na definição de outros teóricos e *designers* a respeito do tema, o autor concebe a sua definição, com seis pontos principais:

- **Regras fixas:** um sistema é baseado em regras. As regras devem ser bem definidas para que não se tenha nenhum tipo de desentendimento ou discutir sobre elas toda vez que jogar.
- **Resultado variável e quantificável:** os jogos têm resultado variáveis e quantificáveis.
- **Valor atribuído aos resultados possíveis:** os diferentes resultados potenciais do jogo recebem valores diferentes, alguns sendo positivos ou negativos.
- **Esforço:** que o jogador invista esforço para influenciar o resultado, ou seja, os jogos são desafiadores.
- **Apego do jogador ao resultado:** é uma característica psicológica da atividade do jogo, está relacionada ao resultado produzido pelo jogador. Um jogador pode se sentir emocionalmente feliz ao alcançar a vitória e triste ao perder no jogo. Entretanto, isso não está relacionado apenas ao esforço do jogador, em jogos de sorte esse sentimento também ocorre na vitória ou derrota.
- **Consequências negociáveis:** O mesmo jogo (conjunto de regras) pode ser jogado com ou sem consequências na vida real.

Na visão de Juul (2005, p. 64), é possível valer-se desses seis requisitos propostos e transformar seja o que for em um jogo. Como exemplo, duas pessoas caminhando na rua podem transformá-la em uma corrida, basta para isso atribuir

valores aos resultado entre os jogadores e determinar qual é a melhor forma de chegar ao destino primeiro.

Apesar dos jogos de videogame seguirem o modelo padrão dos jogos, em grande parte, rompem com esse padrão convencional. O autor conceitua alguns exemplos que isso ocorre onde, embora os videogames serem baseados em regras tanto quanto outros jogos, eles modificam esse modelo padrão, pois é o computador que retém as regras. Isso garante aos videogames muita versatilidade, com isso permite-se regras muito mais complexas, libera o jogador de ter que controlá-las. Ademais, permite jogos em que o jogador desconheça as regras desde o seu início. Essa modificação é igualmente vista no conceito de resultado variável com o jogos do segmento RPG online. O jogador nunca atinge um resultado final, mas apenas temporário ao sair do jogo. Outro exemplo é a inclusão de códigos que modificam o jogo, o jogador em muitos casos, é livre para alterar algumas das regras básicas de um jogo (JUUL, 2005, p. 74 e 284).

Na literatura pela busca de uma melhor compreensão na definição de jogo relativo à indústria dos vídeo games, temos a definição Schuytema, como segue:

Um game é uma atividade lúdica composta por uma série de ações e decisões, limitada por regras e pelo universo do game, que resultam em sua condição final. As regras do universo do game são apresentadas por meios eletrônicos controlados por um programa digital. As regras e o universo do game existem para proporcionar uma estrutura e um contexto para as ações de um jogador. As regras também existem para criar situações interessantes com objetivo de desafiar e se contrapor ao jogador. As ações do jogador, suas decisões, escolhas e oportunidades, na verdade, sua jornada, tudo isso compõem a “alma do game”. A riqueza do contexto, o desafio, a emoção e a a diversão da jornada de um jogador, e não simplesmente a obtenção da condição final, é que determinam o sucesso do game (SCHUYTEMA, 2008, p. 7 apud LEMES, 2009, p.20).

No pensamento de Fullerton (2019, p. 31-32 e 57), um jogo é composto por elementos formais que formam a sua estrutura. Os elementos formais correspondem ao jogadores, objetivos, procedimentos, regras, recursos, conflitos, limites e resultado, são a essência dos jogos. Sem estes elementos os jogos deixam de ser jogos. No enfoque da autora independente da categoria de jogo, quão díspares possam ser, ambos compartilham de elementos em comum. Em seu experimento temos essa comparação da ação do jogador entre um jogo de cartas e a interação de um jogo

digital de tiro em primeira pessoa (FPS). Percebe-se que são dois universos bem distintos, apesar disso existe a possibilidade de identificar elementos comuns entre ambos. Nesse contexto a autora explana sobre os conceitos inerentes as duas experiências, como:

Quadro 7- Comparação Go Fish x Quake

		Go Fish	Quake
Aspectos do jogo		Jogo de cartas baseado em turnos. Execução do jogo através de baralho comum. Conjunto de regras de domínio público passado de geração em geração.	Jogo de tiro FPS, 3D em tempo real. Execução do jogo necessita do software comercial e um computador com as configurações mínimas Protegido por direitos autorais
Jogadores	Participante voluntário que participa e consome o entretenimento, aceita voluntariamente as regras e restrições de um jogo.	Grupo de ao menos três jogadores desafiando uns aos outros.	Jogador contra o sistema do jogo, para o modo de um jogador.
Objetivos	Metas estabelecidas pelo jogo	Baixar na mesa sempre que formar quatro cartas de mesmo número e naipes diferentes (<i>book</i>).	Permanecer-se vivo matando os inimigos através de tiros e completar o nível em que você está.
Procedimentos	O modo como os jogadores podem cumprir os objetivos estabelecidos pelo sistema do jogo As regras definem o que o(s) jogador(es) pode(m) executar ou não no jogo. . As regras são orientações arbitrárias às quais o jogador se submete. Esclarece o que acontece em várias situações que possa surgir	Distribuição de cartas, pedir a um jogador específico cartas de um determinado naipe As cartas são colocadas viradas para baixo em uma pilha de compra	O personagem pode andar, correr, atirar, pegar itens. A variedade de munição encontrada no jogo, atende a armamentos específicos. As balas são para as espingardas, os pregos para pistola de pregos e perfuradores, etc.
Recursos	Aplicação de valores a vários objetos do jogo. Esses objetos, tornam-se valiosos porque podem ajudar os jogadores a atingirem seu objetivo. Encontrar e gerenciar estes	Certos naipes são mais desejáveis para formar o <i>book</i>. Depende de uma troca direta de recursos (cartas).	Certas armas são mais poderosas, é fundamental o gerenciamento das munições.

	recursos é uma parte fundamental para muitos jogos.		O game designer distribui os recursos pela área do jogo
Conflito	O relacionamento entre o(s) objetivo(s) do jogador(es), as regras e os procedimentos que limitam e orientam o comportamento do jogador(es), cria um novo elemento distintivo dos jogos, o conflito. O conflito é o obstáculo que o jogador precisa transpor para ter um resultado favorável	Não é possível pedir aos demais jogadores da mesa ao mesmo tempo se eles possuem a carta do valor que você deseja. Ao invés disso, o jogador tem que se arriscar e escolher somente um jogador por vez.	Para completar a missão é necessário passar por vários obstáculos, inimigos e perigos. Simplemente não é possível passar pelo labirinto sem passar por esses desafios.
Limites	As regras e objetivos que estão impulsionando os jogador(es) se aplicam apenas dentro do jogo e não na "vida real". É a ideia do jogo estar separado da realidade cotidiana, caindo no "Círculo Mágico" descrito por Johan Huizinga. O Círculo Mágico é o mundo temporário onde as regras do jogo se aplicam, o jogador concorda com as regras impostas do jogo, ao invés das regras do mundo real.	O limite é mais conceitual do que físico. Os jogadores não estão limitados fisicamente por alguma das regras, mas por um acordo social entre os jogadores que não vão sair do jogo com algumas cartas ou adicionar mais cartas no baralho	A arquitetura do espaço 3D forma uma fronteira virtual, os jogadores são impedidos de mover seus personagens fora desse limite
Resultado	O resultado de um jogo difere do objetivo, os jogadores podem atingir o mesmo objetivo. Entretanto, outros fatores dentro do sistema podem determinar o resultado de quem é o vencedor ou perdedor. A incerteza do resultado é um aspecto importante no processo do jogo, é o motivador chave para os jogadores.	O jogador que conseguir o máximo de <i>books</i> até o final do jogo é considerado vencedor	O jogador pode ganhar (permanecer vivo) ou perder (ser morto)

Fonte: FULLERTON (2019, p 32-38).

Com base na linha teórica apresentada pelos pesquisadores, é um ponto de partida o entendimento na definição do conceito de um jogo digital, apesar da dificuldade de se estabelecer um consenso quanto a sua definição. Os modelos apresentados servem como premissa de como os jogos são construídos. De modo geral, com os elementos formais que constituem o jogo denotados pelos autores supracitados conseguimos uma melhor compreensão a esse respeito.

Isso posto, parte-se para a busca ao entendimento no que Schuytema (2008) caracteriza como a “alma do game”, essa interação com o jogo promovida pelo jogador carrega elementos que compõem o *game design* no desenvolvimento de um jogo digital, como veremos no tópico a seguir.

4.5.2 Sobre Tecnologia Mecânica, Dinâmica, Estética e Narrativa de jogos

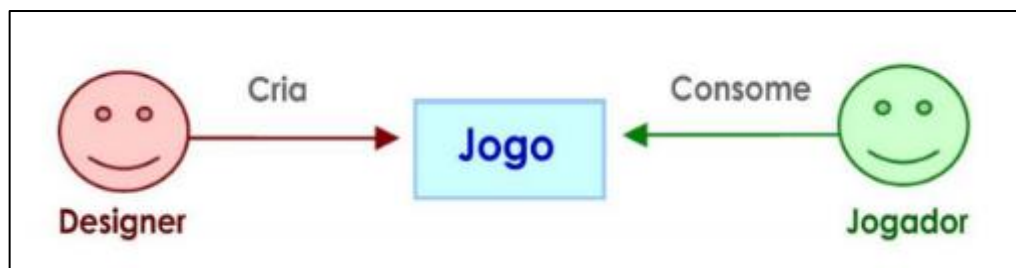
Pretende-se fazer uma revisão da literatura de alguns modelos de *game design* apresentados como parte da percepção dos elementos fundamentais que constituem a criação de um jogo digital.

4.5.2.1 MDA (*Mechanics, Dynamics and Aesthetics*)

Os autores Hunicke, LeBlanc, Zubek (2004), apresentam o modelo MDA (sigla para Mecânicas, Dinâmicas e Estética) e abordam sobre esse novo conceito, onde temos uma nova perspectiva na análise e projeto de jogos digitais. Seu objetivo baseia-se em estabelecer uma conexão entre as equipes de produção, com o intuito de facilitar o desenvolvimento do jogo e um melhorar os processos iterativos.

Ainda, segundo os autores, os jogos como a maioria dos bens de consumo, são comprados, usados e eventualmente descartados. Após todas as etapas do processo de concepção e desenvolvimento e o resultado final é o produto gerado, entretanto, ao chegar ao mercado consumidor temos um imprevisibilidade do que irá ocorrer em relação ao seu consumo.

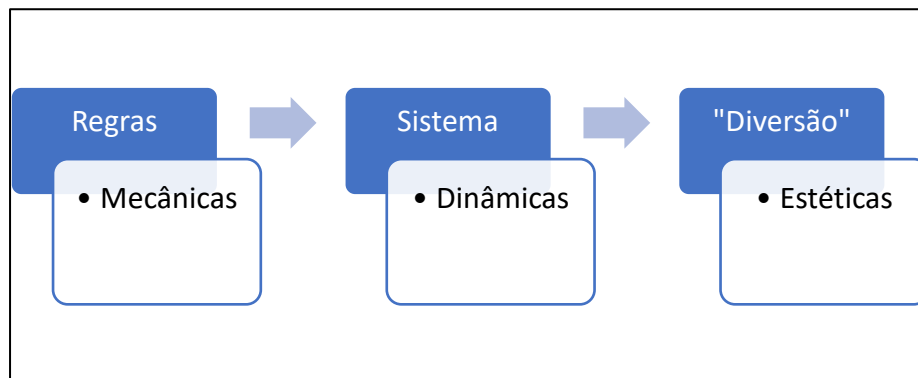
Figura 5 – Produção e consumo do jogo



Fonte: Traduzido de Hunicke, LeBlanc & Zubek (2004, p. 1)

Assim, na perspectiva da metodologia MDA a unidade jogo é decomposta em três componentes específicos, Regras, Sistema e Diversão e faz sua contraposição com suas partes no *design*, conforme figura

Figura 2 – Componentes do MDA



Fonte: Traduzido e adaptado de Hunicke, LeBlanc & Zubek (2004, p. 2)

Hunicke, LeBlanc, Zubek (2004), detalham esta divisão dos componentes, sendo:

[...] Mecânica: Componentes específicos do jogo, no nível de representação de dados e algoritmos.

Dinâmica: Descreve o componente da mecânica quando ela é executada pelas ações do jogador e cada um dos resultados ao longo do tempo.

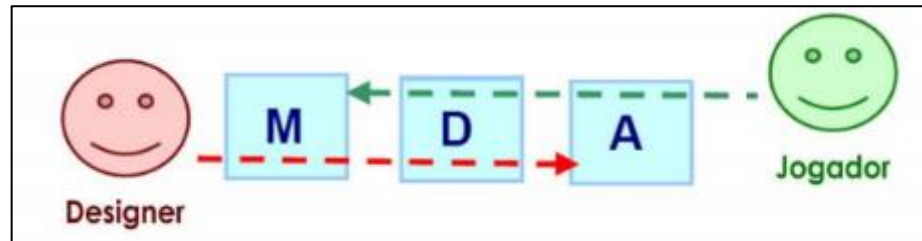
Estética: Descreve as respostas emocionais desejáveis evocadas no jogador, quando ele interage com o sistema de jogo." (Hunicke; LeBlanc; Zubek, 2004, p.2, tradução nossa)

No pensamento dos autores temos que considerar os jogos como artefatos projetados, isso ajuda a enquadrá-los como sistemas que constroem o comportamento por meio da interação. Podemos entender que cada componente do MDA com uma lente, que embora separada estão interligadas para o desenvolvimento do jogo. (Hunicke; LeBlanc; Zubek, 2004)

Ao se trabalhar com jogos é necessário observarmos as perspectivas do *designer* e também a perspectiva do jogador. Nesse contexto no olhar do *designer* a mecânica da origem ao comportamento dinâmico do sistema, e que leva a experiência estética. E pela perspectiva do jogador a primeira sensação é a estética, passando

pela dinâmica observável e, eventualmente na mecânica operativa. (Hunicke; LeBlanc; Zubek, 2004)

Figura 6 – Designer e jogador perspectivas diferentes



Fonte: Traduzido de Hunicke, LeBlanc & Zubek (2004, p. 2)

Na visão dos autores, na perspectiva do *designer* em primeiro momento ele está preocupado em estabelecer os conjuntos de regras e códigos da mecânica, na dinâmica é o processo computacional. O último processo a estética o *designer* avaliar, como por exemplo o teste de um protótipo.

Com relação ao ponto de vista do jogador, no primeiro contato com o jogo é a observação da estética como elementos gráficos, trilhas e efeitos sonoros do jogo. Na dinâmica a preocupação do jogador é voltada nas interações do jogador com o jogo, as ações exercidas pelas mecânicas implementadas pelo *game designer*. O jogador com o decorrer dessa interação com o sistema do jogo, vai compreendendo as mecânicas⁷ à medida que joga.

Na proposta Hunicke, LeBlanc, Zubek (2004), estrutura em oito elementos estéticos que caracterizam as respectivas experiências de seus jogadores. Essa é uma forma de entender o relacionamento do perfil do jogador com o jogo. Ao utilizar

⁷ Uma das tarefas do *game designer* é transformar as regras gerais do jogo em um modelo simbólico e matemático, este modelo denomina-se mecânica central. Os programadores implementam os algoritmos gerados da mecânica central. Em um videogame a mecânica central é escondida dos jogadores, a experimentação é feita através do jogo. Se o jogadores jogarem repetidamente, eventualmente terão consciência da natureza mecanicista do jogo e aprenderão otimizar o jogo para vencer (ADAMS, 2010, p.35-36).

esses elementos de estética, serve como orientação para definir modelos para a jogabilidade, ajuda a descrever a dinâmica e a mecânica do jogo.

Figura 3 - Oito elementos de estética

Elemento	Descrição	Jogo exemplo
Sensação	Jogo como prazer nos sentido	 <i>Fonte: Ori and the Will of the Wisps - Moon Studios (2020)</i>
Fantasia	Jogo como faz de conta	 <i>Fonte: The legend of Zelda: Breath of the Wild Nintendo (2020)</i>
Narrativa	Jogo como drama	

		Fonte: Last of Us Part 2 – Naughty Dog IGN Portugal (2020)
Desafio	Jogo como pista de obstáculo	 Fonte: Celeste – Matt Makes Games Steam (2020)
Companheirismo	Jogo como contexto social	 Fonte: World of Warcraft Blizzard (2020)
Descoberta	Jogo como território não mapeado	 Fonte: Death Stranding - Kojima Productions (2020)

Expressão	Jogo como autodescoberta	 <i>Fonte: Sims 4 - EA (2020)</i>
Submissão	Jogo como passatempo	 <i>Fonte: Bejeweled – EA (2020)</i>

Fonte Traduzido e adaptado de Hunicke, LeBlanc, Zubek (2004, p.2)

A metodologia MDA auxilia o designer a abstrair e conceituar o comportamento da dinâmica dos sistemas do jogo, entendendo esse comportamento ajuda a desenvolver técnicas de design, melhorias iterativas e entender como certas decisões impactam a experimentação do usuário final. Dessa forma podemos decompor essa experiência e fazer uma análise iterativa, quantitativa e qualitativa desse resultado e aperfeiçoar o desenvolvimento do jogo.

4.5.2.2 Tétrade elementar

No âmbito de estudo de jogos Schell (2020, p.53), sugere o uso do téttrade elementar para concepção dos jogos. O autor faz uma síntese desses quatro elementos, como segue:

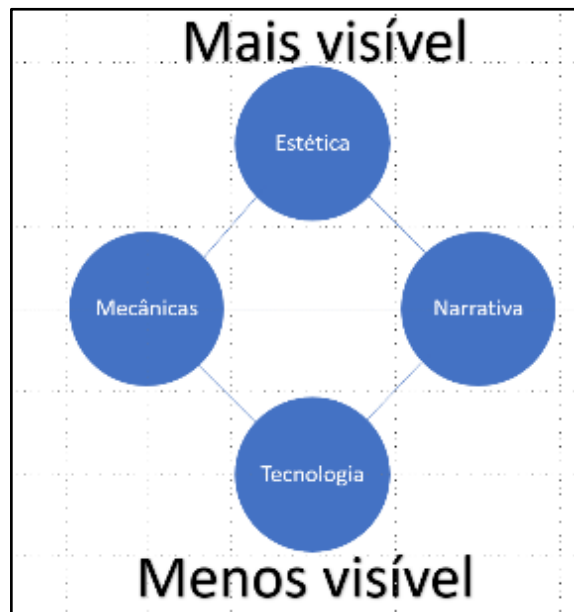
Quadro 8 – Especificações dos quatro elementos do téttrade de Schell

Mecânica	Esses são os procedimentos e as regras do jogo. A mecânica descreve o objetivo do jogo, como os jogadores podem ou não alcançá-lo e o que acontece quando tentam. A mecânica é que define o jogo em si. Ao escolher o conjunto de funções inerentes a mecânica, indubitavelmente será necessário escolher a tecnologia que deverá suportá-la, a estética que enfatiza o aspecto do jogo e a narrativa que permite à mecânica do jogo fazer sentido para os jogadores.
Narrativa	Sequência de eventos que se desdobram no jogo. Ela pode ser contada linearmente e previamente determinada, ou ramificada e emergente. A mecânica ajuda a contar a história e a deixa aparecer. A estética ajuda a reforçar a ideia da história e faz uso da tecnologia mais adequada para essa narrativa. A narrativa se relaciona singularmente com a mecânica
Estética	A estética tem a ver com a aparência, sons e sensações inseridas no jogo. É um dos aspectos mais importantes, pois se relaciona direto com a experiência do jogador. O uso da tecnologia apropriada permitirá que estética surja, mas também que a amplifique e a reforce. A escolha da mecânica faz com que o jogador se sinta imersivo naquele mundo definido pela estética. A narrativa ajuda a emergir a estética no ritmo certo e cause maior impacto.
Tecnologia	A tecnologia diz respeito a quaisquer materiais e interações que tornem o jogo possível. A tecnologia é o meio que a estética acontece, em que a mecânica ocorre e onde a narrativa será contada.

Fonte: Schell (2020, p. 53-54)

Na visão de Schell (2020, p.55), todos os elementos apresentados são de igual teor de importância. Os quatro elementos compõem o universo do jogador e deve ser considerado pelo *game designer*. O autor orienta que os quatro elementos sejam o modelo projetual na condução no desenvolvimento do jogo e que contemple à todos da equipe da produção do jogo. Desse modo, cada parte desempenha a mesma relevância sobre a experiência jogador. Na figura abaixo é apresentado o modelo proposto por Schell, sendo:

Figura 4 – Tétrade Elementar de Schell



Fonte: Traduzido e adaptado de Schell (2020, p.54)

4.5.3 Interface e Experiência do Usuário

Nessa seção pretende-se abordar os conceitos de UX e UI voltados para o desenvolvimento dos jogos digitais.

4.5.4 Teoria do Flow

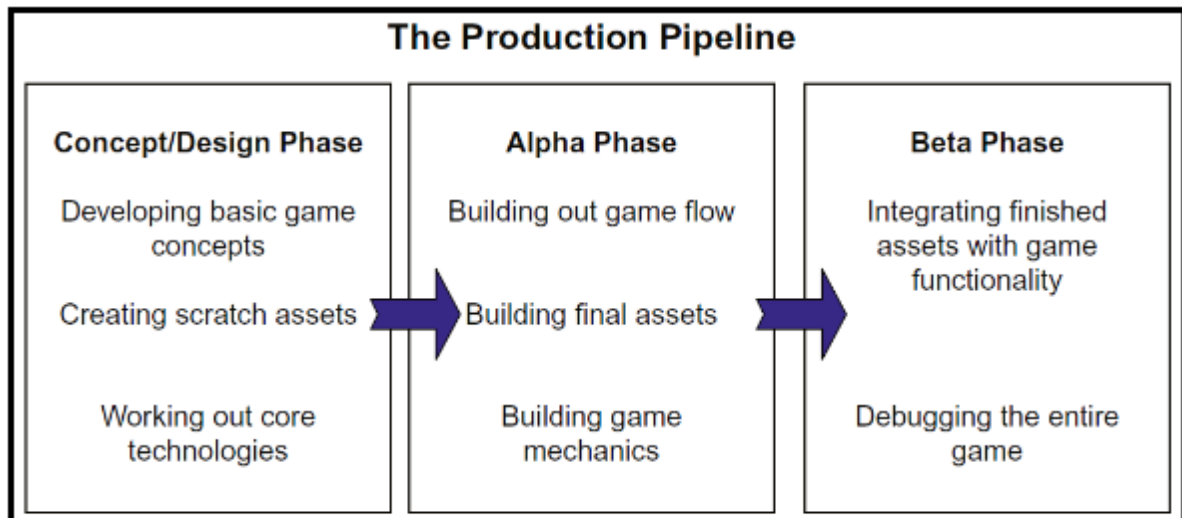
Nessa seção pretende-se abordar a Teoria do Flow de Mihaly Csikszentmihálvi para jogos digitais e a investigação dessa experiência jogador quando atinge o estado de Flow e seu processo cognitivo durante a atividade para um determinado jogo. E nesse aspecto, qual é a incumbência do Game Designer na criação destas experiências.

4.6 Fundamentos para desenvolvimento e produção dos jogos digitais

4.6.1 Projeto de um jogo digital

Nessa seção pretende-se discutir a definição de projeto de jogo digital e análise propriamente dita.

4.6.2 Etapas de desenvolvimento do jogo digital



Production phases and key activities.

4.6.3 Perfil do Jogador

Nessa seção pretende-se delinear a respeito dos quatros perfis de jogadores, segundo Richard Bartle (Conquistadores, Exploradores, Combativos, Socializadores) para arquitetar o design de um jogo

4.6.4 Público-alvo

Nessa seção pretende-se delinear sobre a qual público se destina o desenvolvimento do jogo (Serious Game, Adverggames, Simulação, Casual)

4.6.5 Plataformas de desenvolvimento do jogo digital

Nessa seção pretende-se abordar uma visão geral das plataformas de desenvolvimento.

4.6.6 Métodos de prototipagem do jogo digital

Nessa seção pretende-se delinear sobre o processo de prototipagem (Machinations, baixa fidelidade, alta fidelidade, *mockup*)

Como normalmente é o fluxo de trabalho de programação do jogo?

A primeira etapa é a pré-produção. Nós prototipamos áreas de jogo desconhecidas ou arriscadas, escrevemos documentos que dão a visão central do design do jogo e tomamos decisões técnicas importantes, e avaliamos o escopo e o orçamento do projeto. A pré-produção está essencialmente focada na redução de riscos. Se você não tem certeza sobre sua jogabilidade, então gasta seu tempo de pré-produção criando certifique-se de saber que sua jogabilidade está certa. No momento em que você entrar em produção, suas incógnitas devem ser reduzidas para que você possa realmente começar a trabalhar e fazer ajustes na aeronave conforme os prazos mudam. E se você é um programador líder, é seu trabalho comunicar o que tudo isso significa. Se alguém diz: “Queremos mudar isso para aquilo”, você deve dizer: “Bem, isso significa X em termos de tempo ou Y em termos de risco”. Como programador, independentemente da posição, você deve fornecer feedback para que as pessoas possam tomar decisões informadas colocar na produção ou na parte de desenvolvimento da matriz cap 6 (RUGGILL, MCALLISTER, NICHOLS, & KAUFMAN, 2017)p20

4.6.7 Engenharia de Software e Hardware

Nessa seção pretende-se abordar uma visão geral sobre ES (requisitos funcionais e não funcionais)

5. QUALITY ASSURANCE (QA) A PLAYABILITY TESTING

Na busca da excelência no desenvolvimento do jogo, as empresas desenvolvedoras apoiam-se nas atribuições da equipe de Garantia de Qualidade (QA) para atingir o resultado planejado e o padrão de qualidade estabelecido pela empresa, bem como, atingir expectativa inerente ao seu público-alvo de consumo, neste caso o jogador. Assim sendo, quais seriam estas atribuições do time de QA e qual a sua importância no mercado de desenvolvimento de games? Qual seria a influência de uma equipe de QA para a melhoria na produção de um jogo digital? Para responder essas questões, primeiramente temos que definir o que é garantia de qualidade, abordar as etapas da área de QA e qual a sua correlação com as demais áreas de desenvolvimento do projeto de um jogo digital.

5.1 Definição de *Quality Assurance* (QA) nos jogos digitais e sua estrutura

A Garantia de Qualidade (QA) é o processo de teste e ajuste que ocorre antes do lançamento do jogo no mercado, para que o mesmo tenha toda a parte conceitual do projeto que foi estabelecida e concebida pelos seus desenvolvedores. A equipe de QA tem a atribuição de avaliar os elementos estéticos, técnicos e ou interativos de um jogo para possíveis melhorias durante os estágios de desenvolvimento, e dependendo, mesmo após o seu lançamento. (RUGGILL, MCALLISTER, NICHOLS & KAUFMAN, 2017, p. 213)

Em consonância a esse pensamento Irish (2005, p.259), afirma que o QA e o teste de jogabilidade são duas etapas muito importantes antes do lançamento do produto no mercado. Esse processo garante que o produto funcione conforme projetado, que forneça funcionalidade, usabilidade e diversão suficientes para ser atraente para os críticos e usuários finais. O ideal é que essas etapas de testes não sejam ignoradas ou apressadas, há o comprometimento do projeto com o risco do lançamento do produto com falhas de funcionalidade, jogabilidade ou conflitos de software e hardware.

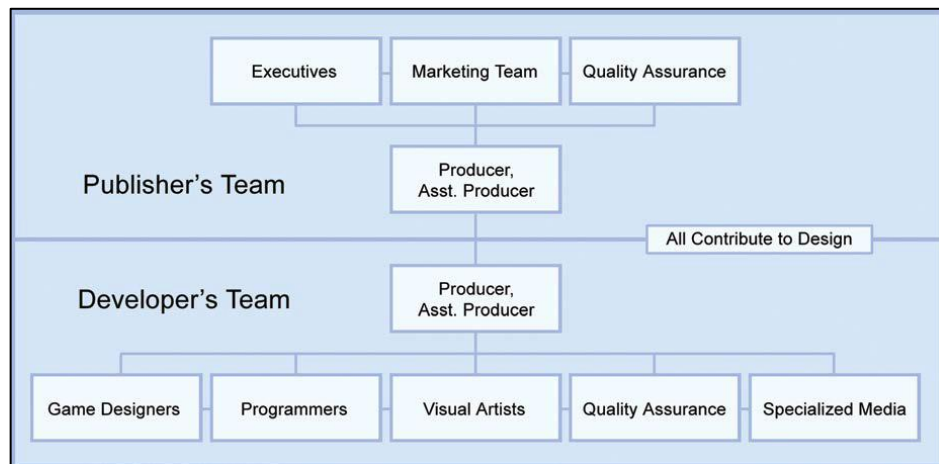
Na visão de Mencher (2003, p. 28-29) e Novak (2017, p. 325), é fundamental estabelecer padrões e procedimentos para o desenvolvimento do jogo, assim concebe-se uma estrutura e a partir desta temos a evolução do software. Assim a função do QA é do monitoramento do processo, avaliação do produto e auditoria para satisfazer aos requisitos do jogo. Como por exemplo, a equipe de QA se preocupa com a documentação e os padrões estabelecido pelo design, que especificam a forma e o conteúdo do jogo. Verifica o padrão do código escrito pelo programadores, compatibilidade de hardware e software empregados, e claro o teste de funcionalidade do jogo propriamente dito.

Nessa perspectiva, Ruggill *et al.* (2017, p. 213) descreve que esse conjunto de padrões empregados na busca de falhas e avaliação do projeto podem ser definidos pelos desenvolvedores e ter a participação da *Publisher*, da plataforma ou ambiente host (Microsoft Xbox Live, Steam) em desenvolvimento, por organizações de classificação etária ou políticas e leis públicas. Esse padrões podem incorrer, por exemplo, em questões de estética (uma editora de audiovisuais), tecnológico (estabilidade operacional e conectividade de uma rede ou serviço) ou moral (conteúdo apropriado para uma determinada idade).

Caso o desenvolvedor opte em trabalhar com alguma *Publisher* ou *Third-party*, na maioria das vezes ambas contemplam no seu quadro de funcionários uma equipe destinada a garantia de qualidade e testes (MENCHER, 2003). A equipe de QA da *Publisher* tem uma similaridade com o estafe de QA de desenvolvimento. Todavia, há duas exceções, primeiro eles provavelmente não estão familiarizados com jogo, pois eles não trabalham lado a lado com a equipe de produção de desenvolvimento. E segundo, o objetivo geral é deliberar e dar o aceite da versão compilada para análise, atendendo os requisitos técnicos e a rigorosidade da *Publisher* (FULLERTON, 2019, p. 407). O foco central da equipe de QA de desenvolvimento é encontrar o máximo de erros possíveis no software, enquanto a preocupação da equipe de QA da *Publisher* é a validação do jogo para que atenda as suas expectativas e cumprimento dos marcos de entrega (MENCHER, 2003).

A Figura 8, ilustra a estrutura básica organizacional das equipes de Desenvolvedores e as *Publisher's* na indústria dos jogos.

Figura 7 – Setor de QA estrutura organizacional Desenvolvedores e Publisher's



Fonte: Fullerton (2019, p. 392)

Na produção de jogos de alto nível, considerados *Triple A*, temos um maior investimento financeiro para realização do jogo. Um título considerado *Triple A*, gera um grande expectativa de estar entre os melhores lançamentos do ano e com isso espera-se uma alta qualidade na produção do jogo. Para Berthke (2003, p.52), esse tipo de jogo recebe da *Publisher* uma quantidade considerável de testadores de QA da sua equipe interna. Essa equipe segue as diretrizes estabelecidas pela *Publisher*, assim há uma maior eficiência em relatar para os desenvolvedores os possíveis problemas encontrados na análise do projeto. Dessa forma temos uma maior qualidade nos relatórios e evita-se algum atraso no lançamento do jogo no mercado.

Para Chandler (2012, p.34), o testador de QA é a parte primordial do processo de desenvolvimento de jogos e lidam com a exploração na busca de falhas presentes no jogo. Segundo Mencher (2003, p.29), há uma subdivisão da equipe de QA em grupos específicos que focam nos padrões do sistema, na compatibilidade, na jogabilidade, ou no fator diversão daquele jogo a ser testado.

Conforme apresentado por Irish (2005, p. 261), Chandler (2012, p.34-35) e Novak (2017, p.325-327) a composição das áreas de responsabilidade de QA no processo de teste dos jogos, se configura da seguinte forma:

- **Gerente de QA** É responsável por múltiplos projeto de games, suas responsabilidades envolvem a administração e gerenciamento dos planos de testes. Os líderes de QA ficam subordinados ao gerente de testes, o foco principal é garantir que os testes serão concluídos dentro do prazo.
- **Líder QA** A responsabilidade do líder QA se concentra na criação do plano de teste e supervisão da equipe. Ele trabalha juntamente com o produtor e outros líderes de projeto, com a finalidade de estabelecer o cronograma para a execução e a viabilidade dos testes. Assim ele garante que o jogo seja totalmente testado, mas para isso ele deve estar em sintonia com a equipe de desenvolvimento e saber exatamente como cada parte do jogo funciona. Com essa compreensão, enviará a demanda do plano de testes para a área específica para averiguação.
- **Testador de QA** Mediante ao plano de testes com base na documentação do design, estes são responsáveis pela verificação da funcionalidade do jogo, anotar as falhas encontradas durante o teste e gerar um relatório destes eventos ocorridos. Faz parte também das suas atribuições testar novos recursos, protótipos e conferem se o jogo atende todos os requisitos técnicos do fabricante do console. Eles são instruídos a jogar por todos os caminhos possíveis e de todas as formas, com o objetivo de que o jogo venha a falhar em algum momento. Na condição de testadores a maior parte do tempo de trabalho é jogar, logo conseguem ter opiniões embasadas sobre o fator diversão daquele jogo. temos o aspecto positivo como contribuição para a equipe de desenvolvimento.
- **Testador de Compatibilidade e Formato** Sendo o testador de compatibilidade para games de computador e de formato games para console. Esse teste tem a missão de determinar se um game é compatível com diferentes plataformas, funcionando bem em todas elas. Visa verificar alguma inconformidade entre o *software* desenvolvido e o

os recursos específicos de *hardware* que o jogo promete. Caso ocorra a incompatibilidade, é necessário o fornecimento do *hardware* para correção da falha de compatibilidade pelos programadores. Geralmente essa etapa é conduzida por empresas externas sob supervisão do líder QA. Essa fase também tem que estar inserida no cronograma do plano de testes.

- **Testador de Jogabilidade, Usabilidade e Beta** Diferente dos testadores profissionais da equipe de QA, a função desses testadores é jogar e fornecer uma avaliação sobre o jogo, em geral esse testador pode ser qualquer pessoa. Esse tipo de teste promove a exposição do jogo em desenvolvimento ao público-alvo, procura analisar novas estratégias de mercado, novas mecânica ou teste de usabilidade. Normalmente esses testes ocorrem do meio para as etapas finais do projeto, sendo de grande valia e requer um testador dedicado para apurar o trabalho à medida que é implementado.
- **Testador Focal** Apesar de não fazer parte dos procedimento de avaliação de QA, vale a informação como conhecimento. Quando se pretende fazer o teste diretamente com o público-alvo, são recrutados localmente, e o intuito é saber quais são os aspectos do jogo são mais atraentes. O gerenciamento desses testes é conduzido pelo setor de marketing, onde são feitas grupos de discussão e feito o levantamento das opiniões do grupo sobre um novo produto ou serviço.

Garantia de qualidade interna e externa

Embora a tecnologia tenha avançado, criadores tenham inovado em muitos novos tipos de jogabilidade, em sua forma mais pura o processo de produção de jogos, a centralidade e importância da Garantia da Qualidade para esse processo permanece inalterada.

5.2 Esse universo dos testes, etapas do processo

Depois definirmos qual é a função de *Quality Assurance* na cadeia produtiva no desenvolvimento do jogo, primordialmente temos que conhecer as diretrizes do ciclo de trabalho do grupo de QA. Outrossim Adams (2003, p. 66), diz que o teste se inicia a partir do momento que exista uma estrutura em grande parte jogável, sendo absolutamente essencial para o desenvolvimento do jogo.

Mas por que os testes são importantes? Schultz, Bryant & Langdell (2005, p. 39), elencam uma série de fatores que vão desde ao problemas com o software, a complexidade do seu desenvolvimento, o investimento financeiro e tempo gerado no processo, as expectativas geradas pelos usuários, avaliação da crítica especializada, prazos de entrega, até o nível de diversão propiciado pelo jogo. Um fator que pode ser de ajuda é identificar mecanismos e padrões que não deram certos em outros jogos, e estabelecer um relação com os tipos de problemas que você deve procurar e antecipar isso no seu teste.

Para Adams (2003, p.66), ao efetuar o teste, este não envolve criatividade, somente horas a fio de experimentação de diferentes recursos em diferentes combinações para que todos funcionem. O teste não envolve apenas jogar, tem que ter um talento especial para procurar problemas durante o jogo, mas também é preciso documentar e relatar as falhas. Essas tarefas são feitas repetidamente até o lançamento do jogo no mercado.

“Muitas partes do jogo precisam ser testadas em diversos pontos no ciclo de vida do jogo” ⁸(SCHULTZ, BRYANT & LANGDELL, 2005, p. 5, tradução nossa).

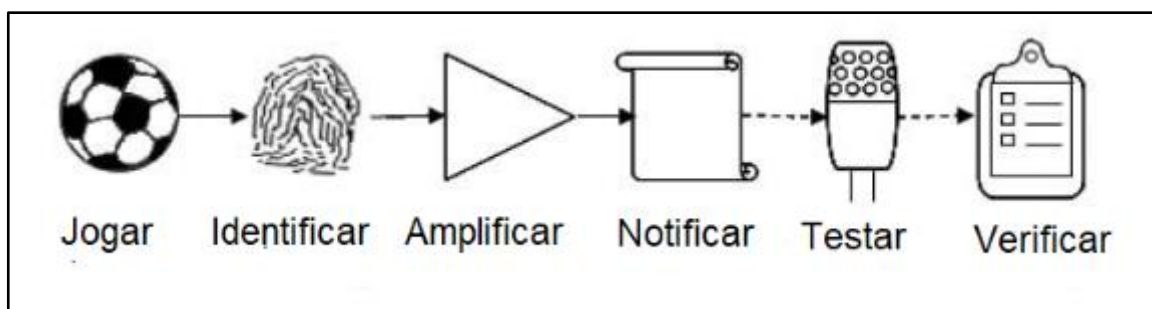
Assim, Scultz, Bryant & Landell (2005, p. 19-36), acreditam que o plano de teste não é somente saber jogar, há inúmeras etapas que devem ser seguidas que mesclam entre saber o que procurar e como procurar, qual área do jogo deve ser explorada e

⁸ “Many parts of the game need to be tested at various points in the game’s life cycle”

seguir a lista de verificação do que deve ser testado naquela etapa. A lista de verificação se concentra na realização de um conjunto de ações e comportamentos naquele determinado ponto de teste, observando-se o cumprimento dessas determinadas ações, e se obtiveram êxito conforme o previsto durante o teste do jogo.

E ainda, os autores mencionam que o outro propósito do teste além de encontrar e identificar defeitos na codificação ou no design do jogo é testar a sua funcionalidade. A segunda função do teste é verificar quais partes do jogo estão funcionando corretamente ou estão com problemas, se estiver tudo certo, aquele trecho é validado pela equipe de QA. Caso contrário é considerado como falha e tem que ser revisto pelos desenvolvedores. O importante é conseguir ao máximo detalhar em qual local e circunstância ocorreu a falha, isso faz com que os desenvolvedores possam agir rapidamente na solução e correção do problema. Se esse detalhamento não ficar bem claro, o reparo fica mais difícil de ser realizado esse defeito provavelmente ficará em segundo plano. A notificação das falhas para a equipe de desenvolvimento, geralmente é registrada através de ferramenta de *software* com este propósito, onde essa faz o gerenciamento e rastreamento das falhas inseridas no programa. Após a correção há as etapas de teste da nova versão concebida pelos desenvolvedores, onde o testador segue os passos descritos anteriormente no momento da falha, na tentativa que ocorra novamente, assim é verificado se a falha foi corrigida. Cabe a observação que nem todo testador notará os mesmos defeitos ao testar o mesmo jogo. Da mesma forma, nem todo testador executará o mesmo teste da mesma maneira. De modo resumido temos as etapas do ciclo estabelecidas para teste de um jogo.

Figura 8 – Atividades de um game teste



Fonte: SCHULTZ, BRYANT & LANGDELL (2005, p. 19, tradução nossa)

Na visão de Irish (2005, p.260 e 264), os procedimentos de equipe de QA incluem o estabelecimento de um plano de teste e métodos para relatar, corrigir, rastrear, verificar e encerrar os *bugs* (ver seção 5.2.1 A definição de *bug* e suas categorias). Ao minimizar os *bugs* durante o processo de desenvolvimento do jogo, há mais tempo disponível para aperfeiçoar sua jogabilidade. O planejamento de testes é de tal importância que dependendo do grau de severidade em questão de *software* e *hardware* pode afetar e muito, a experiência do jogador em um teste de versão.

Segundo Chandler (2020, p. 211), tratando-se de um teste voltado para a experiência do usuário (UX) e o jogo encontre-se no estágio de desenvolvimento, seja ele um protótipo ou alguma versão jogável, certifique-se que o jogo não tenha grandes *bugs*, recursos ausentes ou problemas técnicos que afetará a experiência do usuário. O ideal é que essa versão esteja apta e todas as funcionalidades estejam de acordo, isso garante que os recursos primordiais sejam testados o mais breve possível. Assim, pode-se ter um ciclo maior de teste de quaisquer iterações no recurso. Vale consignar que o teste voltado para o usuário é diferente do teste realizado pela equipe de QA, o propósito do teste de UX é o comentário do usuário em relação aos aspectos de envolvimento, diversão e satisfação ao jogar. Contudo, o teste de UX está implicitamente interligado com os testes realizados pela a equipe de QA, que está a procura de falhas, *bugs* e dos aspectos funcionais do jogo. Caso algumas dessas complicações ocorram durante o teste do usuário, há a quebra desses elementos da experiência do jogador que se pretende avaliar.

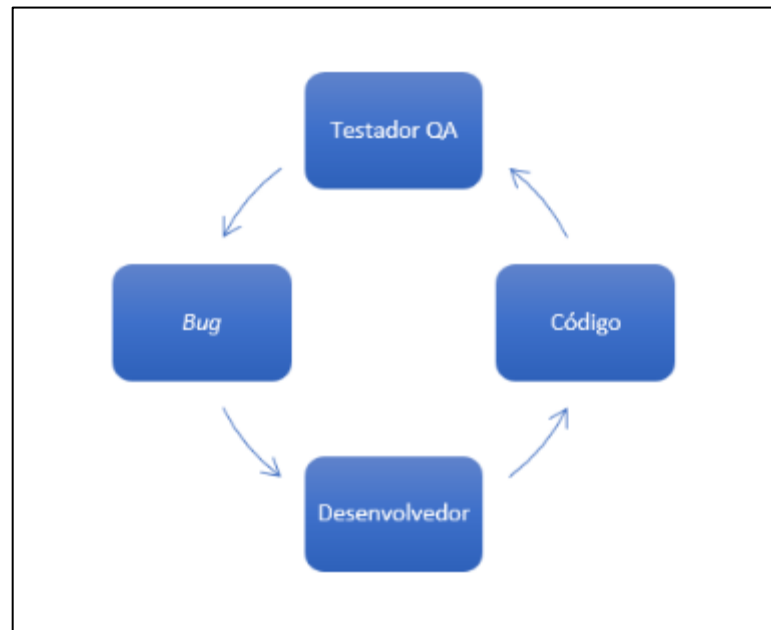
Adams (2003, p.162-163), diz que via de regra um terço da parte do desenvolvimento de um jogo é dedicado ao teste, apesar das tentativas de se obter uma tecnologia de linguagem e engenharia de software à prova de erros. E ressalta que testar videogames, parte-se de um plano preciso que estabelece quais são os recursos a serem testados. O testador tem sua interpretação a respeito do processo de desenvolvimento do jogo, pois ele relaciona-se com o restante da equipe de produção a cada etapa de aprimoramento do jogo.

Como abordado anteriormente o departamento de QA elabora um plano de teste e para Chandler (2020, p.250-251), o intuito do plano é a verificação de todas as

áreas do jogo. Inicialmente é criado o plano de teste com referência à documentação fornecida pelo *Game Design*, a equipe também trabalhará com a equipe de desenvolvimento para manter esse plano devidamente atualizado, caso haja alguma alteração no desenvolvimento do jogo. O plano de teste pode conter centenas de páginas dependendo do escopo, no formato de uma lista de verificação do que foi aprovado ou reprovado durante o teste. Por exemplo, elabora-se uma lista de verificação de personagens jogáveis, o testador faz o procedimento habitual de entrar no jogo e ir na tela de seleção de personagens e confirmar quais personagens estão habilitados para o jogo. No segundo passo, há a verificação se após a seleção dos personagens é possível jogar com eles na fase do jogo. Assim há avaliação dos recursos de funcionalidade do jogo dos botões de interface do usuário (UI) para a seleção dos personagens estão atuantes em seu pleno funcionamento, ou caso contrário, apresenta falha ao fazer esse tipo de ação. Nesse caso o testador de QA irá reportar a falha e será necessário ser revisto, analisado pela área de desenvolvimento para sanar o problema.

Schultz, Bryant & Landell (2005, p. 155-156), considera que o processo do teste é um ciclo de *feedback* entre o testador e o desenvolvedor. O testador planeja e executa os testes, e ao encontrar o *bug* relata para o desenvolvedor, no qual este corrige e faz a compilação de uma nova *build*. O desenvolvedor repassa essa nova versão para a equipe de QA testar e verificar se o erro foi resolvido, do contrário, inicia-se novamente este ciclo até a resolução do problema.

Ciclo de feedback



Fonte: SCHULTZ, BRYANT & LANGDELL (2005, p. 156, tradução nossa)

Os autores apresentam o método básico em seis etapas para o teste do jogo, sendo:

- **Projeto e planejamento do teste:** Esse processo deve ser revisado a cada nova versão da *build*. Verifica-se quais são as mudanças ocorridas entre as versões, se há novas especificações, existe novos recursos adicionados ou retirados, alguma nova configuração que o jogo deve suportar. Deve-se garantir que não seja acrescentado nenhum novo problema durante o processo caso as alterações do escopo já foram atualizadas.
- **Preparação para o teste:** Código, testes, documentos e o ambiente de teste são atualizados por seus respectivos proprietários e feito o alinhamento com as equipes. A equipe de desenvolvimento assinalou a correção do *bug* no banco de dados, a equipe de QA pode começar a executar os testes de verificação da correção do *bug*.
- **Faça o teste:** Execute o testes dessa nova compilação. Caso encontre algum *bug*, faça o relatório da forma mais detalhada possível, isso facilita a sua utilização quando chegar aos desenvolvedores.

- **Relate os resultados:** Após a conclusão dos testes, registre e aponte os defeitos encontrados.
- **Repare o *bug*:** A equipe de teste discute a respeito do *bug* encontrado com a equipe de desenvolvimento e auxilia na reprodução do *bug*, caso necessário. A equipe de desenvolvimento depura o código para encontrar o *bug* e repará-lo.
- **Retorne a primeira etapa e teste novamente:** Após a correção do *bug* e com possibilidade de ser adicionado novos recursos conclui-se a *build*, e com essa nova *build* liberada para a equipe de teste, inicia-se um novo ciclo com possíveis novos *bugs* e novos resultados de testes, temos um nova compilação.

Os autores ainda salientam a aplicabilidade desse método nos mais diversos tipos de testes de garantia de qualidade na cadeia do processo de produção do jogo. Pode-se aplicar a todo jogo, em uma fase em desenvolvimento, em algum módulo individual ou recurso em construção.

Para Syed (2019, p.20), existem dois tipos de caso teste empregados conhecidos como teste positivo e teste negativo. O primeiro teste refere-se aos testes realizados que garantem que tudo funcione como deveria, como por exemplo, abertura de portas, acionamento de botões, etc., faz parte do plano de teste de funcionalidade da equipe de QA. No segundo caso, o teste negativo avalia como o jogo lida com algum comportamento inesperado ou inválido durante o jogo, como pressionar todos os botões do controle ao mesmo tempo. O intuito é provocar o que é chamado de “quebrar” o jogo (*break the game*), essa tarefa faz parte das instruções de um testador de QA.

O Assistente Líder de QA da empresa Sony Online Entertainment faz uma reflexão sobre a importância dos testes do setor de garantia de qualidade, como segue:

A garantia de qualidade e os testes andam de mãos dadas com o processo maior de desenvolvimento de jogos. Você não pode ter um sem o outro. A garantia de qualidade é muito importante para lançar um jogo ou expansão que seja divertido, sem *bugs* e desafiador, tudo ao mesmo tempo. A equipe de QA não cria o jogo, mas pode determinar se a sensação ou vibração de

um sistema específico é a pretendida do lado do desenvolvedor, bem como detectar e relatar *bugs* encontrados. [...] Floyd Billings (LEVY & NOVAK, 2010, p. 57, tradução nossa).

Foram apresentados como se constitui o planejamento de teste e evidenciado os principais pontos referente a esse tipo de teste, da atribuição e interação entre as equipes envolvidos. Percebe-se um processo sistêmico para a eliminação dos *bugs* e outras falhas realizado pelos testadores da equipe de QA em conjunto com a equipe desenvolvimento

5.2.1 A definição de *bug* e suas categorias

Entende-se como *bug*, um defeito inesperado, falha ou imperfeição que ocorre ao executar algum software ou hardware, prejudicando funcionamento e acarretando problemas como travamentos, desempenho do sistema, ou simplesmente não funcionam. (BUG, 2020)

Para Levy & Novak (2010, p.48-49) nos jogos de videogames todo o elemento que não melhora o jogo, mas o prejudica, é considerado com o *bug*. Alguns tipos de bugs são mais perceptíveis e fáceis de identificar, porém outros são quase impossíveis de reconhecer, ocorrem em um momento específico dificultando o trabalho do testador de QA. O objetivo principal do testador de QA, além de localizar todos os *bugs* possíveis, é ter a acurácia de replicá-los, de modo que gere um relatório para que os desenvolvedores possam corrigi-lo. Após a correção pelo time de desenvolvimento, são realizados novos testes pelo testador de QA para verificação se o problema foi resolvido. De forma resumida temos as diretrizes e o ciclo de trabalho de um *game tester* na cadeia de produção e desenvolvimento de um jogo digital.

Os autores Levy & Novak (2010, p.50-51), classificam os *bugs* em quatro níveis de gravidade, a saber:

- **Baixa prioridade:** São *bugs* encontrados durante o desenvolvimento que não impactam no funcionamento do jogo ou raramente acontecem, não afeta o fluxo ou forma como o jogo é percebido significativamente pelo jogador. Como por exemplo, distorção do som momentânea que

ocorre as vezes, algum erro de grafia na interface do usuário. Esse tipo de erro, as vezes é ignorado em favor de *bugs* de maior relevância e que impactam diretamente durante a partida.

- **Média prioridade:** São *bugs* que ocorrem com maior frequência do que os *bugs* de baixa prioridade, e quase sempre esse evento pode ocorrer durante o jogo. Esse tipo de erro deve ser corrigido, assim evita-se a frustração na expectativa do jogador, como por exemplo o *bug* ocorre no título de apresentação do jogo. Não afeta sua jogabilidade, mas causa uma certa irritação no usuário, antes mesmo de se iniciar a partida.
- **Alta prioridade:** São *bugs* que ocorrem na maior parte do jogo prejudicando a jogabilidade do usuário, esse tipo de erro deve ser corrigido antes de ser lançado no mercado. Como por exemplo ao abrir um porta para mudar de ambientação com o em *Resident Evil* e não conseguir, indiscutivelmente é um *bug* de alta prioridade, impacta diretamente no andamento do jogo.
- **Crítico:** Esse tipo de bug exige atenção imediata da equipe de desenvolvimento. *Bugs* críticos causam desastres como travamento, congelamento ou os dados são corrompidos durante a partida. Imagine a perda de 50 horas de gravação do jogo durante a leitura do *load*, ou congelamento da tela durante a partida de um jogo do tipo *First-Person Shooters* (FPS). Outro exemplo, se a *publisher* durante o processo de certificação do jogo ocorrer esse tipo de *bug*, há a devolução para o desenvolvedor para as devidas correções antes da publicação definitiva do jogo.

Tabela 3- Resumo dos bugs em níveis de gravidade

Bug e seus níveis de gravidade				
	Baixa	Média	Alta	Crítico
Descrição	Não afeta o fluxo ou a percepção do jogador	Causa aborrecimento,	Afeta gravemente a jogabilidade	Interrompe a jogabilidade

		mas não afeta a jogabilidade		
Exemplos	- Eventual falhas no gráfico; - Pequenas distorções no som; - Erro de digitação na interface do usuário;	- Sticky spot; - Frequentes falhas no gráfico; - Erro de digitação na tela de título; - Perda de alguns efeitos sonoros;	- Stucky spot; - Sem a presença de efeitos sonoros; - Faltando partes de diálogo; - Não conseguir abrir a porta para terminar a fase; - Não conseguir trocar de armas;	- Travamentos; - Congelamentos; - Dados corrompidos;
Ação	A correção pode ser feita posteriormente ou nunca ser corrigido;	A correção deve ser feita com brevidade;	A correção deve ser feita;	A correção exige atenção imediata; Se detectado durante a certificação, o jogo é enviado de volta ao desenvolvedor;

Fonte: Levy & Novak (2010, p. 50, tradução nossa)

E ainda, como cita Levy & Novak (2010, p.77), a premissa do trabalho de um *game tester* se baseia no primeiro momento da identificação e anotação do *bug* no banco de dados, feito esse entendimento em seguida devemos classificá-lo pelo nível de criticidade baixa, média, alta ou crítica conforme apresentado na Tabela 3. Outro ponto essencial é a categorização desse *bug* corretamente, com o propósito de fornecer um informação detalhada para a equipe de desenvolvimento para a sua correção. Caso contrário poderá causar atrasos e arruinar o planejamento proposto para o desenvolvimento do jogo. Isto posto, os autores dividiram em oito categorias principais de *bugs*, sendo:

- 1) **Visual:** São erros que afetam os gráficos de um jogo. Eles podem variar de uma cintilação quase imperceptível à problemas de textura gritantes que se destacam durante a partida. [São exemplos de *bugs* visuais:]
 - *Clipping* ocorre quando um polígono se sobrepõe ou penetra em outro polígono. Um exemplo clássico aconteceu muito no jogo *GoldenEye 007* do Nintendo 64, onde o soldado parado atrás de uma porta fechada e sua arma transpassa essa porta mesmo fechada. O *Clipping* também é um problema nos veículos, as pernas de um personagem não devem ficar para fora de um jipe, por exemplo.

- *Z-Fighting* na programação, a coordenada Z corresponde à profundidade. Se houver um problema, as texturas em diferentes profundidades podem sobrepor e lutar (por isso a origem do nome). Durante o teste da Garantia de Qualidade (QA) no Quake 4, era comum a ocorrência do *z-fighting* nas caixas de comando do elevador. Normalmente, a porta do elevador ou a textura da parede acabam em conflito com a textura original correta.

- *Screen Tearing* um *bug* visual “classico”, a tela partida ocorre quando a unidade de processamento gráfico (GPU) não consegue desenhar um quadro rápido o suficiente, resultando nesse efeito “tearing” (“partido”) perturbador. Normalmente as origens desse *bug* estão relacionadas ao desempenho – talvez por falta de tempo ou porque seja especialmente difícil de programar para a plataforma. [...]

- *Missing Textures* quando você encontra ausência de texturas, tudo o que você vê é um plano branco ou espaços reservados (imagens que “são preenchidas” quando o mecanismo percebe que as texturas estão faltando) O problema é que as texturas que faltam são esquecidas e o código quase perfeito ainda contém alguns [...] [erros].

- *Visible Artifact bug* gráfico pode ter baixa prioridade, mas isso não significa que não possam ser estranhos e irritantes. Um artefato visível é geralmente um dos muitos pequenos pedaços espalhados na tela que não estão conectados a ninguém ou a nada. Eles simplesmente estão lá, as vezes flutuando as vezes em um canto (LEVY & NOVAK, 2010, p.78-79, tradução nossa).

Iremos abordar a próxima categoria de *bug* o *Audio*

- 2) **Audio:** os *bugs* podem variar de diferenças de volume a colapsos completos de áudio. Exemplos de *bugs* de áudio incluem quedas de áudio, saltos, distorção, falta de efeitos sonoros e nível de volume. [São exemplos de *bugs* de áudio:]

-*Audio Drop* [a perda de áudio pode ocorrer durante o jogo], você pode perder diálogos importantes, um efeito sonoro de tiro revelador ou sua faixa favorita de uma música. [Como exemplo, podemos citar um ícone comumente presente na franquia do *survival horror Silent Hill*, o rádio de bolso, tem a função de alertar o protagonista com a emissão de um ruído quando há a presença de algum monstro por perto. Este aviso aumenta gradativamente à medida que o personagem se aproxima deles. Em uma possível falha no áudio do rádio impacta diretamente na dinâmica do jogo atrapalhando o feedback para o jogador.]

-*Skipping* Assim como um CD arranhado, *skipping* é facilmente identificável. Em jogos [esse termo] pode estar relacionado ao desempenho, [uma falha] na taxa de quadros. Há, entretanto, uma possibilidade de que o efeito sonoro, trilha musical em questão esteja realmente danificado.

- *Distortion* como o *skipping*, a distorção do áudio também pode estar relacionado ao desempenho, especialmente se a CPU “travar” bem no meio da reprodução.

- *Missing Sound Effect* a falta de um efeito sonoro é semelhante a uma textura ausente, com uma grande diferença: normalmente nenhum marcador de posição é usado. Se faltar um efeito sonoro,

ouvirá o silêncio e já está. A falta de um efeito de som é, em última análise, responsabilidade da programação e ou da equipe de áudio.

- *Volume Level* [considerando que você tenha acesso a tela de opções do jogo e possa controlar o nível de volume como] motor, ruído da estrada, oponentes e faixa de música, isso não é um bug. No entanto, se o jogo for mixado de uma certa maneira, e você não pode alterá-lo na tela de opções, [e este apresenta] um efeito de sonoro muito fraco ou uma música dominante e pode ser considerado um *bug* no nível de volume. [Temos um desequilíbrio no nível de volume e pode atrapalhar de certa forma a imersão do jogador] (LEVY & NOVAK, 2010, p.80-82, tradução nossa).

O próximo *bug* a ser relacionado é o *Level Design*

3) *Level Design* quando a fase é mal construída, você pode acabar com um buraco [inesperado] no chão ou até mesmo uma parede invisível. Quando as coisas não fazem muito sentido espacialmente, este é um bom indicador de que você encontrou um *bug* no *Level Design*. [São exemplos de *bugs* de *Level Design*:]

-*Stuck spot* [esse tipo de *bug* dependendo do local o jogador poderá ficar preso, em muitos casos esse problema é de autoria do *level designer*.] A geometria ruim está impedindo seu personagem de se mover. *Stuck spot* são comuns em jogos de FPS, você vai encontrá-los nos cantos e entre caixas. Se você for pego em um, na melhor das hipóteses, você pode carregar uma gravação anterior, na pior das hipóteses, você precisará reiniciá-lo. Estes *bugs* são de alta prioridade porque podem interromper o jogo

- *Sticky Spot* [esse tipo de *bug* é parecido *stuck spot* a diferença é que] você pode sair, só leva tempo e esforço [para conseguir desse estado.] Esses *bugs* têm média prioridade, um *sticky spot* pode evoluir para um *stucky spot* se o *level designer* tomar algumas decisões equivocadas após o [recebimento do] relatório de *bug*. Por outro lado, um *stucky spot* pode ser rebaixado para um *sticky spot* por um hábil desenvolvedor.

-*Invisible Wall* [a parede invisível, normalmente utilizada pelos desenvolvedores como uma fronteira limítrofe no mundo do jogo para que o personagem não tenha acesso a certas áreas.] É basicamente geometria extra sem a arte que a tornaria visível. As chances são de que a [presença dessa] parede, é a geometria remanescente de uma versão anterior do mapa [do jogo impedindo o avanço do jogador, e podemos atribuir isso a um *bug*].

-*Map Hole* um buraco no mapa é divertido, mas muito sério. Se um jogador cair em um, a ilusão é destruída imediatamente. Pior, alguns buracos no mapa são um tanto rasos e permitem que os jogadores trapaceiem atirando em outros debaixo do mapa. Este foi o caso em alguns mapas para *download* de *Call of Duty 3*. Havia um enorme buraco no mapa da fase, então muitos dos testadores caíam lá de propósito para atirar na cabeça à vontade. A única maneira de encontrar todos os buracos do mapa é andar em cada metro quadrado do mapa, não há outra maneira. É por isso que as vezes o teste do jogo envolve mais ações disciplinadas e repetitivas do que habilidade.

-*Missing Geometry* é o oposto da parede invisível. Se a arte está presente, mas não há parede ou barreira para o seu movimento, falta alguma geometria. A variedade mais comum desse bug é aquela

passagem secreta entre as paredes do castelo (LEVY & NOVAK, 2010, p.84-85, tradução nossa).

Outro *bug* que devemos analisar e de extrema importância durante as partidas é a inteligência artificial

4) Artificial Intelligence a expressão “inteligência artificial” (IA) pode levar você acreditar que os personagens do jogo sempre se comportam com inteligência, na verdade não. Quando os personagens fazem algo completamente ilógico, como olhar para uma parede por cinco minutos, pode apostar que há algo de errado com o código de IA. [São exemplos de *bugs* de *Artificial Intelligence*:]

- *Pathfinding* se houver problemas de localização de caminhos e um personagem controlado por computador não conseguir encontrar seu caminho no mapa, há três explicações possíveis: uma parede invisível está bloqueando o caminho do personagem, um buraco no mapa está quebrando o script, o IA está com defeito. [Antes de qualificar como um *bug* de IA,] as vezes é preciso eliminar as outras possibilidades [como descrito.]

- *Non-Player Character Behavior* quando o comportamento da IA, o *non-player character* (NPC) deve ser tão convincente que você quase acredita que humanos reais estão jogando com você. Quando a IA falha, [...] o comportamento do NPC também pode afetar o equilíbrio [do jogo, como por exemplo,] se você depende da sua equipe de NPC para matar enxame após enxame de inimigos, um desempenho ruim da parte deles tornará o jogo muito difícil.

[...]os *bugs* de IA se tornarão mais comuns à medida que os jogos tentarem reproduzir a inteligência real como comportamentos aprendidos (LEVY & NOVAK, 2010, p.86-87, tradução nossa).

Devemos considerar a questão da física empregada nos jogos e incluir nesta lista de classes de *bugs*

5) Physics a física nos jogos é um subsistema completo, e poderoso nisso. Tanto na jogabilidade quanto a animação são diretamente afetadas pelo código da física, uma vez que recebem “*input*” do motor de física e subsistema. Saber como detectar *bugs* físicos é uma habilidade muito desejável. [São exemplos de *bugs* de *Physics*:]

- *Breakables* uma característica comum dos jogos modernos é uma quebrável geometria. [Faz parte da composição do cenário o emprego de objetos e barreiras destrutíveis. No jogo *Space Invaders*, lançado nos *Arcade* em 1978, temos a utilização das barreiras na parte inferior da tela que tem a função de proteger parcialmente a nave do jogador, entretanto ela são destruídas tanto pelo próprio jogador como pelos inimigos.] Em *Gears of War*, isso é chamado de “cobertura destrutível” e permite que os jogadores se protejam contra o fogo inimigo até certo ponto. Quando o dano é muito extenso, a tampa se quebra em pedaços (ou seja, “objetos destrutíveis”). Agora imagine o que aconteceria se os objetos não quebrassem, ou

quebrassem e comesçassem a flutuar em direção ao céu, estes são dois possíveis *bugs* relacionados a *breakables*.

-*Dynamic Behavior* [os objetos modelados em um jogo podem ter um comportamento inadequado] em termos de física. Por exemplo, [no Quake 4] digamos que seu personagem atinja uma caixa de metal em alta velocidade, a caixa deve se mover por um certo tempo e depois parar. Entretanto, no Quake 4, a caixa às vezes continuava se movendo. Este é um indicador claro de que algo no código da física está muito errado, seja uma física falsa⁹ ou não (LEVY & NOVAK, 2010, p.87-88, tradução nossa).

Manter a integridade do jogo durante todas as etapas do seu funcionamento é primordial para manter o desempenho esperado pelo jogador assim que se inicia a partida. Dado a importância dessa condição a equipe de QA em seus testes, há a preocupação quanto aos possíveis *bugs* inerentes a estabilidade desse processo.

6) *Stability* refere-se à previsibilidade do código, se ele se comporta ou não de acordo com as intenções da equipe de desenvolvimento. [São exemplos de *bugs* de *Stability*:]

-*Freeze* [inesperadamente durante a partida a tela congela], congelar tira qualquer controle que você possa ter sobre o personagem do jogo, uma vez que o sistema deixa de responder. Esta é a razão principal pela qual os congelamentos são *bugs* críticos. Ao detectar um congelamento, certifique-se de copiar e anotar todas as informações de depuração disponíveis [com intuito de colaborar com o desenvolvedor para reparo do problema].

-*Crash* é diferente de um congelamento por um motivo facilmente explicável: a tela escurece. Como um congelamento, o jogador perde a entrada, mas a tela também desaparece. As falhas são igualmente ruins, sendo classificadas como *bugs* críticos.

-*Crash to Desktop* esse tipo de *bug* exclusivo do computador. Ele difere de outras falhas de duas maneiras: quando o jogo trava, ele volta para a área de trabalho; o jogador pode perder a entrada dentro do próprio jogo, mas o sistema operacional ainda funciona bem. Embora geralmente menos sério do que os outros *crash bugs*, uma falha na área de trabalho também é um *bug* crítico.

-*Loading* as fases de um jogo ocupam vários megabytes na memória do console. Como o processo de carregamento é bastante complexo, quaisquer erros no código podem interrompê-lo. Como o processo de carregamento é bastante complexo, quaisquer erros no código podem interrompê-lo, desde recursos faltantes até quebra do script [do jogo]. É considerado um *bug* [de alta prioridade].

⁹ Física falsa refere-se à ideia de levar o jogador a pensar que o comportamento dos objetos no ambiente do jogo seguia as leis da física do mundo real, e a solução apresentada pelos desenvolvedores foi “falsificá-las” através da codificação. Esse método foi utilizado anteriormente ao emprego de GPU's mais sofisticadas, processadores de vários núcleos e *engines*, onde foi possível uma simulação de física mais realista. (LEVY & NOVAK, 2010, p.88)

Congelamento, travamento e erros de carregamento são apenas alguns erros relacionados à estabilidade que você pode ver durante uma sessão de teste de QA. A busca por *bugs* de estabilidade pode exigir mais do programador (LEVY & NOVAK, 2010, p.89-90, tradução nossa).

Outro local passível de ser encontrado *bug* em relação a sua performance no momento do processamento do jogo.

- 7) **Performance** refere-se à velocidade com que o *hardware* processa o código. Exemplos de erros de desempenho incluem taxa de quadros, tempo de carregamento, requisitos mínimos da máquina e erros de tempo de instalação. . [São exemplos de *bugs* de *Performance*:]

-*Frame Rate* quando um jogo está sendo executado, os desenvolvedores definem metas de taxa de quadros para o título. Na maior parte das vezes, os desenvolvedores não se preocupam muito com o desempenho até bem depois da fase beta. Eles raciocinam que primeiro é necessário construir o jogo e então avaliar o desempenho. O problema com essa abordagem é que problemas graves de desempenho podem ser muito complexos a ser corrigido nos últimos dois meses de produção. É por isso que alguns estúdios mantêm a taxa de quadros desejada durante a produção. Garantindo que qualquer falha de desempenho seja detectada logo no início.

- *Load Time* embora o número exato possa variar entre cada jogo ou plataforma, longos tempos de carregamento sempre serão um problema. [Devem ser corrigidos pela equipe de desenvolvimento, principalmente se] os fabricantes de *hardware* pode obrigá-los a corrigir o problema antes do lançamento. [Nos testes do tempo de carregamento,] você será solicitado a carregar todas as fases do jogo inúmeras vezes, marcando o tempo.

-*Minimum Requirements Machine* [tratando-se dos] requisitos do jogo para computador incluem as máquinas de requisitos mínimos, também conhecidas como especificações mínimas na qual o jogo deve ser executado. Na verdade, esta é uma obrigação contratual, quando o desenvolvedor entrega o jogo, ele deve se certificar-se de que o jogo é executado em máquinas com o requisito mínimo. Essa é a única maneira de incluir milhões de possíveis compradores que não têm acesso a uma plataforma de jogos mais robustas.

- *Installation Time* esse *bug* é a típica falha de desempenho porque provavelmente resulta de um problema como o próprio instalador.

Os *bugs* de performance estão, de certa forma, entre os mais fáceis de detectar e os mais difíceis de corrigir (LEVY & NOVAK, 2010, p.90-91, tradução nossa).

A penúltima categoria de *bug* apontada pelos autores é a compatibilidade.

- 8) **Compatibility** [esse tipo de] *bugs* estão relacionados especificamente ao modo como um jogo é executado em determinado *hardware*. A maioria dos testadores nunca lidará com esses *bugs*, uma vez que a compatibilidade é um tipo separado de teste. Exemplos de *bugs* de compatibilidade incluem:

-*Vídeo Card bug* de compatibilidade de placa de vídeo era muito comum durante o teste de QA do jogo Quake 4. Os *drivers* da placa ATI faziam o jogo travar constantemente no computador. [Este tipo de teste são realizados em alguns tipos de placa de vídeo para garantir a sua funcionalidade e condizer com as configurações necessárias descritas pelo fabricante.]

-*Controller* é importante detectar o *bug* de compatibilidade porque é sempre difícil garantir que cada marca de controlador funcione com o jogo. A maior parte do trabalho árduo é feito pelo sistema operacional, mas algum esforço é deixado para os programadores do jogo. Quando um controlador [não é compatível] e não funciona, você precisa reportar um *bug* sobre isso.

-*Operating System* o sistema operacional falhar é um *bug* muito comum. Pode ocorrer do jogo ser suportado pelo sistema operacional, e mesmo assim, o jogo ainda não funcionar. Isso certamente seria definido como um *bug*.

-*Standards* sempre certifique-se de que os padrões como *USB* e *Bluetooth* sejam compatíveis. [O importante é verificar se o problema é no periférico na parte física do equipamento ou está relacionado mesmo a compatibilidade.] (LEVY & NOVAK, 2010, p.92-93, tradução nossa)

A última categoria de *bug* apresentada refere-se a conexão via internet, atualmente é um relação vital para que se tenha o melhor uso do meio via transmissão e recebimento de dados.

9) Networking os *bugs* estão relacionados especificamente à conectividade servidor-cliente e problemas de largura de banda. [São exemplos de *bugs* de *Networking*:]

-*Failed Connection* falha de conexão é um *bug* onde aparece na mensagem de falha de conexão, falha ao conectar, sem conexão e conexão interrompida, mas o significado é sempre o mesmo, seu console não pode “ver” ou “ouvir” outros consoles. [Faz-se necessário a verificação se realmente é uma falha local da rede do usuário ou uma falha de operação do sistema servidor do console.]

-*Dropped Connection* digamos que seu console foi capaz de se conectar, mas você experimentou uma queda na conexão de qualquer maneira. Há uma diferença entre os dois. Claro, se você jogou várias sessões sem problema de conexão e de repente a conexão cair, você não terá problemas para identificar o *bug*. O problema é quando você experimenta uma queda logo após conectar

-*Unaccepted Invitation* este *bug* refere-se apenas os sistemas que permitem convites para jogar. Digamos que um testador tente convidar você para um jogo e você vê o prompt e escolhe aceitar. No entanto, voce acaba enfrentando a tela de título e não o jogo com seu amigo. Esses *bugs* podem se tornar muito complicado conforme a produção se aproxima do beta, já que a maioria dos líderes de teste não presta atenção neles porque não fazem parte do jogo em si.

-*Lag* o atraso é realmente um sintoma de problemas de rede muito sério, como pacotes perdidos ou uso excessivo de largura de banda. É importante escrever um relatório muito detalhado quando você identifica um *lag*.

-*Invisible Player* se você não consegue ver o personagem de outro jogador ou um NPC, você encontrou um *bug* [onde o personagem está invisível], um sintoma de sérios problemas de rede. Esses *bugs* podem estar entre os *bugs* mais desafiadores a serem investigados porque são difíceis de repetição e rastreabilidade. Assim como o *lag*, este *bug* deve ser amplamente documentado. Não é raro ver um jogo lançado com esses tipos de *bugs*, apenas para ser corrigido com um *patch* alguns meses após o lançamento.

-*Scoring Error* não é incomum ver pontuações imprecisas em um jogo que está quase concluído. Como as pontuações são de baixa prioridade, as vezes os *bugs* desagradáveis conseguem ficar ocultos no sistema de pontuação. A lição aqui é simples não ignore *bugs* em potencial só porque eles podem ter baixa prioridade (LEVY & NOVAK, 2010, p.94-95, tradução nossa).

Para os autores essas são uma breve amostra da relação dos possíveis *bugs* dentre os inúmeros encontrados ao se realizar o teste dos jogos.

Assim temos efetivamente o processo de controle de qualidade com a localização de *bugs* no *software*, deparar com falhas no *design* ou nos controles de um jogo, documentar anomalias nos elementos gráficos e de áudio (RUGGILL, MCALLISTER, NICHOLS & KAUFMAN, 2017, p. 213).

5.3.2 Realização de testes no ciclo de produção do jogo, uma visão geral

Durante o processo de produção são definidos alguns marcos de entrega no decorrer do desenvolvimento do jogo, como pré-produção, *kickoffs*, alfa, beta, ouro e pós-lançamento. No pensamento de Schultz, Bryant & Landell (2005, p. 129), independentemente do tipo de plataforma que será lançada o jogo, de quanto tempo de desenvolvimento, semanas ou longo de vários de anos, não importa o tamanho do jogo, o teste do jogo deverá seguir a mesma estrutura desses marcos. Nesse contexto, pretende-se abordar qual a relevância da realização do teste na etapas supracitadas.

5.2.2.1 Teste Pré-produção

O teste começa quando o projeto começa, apesar de não termos testadores nesse estágio, mas sim o planejamento das etapas de produção necessita de avaliação, correção e discussão acerca dos recursos, ativos, concepção de arte, programação, entre outros. Nessa etapa a incumbência da garantia de qualidade, em

específico o gerente de QA, é examinar a documentação do *design*, TDD e o cronograma assim que o projeto é concebido. Assim, elabora-se o escopo de teste com a estimativa de horas, recursos financeiros e humanos para a realização dos testes até o lançamento do jogo no mercado (SCHULTZ, BRYANT & LANGDELL, 2005, p.130).

Na visão de Schultz, Bryant & Langdell (2005, p. 133-135), nessa etapa o líder de QA especifica e determina os critérios de aceite para cada fase de teste, geralmente Alfa, Beta e Ouro do jogo. Cada uma dessas fases requer três elementos necessários no planejamento de certificação para cada fase de teste, sendo:

- **Critérios de entrada:** Realiza-se um conjunto de testes na *build* antes de entrar em uma determinada fase de teste.
- **Critérios de saída:** Conjunto de teste que a *build* deve passar antes da conclusão da fase de teste.
- **Data prevista:** Define a data planejada para uma fase específica de lançamento.

O emprego desses critérios estabelece-se uma sistematização de trabalho, assim pode-se evitar possíveis conflitos posteriormente no projeto, digamos realizar um teste Beta em uma *build* que ainda não cumpriu esses requisitos, por acreditarem que já está pronta para esse fim. Esses critérios devem ser aprovados pelo gerente de QA e divulgados aos membros da equipe do projeto.

Os autores ainda comentam que nesse fase da pré-produção o líder de QA tem algumas outras atribuições como a participação regular das revisões de *design*. Isso faz com que ele fique ciente na ocorrência de alguma mudança do escopo do *design*, à vista disso aconselha o gerente de projeto sobre quaisquer desafios técnicos ou complicações de teste que possam surgir, o que é capaz de acarretar a alteração do escopo e o fluxo do teste. Outra atividade a cargo do líder é a configuração do banco de dados de rastreamento dos *bugs*, trabalho em conjunto com o gerente de projetos em definir qual é o nível de permissão dos demais integrantes da equipe, direito de editar o campo do banco de dados, visualização, acesso local ou remoto, entre outras configurações. O líder de QA desempenha o papel de delinear o plano

de teste que atua como um manual para equipe de teste abrange os objetivos da equipe de teste, determina o cronograma de teste, metas intermediárias para um ou mais marcos de testes antes do lançamento final do jogo, riscos que possam comprometer os objetivos do teste e informações como gerenciar esse tipo de risco.

Existe a possibilidade da equipe de teste ser submetida a executar testes preliminares direcionados de certos recursos do jogo a fim de ajudar a equipe de desenvolvimento na construção do código. Este tipo de teste é chamado de teste modular, a medida que o código se torna funcional, os módulos são testados até a compilação completa aspirante à versão de teste Alfa seja aceita (SCHULTZ, BRYANT & LANGDELL, 2005, p. 137).

No entendimento de Cohen, Bustamante & Park (2010, p. 218), durante a pré-produção a equipe trabalha para gerar o protótipo do jogo. Para alcançar esse designio devem ser testados as mecânicas, os recursos, *design* e arte pretendido, para garantir que a compilação feita na *engine* funcione corretamente no *hardware* de destino. A finalidade dos testes é certificar que o projeto desenvolvido possa funcionar sem que ocorra travamento ou problemas de funcionalidade, diminui-se o risco de descobrir que o projeto não funcionará nesse *hardware* de destino somente no estágio de produção. Os autores citam que alguns dos fracassos ocorridos no desenvolvimento foram causados por falta de planejamento e testes.

No pensamento de Irish (2005, p. 5), o autor recomenda que na pré-produção seja feito um protótipo ou minijogo para que se tenha um caso de teste do jogo que está sendo desenvolvido. Deste modo, permite que a equipe aprenda sobre o processo e façam ajustes pertinentes antes de prosseguir com o desenvolvimento do projeto.

A equipe de design aproveita o protótipo para teste, vendo seus designs em ação com outros jogadores, identificando o que é confuso ou frustrante e o que é divertido e envolvente. Isso os ajudará a estruturar e construir os designs de nível completo e a jogabilidade. O produtor deve revisar com quem joga o protótipo o que gosta e não gosta no design e na mecânica, fornecendo informações aos designers sobre todos os pontos com recomendações de como pode ser melhorado (COHEN, BUSTAMANTE & PARK, 2010, p. 224, tradução nossa).

5.2.2.2 Teste *Kickoffs*

Reuniões iniciais são benéficas para o desenvolvimento do *software*, há o balizamento do processo, melhor solução de problemas e redução do tempo de ciclo. A elaboração de uma lista de verificação prepara o testador no entendimento dos requisitos, documentação do que se pretende testar no jogo, ciência dos testes (SCHULTZ, BRYANT, & LANGDELL, 2005, p. 138-141).

5.2.2.3 Teste Alfa

Nessa ciclo é chegada a hora de culminar a experiência da equipe de teste para a melhoria do jogo, na observação do testador ao longo do teste alfa o design do jogo é aprimorado, nesse momento os esforços da equipe desenvolvimento são avaliados.

A fase alfa se caracteriza pelo primeiro grande marco de um jogo, exige a conclusão de recursos, o jogo pode ser jogado do início ao fim. No entanto, pode haver algumas lacunas e os recursos de arte não podem ser o finais, mas os mecanismos e a interface do usuário estão completos, geralmente neste estágio temos *placeholder*¹⁰ e *assets* temporários. Ao contrário do que alguns podem pensar, o alfa é mais sobre bloquear os recursos do que ter um jogo livre de bugs. Os testadores nas suas atividades se deparam com modelos sem definição e distorcidos, arte faltando, sem som e outras adversidades no período de teste alfa, é trabalho do testador ajudar a superar esses problemas e chegar a um título estável e completo no final do estágio alfa (LEVY & NOVAK, 2010, p. 52 e 215-216; NOVAK, 2017, p.347).

Adams (2003, p. 66-67 e 305), comenta que o teste alfa é o ponto de desenvolvimento em que todos os recursos do jogo estão presentes, mesmo se não houver todos os gráficos e dados disponíveis o jogo pode ser plenamente testado.

¹⁰ É a utilização de um esboço de um arte que facilita o trabalho dos artistas e desenvolvedores até a arte final..... explicar melhor

Em um jogo de futebol o jogo pode ser jogável, mas nem todos os estádios estão prontos. O teste Alfa é um teste interno feito pela *Publisher*, desenvolvedor ou uma combinação de ambas frentes. A maior parte do tempo dedicado pela equipe de desenvolvimento são os ajustes e correções de *bugs* encontrados durante o teste alfa.

Os autores Schultz, Bryant & Langdell (2005, p. 142-143), citam que ao longo do teste alfa o *design* do jogo é aprimorado, recursos são testados e revisados ou descartados. É fundamental que os conjuntos de testes sejam estritamente observados, nesse processo todos os módulos do jogo devem ser testados ao menos uma vez. Determina-se a taxa de quadros, tempo de carregamento, dentre outros aspectos que servem como parâmetro para a liberação do teste. Tratando-se do estágio inicial de teste é aceitável haver alguma discrepância nesses valores determinados, mas nos testes subsequentes já não são mais admitidos. São definidos alguns critérios de entrada da fase alfa, como:

- Os principais recursos do jogo existem e podem ser testados, dependendo alguns módulos são separados para fins de teste.
- O testador pode percorrer o jogo por algum caminho do início ao fim. Isso caracteriza que o jogo é linear ou temos algum componente linear, por exemplo modo carreira de um jogo de corrida. Em jogos não lineares, que ocorre na maioria dos casos, há a necessidade de estabelecer com as áreas de teste e projeto uma meta para a conclusão do teste para esse tipo de jogo, por exemplo três de doze minijogos.
- Cada fabricante da plataforma de hardware proprietária tem sua própria lista de verificação de requisitos técnicos, o teste será realizado em relação a essa lista para garantir que o jogo esteja em conformidade com esses padrões estabelecidos, esses requisitos são exigentes e rigorosos, o código passa pelo menos em 50% dessa lista.
- Interface básica está completa temos o menu principal, a maioria dos submenus, a interface do jogo, Heads-Up Display (HUD) deve ser ao menos funcional. A documentação preliminar sobre nova funcionalidade, alguma alteração no mapa dos controles e *cheat codes* está disponível para o QA.

- O jogo é compatível com a maioria das configurações de hardware e software especificadas. Seja executado em múltiplas plataformas de consoles ou em um computador para uma gama de especificações e configurações destes gêneros.
- O script de cada fase está implementado, refere-se principalmente ao modo história para um jogador. Um jogo candidato alpha que exigisse que o testador carregasse cada nível manualmente falharia neste critério.
- Os periféricos originais de cada fabricante da plataforma devem ser suportados durante a fase de teste alfa e funcionar adequadamente.
- A arte final ou esboço (*placeholder art*) está disponível para todas as áreas do jogo. Todos os níveis e personagens devem ser texturizados e animados, provavelmente será refinado até a versão Beta
- O modo multijogador online pode ser testado. Deve ser implementada a rede LAN para pelo menos dois consoles possam se conectar.
- O áudio provisório (*placeholder audio*) está implementado, são atribuídos áudios provisórios onde houver necessidade. Existe a possibilidade que ainda a sessão de gravações de áudio finalizadas ainda não tenha sido concretizada na etapa alfa.

Para Flynt e Salem (2005, p. 370, 625-627), nesta fase a atividade principal é procurar e eliminar defeitos detectados na funcionalidade do jogo. Permite verificar se a funcionalidade implementada é realmente a desejada para aquele jogo, há a oportunidade de interação entre desenvolvedores e testadores, eles podem dar sugestões sobre recursos que podem ser eliminados e o que pode ser melhorado. Os autores comentam da necessidade de se realizar um teste alfa para validar alguns aspectos funcionais do jogo, coletar informações de como será conduzido o progresso do seu desenvolvimento e quais serão as diretrizes a serem tomadas. Após o teste os resultados são registrados e os desenvolvedores devem estar preparados para refinar o sistema.

O teste se torna mais rigoroso no estágio alfa. Pode haver algum trabalho para o polir os recursos do jogo, muito do empenho é destinado para encontrar os *bugs* que travam ou matam o jogo completamente (IUPPA & BORST, 2010, p. 326).

Logo, os testadores executam o plano de teste elaborado pelos gestores da equipe de QA, para verificação de cada recurso. O contingente de testadores decorre do tamanho e importância do projeto, o teste alfa pode durar de semanas a meses. O fluxo de trabalho permeia em encontrar *bugs* e registrá-los no banco de dados, na correção da mecânica do jogo, é um processo iterativo com as várias equipes envolvidas no projeto. Quando o time de QA considerar que houve êxito na correção e concluída parte de testes, eles podem disponibilizar para o próximo ciclo de teste beta. (ADAMS, 2003, p. 66-67; ADAMS E., 2010, p. 384)

5.2.2.4 Teste Beta

Ao término da fase alfa, a equipe de desenvolvimento tem uma ideia muito clara do jogo que está criando, o foco é aperfeiçoar o que já foi criado pela equipe. A fase beta é a etapa onde o desenvolvedor corrige quaisquer *bugs* importantes, e também é a chance de aprimorar o jogo antes da etapa ouro. A esta altura sua equipe de testadores pode ser expandida para garantir todas as interações e caminhos possíveis durante o jogo, na busca de *bugs* que estejam ocultos no software. Algumas atividades durante o teste beta vão além da procura por *bugs*, cabe verificar se há quebra de progressão, erros de carregamento, erros de instalação, plataformas suportadas, averiguar itens e acessórios utilizados, verificação de desempenho e da taxa de quadros. Os testadores inclusive podem fornecer as melhores opiniões e informações, que serve de análise para decisões relacionadas à alteração de *design*, rebalanceamento, adição ou corte de recursos. No que se refere a comentários e sugestões de jogabilidade devem ser registrados para uma possível implementação futura. (LEVY & NOVAK, 2010, p. 217-218; SCHULTZ, BRYANT, & LANGDELL, 2005, p. 144; IUPPA & BORST, 2010, p. 319; MENCHER, 2003, p. 30)

Na visão de Adams (2003, p. 68, 306), a fase beta se enquadra no momento que todo o conteúdo do jogo estiver pronto, todos os níveis projetados, toda arte e áudio estão criadas, ou seja o jogo está completo, todos os recursos estão presentes no software e todos os *assets* foram criados. Os testadores internos neste estágio concentram-se em averiguar se cada recurso funciona e também se funciona

plenamente em todos os níveis do jogo. Todos os elementos devem ser verificados para que se garanta esse objetivo.

Além disso, é o marco mais importante, o objetivo durante o beta é estabilizar o projeto e eliminar o máximo de *bugs* possíveis, isolar todos os *bugs* significativos, realização de testes completos e problemas de performance. Todos os *bugs* de alta prioridade deve ter sido solucionados e os *assets* definitivos, ao atingir a versão beta inicia-se a etapa de polimento do jogo. O marco beta pode ser dividido em duas categorias de teste o beta aberto e o beta fechado (LEVY & NOVAK, 2010, p. 53-54; NOVAK, 2017, p. 348).

Ao se optar na realização pelo teste externo, deve-se ter em mente que há uma ampla gama de configurações de hardware que podem ser diferentes daquelas utilizadas pelos testadores internos da equipe de produção. O beta aberto, conhecido como beta externo, permite que o público em geral participe do teste do jogo. A execução do teste aberto é mais complexa, pois atua de forma aleatória. Como os testadores não são profissionais ocorre a incerteza se eles investigarão detalhadamente o jogo, ou serão capazes de descrever os *bugs* encontrados de forma efetiva para os programadores. O teste beta aberto requer um gerenciamento acurado, a qualidade dos relatórios de *bugs* tende a ser bastante irregular (ADAMS, 2003, p. 68-69; 108-111).

Frequentemente tem se optado pelo teste beta aberto, devido principalmente, pelo número crescente dos gêneros de jogos online, principalmente os MMOGs (*Massively Multiplayer Online Games*). Como esse tipo de jogo aumenta o número de componentes online exponencialmente a medida que os jogadores se conectam simultaneamente nos servidores, não há contingente de testadores suficiente para que equipe de QA consiga testá-lo. Nesse caso o beta aberto é utilizado para teste dos servidores, equilibrar a jogabilidade, encontrar *bugs* diversos e testar a funcionalidade do jogo a medida que um novo usuário se conecta a rede. (LEVY & NOVAK, 2010, p. 54-55).

Para Berthke (2003, p. 155) esse tipo de teste beta aberto tem grande valia pois coloca na mão de quem irá adquirir futuramente o jogo, é uma maneira de saber se

está sendo desenvolvido um jogo que realmente as pessoas apreciarão. Irish (2005, p. 265) considera uma ótima oportunidade da equipe de desenvolvimento identificar os pontos fracos do seu produto, antes de chegar no mercador consumidor. Contudo, no teste aberto corre-se o risco de reverberar uma propaganda negativa, por não atender a jogabilidade esperada ou não ser divertido o suficiente, mesmo sendo um produto inacabado.

Em relação ao beta fechado, chamado também de beta interno, configura um grupo específico de pessoas selecionadas para executar o teste e contribuir para melhoria do jogo, esse teste ocorre primeiro e é cuidadosamente controlado. Os recursos e código do jogo devem estar noventa por cento completos de acordo com o projeto e a documentação técnica. Se o beta fechado for bem sucedido, pode-se optar pelo beta aberto. Os testadores recrutados participam do teste no escritório da *Publisher* ou da desenvolvedora, e preenchem um questionário ou ocorre a discussão em grupo sobre o jogo testado (LEVY & NOVAK, 2010, p. 54; SCHULTZ, BRYANT, & LANGDELL, 2005, p. 301; IRISH, 2005, p. 265).

Os autores Schultz, Bryant e Langdell (2005, p. 144-145) e também Mencher (2003, p. 30), destacam os critérios de entrada na fase beta, sendo:

- Todos os recursos estão implementados
- O código deve passar em 100% da lista de verificação do fabricante da plataforma de hardware proprietária, pronto para submissão da pré-certificação. Este processo permite que a equipe de QA do fabricante da plataforma avalie o jogo em relação aos seus critérios e avise aos desenvolvedores sobre possíveis problemas de conformidade.
- O testador pode percorrer todos os caminhos do jogo, quaisquer *bugs* que prejudiquem o andamento do jogo são eliminados.
- O jogo está com a interface do usuário finalizada.
- O jogo é compatível com todas as configurações de hardware e software especificados.

- Toda a lógica do jogo e inteligência artificial estão finalizadas. A programação está completa em questão de jogabilidade, o jogo conhece suas próprias regras, os perfis de IA estão completos.
- Todos os controles designados pelos desenvolvedores e *Publishers* são compatíveis e funcionam plenamente.
- A arte final está implementada, não deve haver nenhum esboço nesse etapa do processo.
- O áudio final está implemantado, todo o áudio preliminar utilizado anteriormente foram substituídos. Eventualmente algum recurso ainda precisa ser integrado, mas não deve impactar o jogo.
- Todos os modos online estão completos e podem ser testados.
- A versão do texto empregada no jogo está finalizado no idioma original e em outros idiomas.

Berthke faz uma reflexão sobre os objetivos da utilização dos testadores na fase beta na condução do desenvolvimento do artefato

Por que você está usando testadores beta? Você está procurando publicidade gratuita para o jogo? Você está procurando publicidade gratuita para o jogo? Ou você está procurando ótimos testes de compatibilidade? Ou o seu jogo é um jogo multijogador massivo que simplesmente deve ter o beta testado por um grande número de consumidores? Você está a procura de uma segunda opinião sobre a interface ou o equilíbrio do jogo? Todos esses são bons motivos para usar testadores beta, e seu plano para os testadores beta deve refletir esse motivo (BERTHKE, 2003, p. 256, tradução nossa)¹¹.

Já Schultz et al. (2005, p. 302), expõe algumas estratégias para orientar o testador beta de como proceder na execução do teste. Alguns exemplos se baseiam em encontrar maneiras de ficar preso no ambiente do jogo, tentar encontrar maneiras de acessar modos ou locais bloqueados, tentar comprar itens e habilidades que foram projetados para serem utilizados em um nível mais alto que o seu. Em situações de um jogo multijogador faça alguns eventos para que aconteçam ao mesmo tempo,

¹¹ Why are you using beta testers? Are you looking for free publicity for the game? Or are you looking for great compatibility testing? Or is your game a massively multiplayer game that simply must be beta tested by a great number of consumers? Are you looking for a second opinion on the interface or the game's balance? All of these are good reasons to use beta testers, and your plan for the beta testers must reflect the reason.

como o máximo de pessoas possíveis em uma mesma partida, vários jogadores no mesmo espaço do jogo ou uma troca de itens entre jogadores.

Berthke (2003, p. 295) aconselha que o gerenciamento seja feito com cautela caso esteja trabalhando com prazo e orçamento fixos para o lançamento do jogo. Cabe a desenvolvedora priorizar os *bugs* de alta prioridade que impactam diretamente no jogo, e descartar os *bugs* insignificantes que não atrapalham o conteúdo do jogo de forma substancial. Sem dúvida, nesse caso, não estamos nos referindo a *bugs* simples de fácil resolução, como erro de ortografia entre outros. O autor defende a ideia de que o plano de teste beta deve ser liberado gradualmente aos testadores, em um grupo de teste numericamente menor no seu início, após análise dos testadores temos a correção dos *bugs* encontrados pelo grupo. Em seguida, constitui-se um grupo com maior quantidade de testadores, repete-se o mesmo ciclo de trabalho sucessivamente, até o número final de testadores estabelecido para o teste. Os testadores trabalham na busca de *bugs* e falhas no equilíbrio do jogo que escaparam à atenção das equipes internas de controle de qualidade.

Em algum momento cabe ao gerente de projeto decretar que o teste beta está concluído, o jogo está em um estado chamado de bloqueio de *design*. O foco da equipe de teste nesse momento é continuar com os testes em relação a cada compilação de maneira iterativa, pois cada defeito corrigido pode ser que seja introduzido um novo defeito. Ao final da fase beta, algumas decisões deverão ser tomadas pela equipe de projeto em relação ao *bugs* encontrados durante as etapas de desenvolvimento. Em determinados casos opta-se por uma atualização de software pós-lançamento ou *patch*, em outra situação a correção do *bug* incorre em alto risco técnico envolvido na sua correção superem o impacto negativo do defeito, ou por questões de prazo de entrega. Deve ser feita uma avaliação do *bug* com os setores de desenvolvimento, produção e qualidade, e discutir os aspectos positivos e negativos de corrigi-lo ou deixá-lo no jogo. Independente da escolha, é fundamental que a equipe de teste sempre descreva sobre cada *bug* encontrado, é uma informação preciosa para que as melhores decisões de negócio sejam tomadas (SCHULTZ, BRYANT, & LANGDELL, 2005, p. 145-146).

5.2.2.5 Teste Ouro

Após o término da fase beta, é chegada hora de ingressar na fase ouro, a equipe de desenvolvimento e o departamento de QA verificam se o jogo está pronto para ter o código liberado para aprovação final. A equipe de QA verifica o jogo conforme o plano de testes e outros requisitos de liberação do código (CHANDLER, 2012, p. 371-372). Os testadores executam um intenso período de testes a procura de algum bug remanescente que não possa ser desconsiderado devido a sua gravidade, e cause o impedimento do envio do jogo para aprovação final. A equipe de teste é a última barreira antes do lançamento definitivo, e se encarrega em tentar “quebrar” pela última vez o jogo (SCHULTZ, BRYANT, & LANGDELL, 2005, p. 147). Para Berthke (2003, p. 296), o desafio em gerar o candidato final é evitar criar novos *bugs* ao solucionar os *bugs* antigos, ocasionalmente os novos *bugs* podem ser de maior gravidade do que os *bugs* anteriores.

Os autores Schultz, Bryant e Langdell (2005, p. 147) e Mencher (2003, p. 31), destaca os critérios de liberação na fase ouro, sendo:

- A gerência avaliou o produto e o banco de registro de *bugs*, e considerou que o jogo está pronto para o teste final.
- A instalação está pronta e completa para o lançamento.
- Todos os recursos estão implementados e prontos para o lançamento.
- Todos os *assets* finalizados e presente no jogo.
- Os *bugs* críticos de alta prioridade já foram corrigidos.
- A performance esperada para o jogo foi alcançada para o lançamento.
- O produto suporta as plataformas estipuladas para lançamento.

Levy e Novak (2010, p. 218-219), menciona que atualmente as plataformas estão conectadas à internet, os jogos são regularmente lançados na condição de quase finalizados e corrigidos por meio de atualizações via *download*.

No caso do lançamento do jogo for para uma plataforma de console, há o processo de enviar o jogo para a validação do fabricante, chamado de teste de liberação. O fabricante testa meticulosamente o jogo, no qual deve atender ou exceder

aos seus requisitos de garantia qualidade, funcionalidade e jogabilidade. Existem duas fases de teste, a primeira testa o código em relação à lista de verificação de requisitos técnicos e a outra teste o código relacionado à funcionalidade e estabilidade do jogo. A equipe de QA do fabricante emite um relatório de *bugs* encontrados no teste de certificação. Após análise da desenvolvedora de quais *bugs* devem ser corrigidos, o jogo é reenviado para o fabricante. Se esta versão do jogo foi aceita ela se torna elegível para receber certificação final, a validação para lançamento do jogo (IRISH, 2005, p. 31; SCHULTZ, BRYANT, & LANGDELL, 2005, p. 148).

5.2.2.6 Teste Pós-produção

Após o lançamento do jogo no mercado, este pode sofrer atualizações de *software* ou *patches* para correção de algum *bug* que tenha passado despercebido ou intencionalmente para ajuste posterior. Cada nova correção ou algum aprimoramento necessário no jogo, envolve planejamento e requer mais testes. Geralmente a equipe de desenvolvimento lança mais de um pacote de atualização, o teste se torna cada vez mais complexo a interoperabilidade deve ser testada. Novos pacotes de expansão devem ser testados cada nova implementação, pois podem afetar o funcionamento da versão anteriormente instalada (SCHULTZ, BRYANT, & LANGDELL, 2005m p. 148-149).

5.2.3 Outros métodos de teste utilizados no desenvolvimento de um jogo

Nesta subseção descrevemos de forma resumida alguns outros métodos empregados no plano de teste da equipe de QA.

5.2.3.1 Teste Caixa Preta – “*Black Box*”

A grande maioria dos testes realizados em games são do tipo caixa preta. No teste caixa preta, o testador não tem acesso ao código fonte da programação do jogo, não faz ideia de como foi o seu desenvolvimento e a sua implementação. A preocupação aqui é a encontrar defeitos utilizando os mesmos dispositivos de entrada disponíveis para um jogador típico. Temos vários dispositivos que são conectados na

entrada do videogame como: *joysticks*, controladores dos mais diversos, *headset*, mouse, dentre outros. Esses dispositivos acionam os comandos que são interpretados pelo *software*, e produz na saída desse processo uma determinada reação para aquele dispositivo quando acionado (pulo, movimento do personagem, vibração do controle, etc.). Assim, com base no que é visto, sentido e escutado durante a partida, constantemente o jogador altera e ajusta esses comandos de acordo com o que é proposto no jogo. O jogo por sua vez, faz ajustes na saída em decorrência aos comandos recebidos na entrada, cria-se um ciclo contínuo de *feedback* entre o jogador e o jogo. Nessa interação é onde temos a descoberta do que está errado com jogo, o testador pode avaliar a funcionalidade ou jogabilidade e quanto divertido é a proposta, prestando atenção a esses detalhes. O teste de caixa preta faz com que os testadores pensem e se comportem como jogadores. Em contra partida, como não se conhece o processo de implementação, o teste pode se tornar demorado a procura de *bugs* (SCHULTZ, BRYANT & LANGDELL, 2005, p. 151-153; MENCHER, 2003, p. 27-28; BERTHKE, 2003, p. 155).

5.2.3.2 Teste Caixa Branca – “*White Box*”

O teste caixa branca é possível verificar o código fonte implementado no desenvolvimento do jogo. Há um grande desafio identificar um trecho do código do jogo e prever como será a interação com os demais trechos, e se o código foi responsável por todas as combinações e ordens de entrada possíveis. Ao realizar o teste caixa branca, são executados módulos específicos e os diferentes caminhos que o código pode seguir nesse módulo. Existem situações que esse tipo de teste é recomendado, por ser mais prático e necessário, como o teste feito pelos desenvolvedores antes de integrar o novo código ao resto do jogo, teste módulos de códigos que se tornarão parte de uma biblioteca reutilizável em vários jogos ou plataformas ou teste de métodos ou funções que são partes essenciais de um mecanismo de um jogo ou produto de middleware. Um testador precisa entender a codificação e lógica de programação para identificar um erro de código quando for realizar um teste caixa branca (SCHULTZ, BRYANT & LANGDELL, 2005, p. 153-154; MENCHER, 2003, p. 27-28; BERTHKE, 2003, p. 154-155).

5.2.3.3 Teste Ad Hoc

Ad hoc é uma frase em latim que podemos traduzir como “para este propósito específico”, é um único teste improvisado para responder uma pergunta específica. É o modo de teste que melhor explora o jogo, vagando por ele como se fosse um labirinto. Esse tipo de teste aborda um método menos estruturado na busca por defeitos de um jogo, permite encontrar *bugs* que as vezes não são descobertos usando a lista de verificação e o plano de teste, mesmo diante de um planejamento criterioso constituído pelo equipe de gerenciamento (SCHULTZ, BRYANT & LANGDELL, 2005, p. 287).

Quando é feito o teste ad hoc, não há preocupação de uma lista de verificação para acompanhar seu teste, é como se estivesse jogando despreziosamente na sua casa. Apesar disso, o teste visa à busca de *bug* e na exploração do jogo. Para Gordon Walton, na época codiretor da BioWare Austin, comenta que esse teste é valioso para a validação de usabilidade e a jogabilidade, mas não é um substituto para práticas metódicas da garantia da qualidade. O gerente de QA Jerome Strach da empresa Sony Computer Entertainment America na época, menciona que o teste ad hoc é onde o QA para jogos realmente se destaca dentre os outros aspectos da garantia de qualidade (LEVY & NOVAK, 2010, p. 144-145).

Existem dois tipos principais de teste ad hoc, no primeiro, teste livre, o testador improvisa no decorrer da execução do teste, de forma mais intuitiva e menos estruturada. E o segundo, teste direcionado, que visa resolver um problema ou encontrar uma solução específica. O teste ad hoc oferece a oportunidade de jogar da forma que você gostaria, permite que explore modos e recursos do jogo que estão além do seu escopo de testes, um novo olhar em várias partes do jogo. Apesar de menos estruturado, o teste em ambos os casos deve ser documentado e ser passível de reprodução na ocorrência de um *bug*. Realizar o teste ad hoc no início ajudará a revelar quaisquer deficiências remanescentes em seu plano de teste, tabelas combinatórias e diagramas de fluxo de teste (SCHULTZ, BRYANT & LANGDELL, 2005, p. 287).

Na visão de David Wilson¹², a maior utilidade da utilização desse tipo de teste é na descoberta de situações incomuns no jogo que o usuário final pode tentar executar e que não são previstas pelos desenvolvedores. Essas ações incomuns podem gerar um bom número de problemas importantes que podem ser corrigidos antes do lançamento do jogo. Entretanto, é quase impossível cobrir todas as funcionalidades de um software dessa forma. E também, é provável que os testadores se concentrem em áreas pelas quais têm preferência em relação as outras, não testando outras áreas do jogo. Para diminuir esse problema, são feitos planos de testes a parte específicas do *software*, mesmo assim não é suficiente para compensar essa particularidade.

5.2.3.4 Teste Automatizado

Teste automatizado se baseia em qualquer forma de teste realizada por meio de ferramentas de automação. Esse método é muito debatido na indústria de jogos, alguns argumentam que todos aspectos dos testes de jogos devem ser automatizados, outros defendem que sua viabilidade em certos casos específicos. No processo de automatização do teste, temos algumas vantagens quanto a sua utilização, como por exemplo: maior confiabilidade, maior consistência nos resultados, teste mais rápido, o alto poder de simulação de jogadores virtuais simultâneos em uma partida sem a necessidade de todo esse contingente de testadores fisicamente e um maior nível no índice de repetição do teste (SCHULTZ, BRYANT & LANGDELL, 2005, p. 327).

Entretanto para objetivar o emprego da automação do teste, algumas questões devem ser avaliadas, sendo: os custos de implementação do sistema de teste automatizado, a capacidade de reutilização do código de teste automatizado de um jogo que possa ser utilizado a cada novo título, a improbabilidade de substituir o fator humano quanto a determinação do fator diversão, jogabilidade e aspectos da mecânica do jogo. Deve ser definida uma estratégia para o teste automatizado, onde

¹²Disponível em: https://www.gamasutra.com/view/feature/132398/quality_quality_assurance_a_.php - Acessado em 17/12/2020

envolve uma combinação de habilidades de desenvolvimento de *software* e gerenciamento de banco de dados, levando-se em conta os requisitos de teste e também a expertise do corpo técnico (SCHULTZ, BRYANT & LANGDELL, 2005, p. 327-332).

Mencher (2003, p. 32-33), comenta que ao contrário dos *softwares* tradicionais, sendo projetados e codificados de forma linear o comportamento torna-se previsível, com alguns caminhos lógicos ou opções para executar determinadas funções. O jogo não é projetado dessa forma, o software foi programado para desafiar o usuário, dependendo das ações que o jogador escolher o programa se comporta de modo diferente para aquela ação. Por esse motivo, na visão do autor, o uso das técnicas de teste automatizado na área de jogos não tem sido muito eficaz. Outra observação, dependendo do tempo do ciclo de desenvolvimento do jogo, a utilização da ferramenta de teste automatizado não se justifica devido ao tempo gasto de desenvolvimento para sua utilização. O teste automatizado se torna atraente na medida que o tempo do ciclo de desenvolvimento do jogo for maior.

A dinâmica de teste de alguns jogos pode ser tão complexa, que os testadores necessitam do auxílio de um computador para revisar todas essas mudanças ocorridas. Alguns tipos de jogos são mais adequados para o teste automatizado e se beneficiam quanto ao uso, dentre eles podemos citar os MMOG e alguns tipos jogos de RPG possuem um mapa extenso, a escala total desses sistemas e sua natureza distribuída os tornam difíceis de testar. O teste automatizado não é para todas as partes do jogo, mas para aqueles elementos que tem um grande número de ações repetitivas, simular um grande número de servidores e clientes simultaneamente. Temos os jogos baseados em níveis, como os de tiro em primeira pessoa, utilizando um script simulando um jogador atirando repetidamente em um determinado padrão. Ou também algum aspecto de *design*, como os componentes visuais da GUI (botões, menus, ícones, etc) que podem ser testados sistematicamente a cada nova *build* (SCHULTZ, BRYANT & LANGDELL, 2005, p. 333-336).

5.2.3.5 Teste de Funcionalidade

Durante o teste de funcionalidade, o testador certifica que o jogo está funcionando corretamente e também analisa o comportamento esperado durante a execução do teste. Este tipo de teste é realizado para confirmar que todos os recursos do jogo retornam o resultado esperado, como seleção de personagens, carregamento do jogo, travamentos, dentre outros. Os testadores de QA estão a procura de problemas genéricos dentro do jogo no que diz respeito aos gráficos, na interface, áudio ou na mecânica, tudo que seja funcional, neste momento não preocupação quanto critérios subjetivos como o fator diversão do jogo (LEVY & NOVAK, 2010, p. 147). Chandler (2020, p. 248), comenta que esse teste é mais estruturado, onde é seguido o plano de teste e a funcionalidade do jogo é verificada quanto a *bugs*.

5.2.3.6 Teste de Fumaça – “*Smoke Testing*”

Esse teste possibilita testar a *build* e determinar algum problema na execução da compilação a cada módulo implementado, o testador verifica problemas de desempenho de inicialização e as funções básicas como o menu principal, entrada e saída do jogo. O teste de fumaça deve ser realizado a cada nova versão enviada para teste, na sua essência é um curto ciclo de testes de validação se o *software* realiza ou não suas funções principais (SCHULTZ, BRYANT & LANGDELL, 2005, p. 161).

5.3.3.7 Teste de Regressão

O teste de regressão é o processo de verificação que os testadores realizam a procura de *bugs* antigos no código atual. O teste de regressão é essencial para reduzir os possíveis danos ocasionados por algum *bug* que passe despercebido. Outro fator importante, é que nessas correções de *bug* executada pela equipe de desenvolvimento podem gerar outros possíveis *bugs* (LEVY & NOVAK, 2010, p. 148).

O conjunto de testes para o teste de regressão é a lista de *bugs* inseridas no banco de dados que a equipe de desenvolvimento fez as devidas correções. Cada testador terá sua lista de verificação e reproduz as etapas descritas da ocorrência do *bug*, assim ele consegue verificar se realmente o defeito foi corrigido. Alguns defeitos são facilmente verificáveis como erros no texto, recursos ausentes, no entanto outros

defeitos são difíceis de reprodução como os travamentos. Terminado o teste o líder de QA pode optar em permanecer em aberto para o próximo teste de regressão, antes de sinalizar que o defeito foi definitivamente resolvido. Com o teste de regressão temos uma boa noção de como está andamento do projeto, uma alta taxa de correção dos defeitos encontrados demonstra a efetividade da equipe de desenvolvimento. No entanto, uma baixa taxa de correção é motivo de preocupação (SCHULTZ, BRYANT & LANGDELL, 2005, p. 161-162).

5.2.3.8 Teste de Progressão

O teste de progressão envolve jogar integralmente, com o objetivo de garantir que a jogabilidade nunca é interrompida, o testador está a procura de quebras de progressão que ocorrem durante a partida. Como por exemplo, não conseguir abrir uma porta que leva à próxima área, essa incapacidade do progresso do personagem se caracteriza como uma quebra de progressão, é esse tipo de problema que o testador observa (LEVY & NOVAK, 2010, p. 148).

5.2.3.9 Teste de Compatibilidade

O teste de compatibilidade visa verificar diferentes combinações de sistemas operacionais, processadores e placas gráficas para determinar o requisito mínimo do sistema para que se possa jogar (CHANDLER, 2020, p. 248).

5.2.3.10 Teste de Conformidade

O teste de conformidade como já mencionado anteriormente, faz parte do processo de teste de certificação exercido pelos fabricantes de *hardware* proprietárias (como por exemplo Sony, Microsoft, Nintendo), seguindo as diretrizes da empresa (CHANDLER, 2020, p. 248; LEVY & NOVAK, 2010, p. 61).

5.2.3.11 Teste de Localização

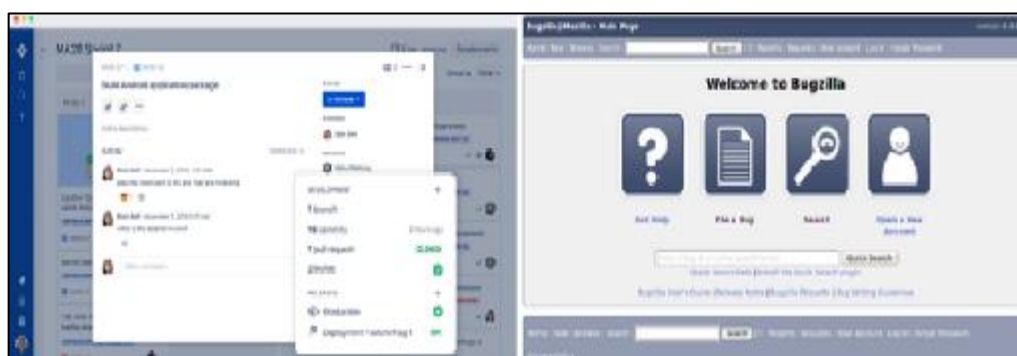
O teste de localização verifica as traduções das versões na busca dos possíveis erros de tradução (CHANDLER, 2020, p. 248).

5.3 Ferramentas usadas pela equipe de QA

Chandler (2020, p.252) comenta que antes do início do teste, as equipes de QA e desenvolvimento determinam como será feito o procedimento de trabalho *bugs* encontrados. Tem que ser estabelecido um processo claro de como será o relato, rastreamento e resolução do *bug*. Para aumentar a eficiência na rastreabilidade do *bug*, deve ser utilizados banco de dados centralizado e recomenda o uso dos *softwares* JIRA ou Bugzilla. Ambos os *softwares* oferecem uma funcionalidade robusta no rastreamento e inserção de *bugs*.

Além disso, oferecem um acesso fácil aos integrantes da equipe, interface amigável, uma comunicação rápida entre os envolvidos na tarefa e atualizações constantes a cada novo evento.

Figura - Software JIRA e Bugzilla



Fonte: Página de internet¹³

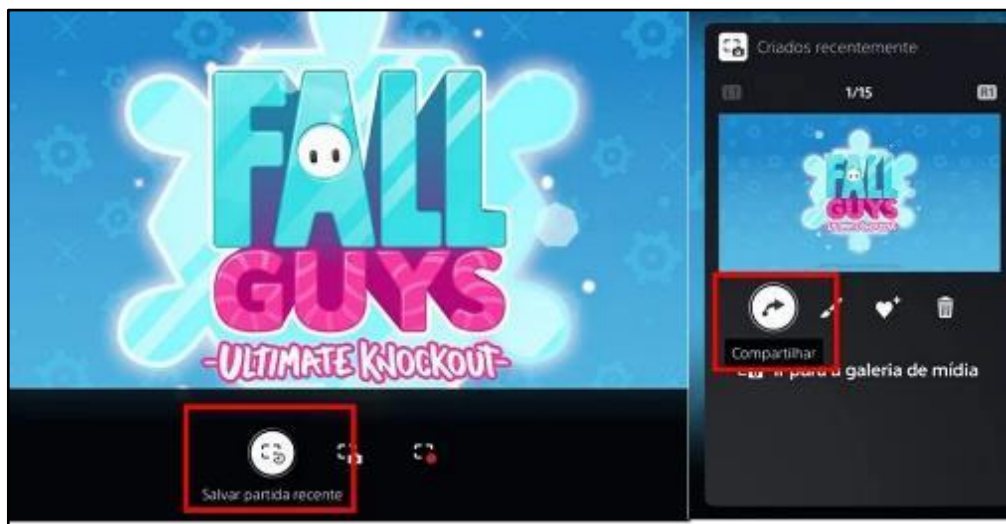
Outro método que corrobora com o trabalho do testador é o uso de um *software* de captura de vídeo, é o modo mais fácil de registrar e indicar aos desenvolvedores onde

¹³ Disponível em: Bugzilla-<https://pt.wikipedia.org/wiki/Bugzilla> e JIRA - <https://www.atlassian.com/br/software/jira/features> - Acesso em: 18/10/2020

o problema está ocorrendo. Para Levy e Novak (2010, p. 126-127) menciona que a captura de vídeo cobre todas as etapas do relatório de *bug* e substitui as várias capturas de tela que o testador teria que fazer para colocar no seu relatório, e claro evidenciar o momento do *bug*.

Os consoles da nona geração (PS5, Xbox Series X/S, Nintendo Switch) permitem através do controle que seja feita a captura de vídeo e tela com uma alta resolução diretamente do console. O compartilhamento da gravação e da tela pode ser feito diretamente do console para redes sociais e plataformas de conteúdo (MICROSOFT, 2020; NITENDO, 2020; SONY, 2020). É uma possibilidade para o registro de bugs encontrados, mesmo pós-lançamento do jogo, pode se tornar uma boa ferramenta de trabalho.

Figura 5 - Captura de tela e vídeo console PS5



Fonte: Canaltech, 2021¹⁴

5.4 Playability Testing

O teste de jogabilidade está focado em saber quanto bom é o jogo. Esse questionamento do quanto é esse “bom”, implica em uma natureza subjetiva que nos

¹⁴ Disponível em: <https://canaltech.com.br/consoles/como-salvar-prints-e-videos-do-playstation-5/> acesso em: 21/01/2021

leva a outras perguntas, como: O jogo é muito fácil? Muito difícil? Fácil de aprender? Os controles são intuitivos? A interface é clara e fácil de navegar?, entretanto a pergunta mais importante é aquela “O jogo é divertido?”. Ao contrário dos outros tipos de testes supracitados esse é voltado para questões de julgamento, sem dúvida um dos testes mais difíceis que possa ser feito (SCHULTZ, BRYANT & LANGDELL, 2005, p. 296).

Para Levy e Novak (2010, p. 66), os jogos precisam ser divertidos, ser funcional como esperado não é suficiente, os *designers* devem explorar este fator diversão, a única forma de garantir esse fator diversão é testar.

Este tipo de teste é quando se procura testar a experiência da diversão na sua integralidade e se os elementos específicos do jogo são divertidos. O teste deve identificar quais são os elementos na experiência não foram divertidos ou estão inadequados. A realização desse teste com frequência garante que o jogo seja divertido no início do desenvolvimento e continue no decorrer do processo de desenvolvimento. O teste de jogabilidade deve ser feito com pessoas não envolvidas no desenvolvimento do jogo e que fazem parte do público-alvo. Recomenda-se executar o teste desde o momento que o jogo se torna jogável até o estágio beta do processo de QA (IRISH, 2005, p. 266-267).

Schell (2020, p. 483-484) também menciona da necessidade selecionar as pessoas que fazem parte do seu público-alvo para execução do teste, eis algumas opções mais comuns:

- **Desenvolvedores:** como fazem parte do desenvolvimento do jogo, eles podem fornecer uma grande quantidade de *feedback* sobre aspectos importantes do jogo. A desvantagem é como estão envolvidos diretamente no seu desenvolvimento, há o comprometimento quanto as opiniões oriundas a respeito do jogo. Em contrapartida, perde-se *insights* preciosos, é necessário saber filtrar essas opiniões.
- **Amigos:** a vantagem é a disponibilidade na realização do teste e a facilidade na comunicação. A desvantagem pelo vínculo de amizade,

eles estarão predispostos a gostar do jogo e apontar só aspectos positivos ao término do teste.

- **Jogadores especialistas:** devido a experiência de jogar os mais variados tipos de jogos, podem fornecer um relato detalhado, usando a terminologia técnica e exemplos específicos de como seu jogo se compara aos jogos do mesmo tipo. Como esse tipo de jogador é uma pequena parcela do público, ele podem influenciar desfavoravelmente o jogo como desafios mais complexos e difíceis do que para o jogador médio.
- **Testadores temporários:** como nunca tiveram contato com o jogo, perceberão novos detalhes que não foram observados. A desvantagem é como o jogo é testado inúmeras vezes em muitas sessões, corre-se o risco do testador se entediar após várias partidas.

O local escolhido para a realização do teste, também tem suas particularidades, Schell (2020, p. 485-487), discorre sobre algumas opções de locais:

- **Estúdio:** o teste é realizado no local onde está sendo desenvolvido o jogo. Devido a praticidade, fornece a todos da equipe a oportunidade de observar o teste do jogo. O testador pode não se sentir completamente a vontade em um ambiente estranho.
- **Laboratório de playtest:** é o local propício e projetado para o teste, como espelhos bidirecional, câmeras. A desvantagem é o alto investimento para realização do teste.
- **Local público:** a realização do teste pode ser em shopping center, um festival de jogos, em geral não custa muito e haverá a oportunidade de ter vários testadores jogando seu jogo. Como está em um local público, os testadores podem se distrair no momento do teste. E também tem a heterogeneidade do público, há a possibilidade de não encontrar seu público-alvo para o teste do jogo.
- **Casa do testador:** você tem a oportunidade de ver seu jogo em um habitat natural, em condições reais, com a presença de amigos do testador e verificar a interação social entorno do jogo. A desvantagem

está na limitação do local, provavelmente um ou dois *designers* poderão observar o teste e o pequeno número de pessoas testando.

- **Pela internet:** temos a participação inúmeras pessoas para testar seu jogo em máquinas com diferentes configurações. Essa é uma boa opção para entender jogos MMOG. A quantidade de teste realizados se impõem sobre a qualidade do teste, apesar de muitas pessoas estarem jogando, não terá o mesmo nível de percepção senão estiver no mesmo recinto dos testadores.

O autor ainda destaca que o local onde ser feito o teste depende completamente das perguntas que seu teste está tentando responder. O autor comenta que teve um ótimo resultado para um jogo do Facebook, na etapa alfa perguntou aos jogadores em uma caixa de texto abaixo do jogo o que precisa ser melhorado naquele jogo. Obtiveram um retorno positivo dos jogadores, com isso foi possível melhorar experiência do usuário, retirar os *bugs* encontrados e aproveitar as ótimas ideias dos participantes (SCHELL, 2020, p. 487).

No planejamento do teste, certifique-se de que há uma maneira de obter algum tipo de resposta a cada pergunta da sua lista. Caso não consiga as resposta em um único teste, pense em realizar vários minitests para descobrir. Os desenvolvedores estarem presentes no momento do teste é muito mais proveitoso, ao invés de ler dados do levantamento ou assistir às gravações, mas eles devem permanecer isentos e objetivos para não afetar o teste. Dependendo do tipo de teste não se tem a necessidade de informar absolutamente nada ao testador, deixe o jogo falar por si só, principalmente se for esse o seu propósito no teste. Contudo, na maioria dos casos, é necessário dizer algo aos testadores para que tudo comece, tenha cuidado sempre com o tipo de informação a ser passada, para não comprometer o teste. Fique atento as reações dos testadores quando estiverem jogando, suas expressões faciais fornecem um grande quantidade de dados sobre o jogo, que nunca apareceriam nas entrevistas ou perguntas pós-jogo (SCHELL, 2020, p. 488-489).

Com um bom planejamento, é possível encontrar maneiras de registrar eventos importantes do jogo em cada sessão. Quanto mais informações puderem ser coletadas, mais úteis elas serão para evolução do jogo. Uma questão crucial é saber

o momento exato de interromper o jogador no meio da partida para perguntar algo sobre o jogo, acaba interrompendo a espontaneidade entre jogador e jogo. Por outro lado, anotar a pergunta e fazê-la após o término do jogo, provavelmente o jogador não vai se lembrar dessa experiência vivenciada naquele momento. Algumas técnicas têm sido utilizadas como *software* que coleta os dados com os participantes pensando em voz alta enquanto executam o teste. Na área de jogos isso pode ser complicado, pois algumas pessoas verbalizam naturalmente, enquanto outras ficam concentradas e falam muito pouco durante a partida. Deve ser apurado se essa técnica possa contribuir com o teste (SCHELL, 2020, p. 491).

Conforme descreve Schell (2020, p. 491-492), a pesquisa é uma ótima forma dos testadores responderem a perguntas simples sobre o jogo que são facilmente quantificadas. O autor orienta sobre como tirar o máximo proveito da pesquisa:

- No relatório de pesquisa inclua imagens sempre que possível, é um facilitador e faz o testador identificar as perguntas sobre elementos ou cenas do jogo.
- Prefira fazer pesquisas online economizam tempo, não coloque muitas perguntas na pesquisa.
- Utilize uma escala de cinco pontos e claramente identificada (“horrível”, “muito ruim”, “mais ou menos”, “bom”, “excelente”) em sua pesquisa, assim obterá resultados mais consistentes.
- Forneça a pesquisa imediatamente após o término do jogo, o testador vai responder facilmente, pois se lembra dos detalhes da experiência.
- Tenha alguém disponível para responder a dúvida dos testadores sobre a pesquisa.
- Não tome dados da pesquisa como uma verdade, é improvável que sua pesquisa seja verdadeiramente científica e os testadores tendem a inventar coisas quando não tem certeza.

Outra forma de mais detalhada de fazer perguntas aos testadores é a entrevista, assim se tem uma noção da sensação em relação ao jogo. Crie um roteiro de perguntas, faça entrevistas preferencialmente individualmente. Não espere que o participante do teste entenda questões mais técnicas sobre o jogo, tente utilizar uma

linguagem mais fácil para extrair as informações que você procura. Durante a entrevista, mantenha-se imparcial mesmo com uma crítica negativa (SCHELL, 2020, p. 492-493).

5.5 Ágil QA

6. DESENVOLVIMENTO DA MATRIZ DE AVALIAÇÃO

Podemos considerar que o jogo digital em seu processo de desenvolvimento até o lançamento no mercado de jogos, como um produto. E sendo um produto, para sua aceitação nesse mercado e dentro da expectativa do usuário durante a experiência na partida, deverá suprir a essas exigências com um bom produto e com a qualidade esperada. Quantas vezes já não dissemos ou ouvimos que aquele produto ou serviço não tem ou falta qualidade.

Por essa razão, a proposição desta pesquisa é a utilização de ferramentas de gestão e qualidade, que possam contribuir com uma avaliação quantitativa no desempenho do processo de produção do jogo.

6.1 Processo de Gestão e Qualidade

Em primeiro precisamos entender a diferença entre garantia de qualidade (*Quality Assurance*) e controle de qualidade (*Quality Control*)

Precisamos agora definir no processo de produção de um jogo digital, este envolve o desenvolvimento de um software, quais são os parâmetros considerados em questão da qualidade de um software.

software quality can be determined by how well the product performs the functions for which it was intended. For game software, this includes the quality of the player's experience plus how well the game features are implemented. Various activities can be performed to evaluate, measure, and improve game quality. In his book, Quality

is Free, Philip Crosby states that, well, “Quality is free.” This should be the high concept (remember that from Chapter 5?) of your quality program. If the cost of performing some quality function is not expected to produce an eventual saving, find a way to do it cheaper or better. If you can’t, then stop doing it. Pag 105 (SCHULTZ C. P., 2012)arrumar referencia Schultz e Bryant

Para realização desse produto, temos que passar pela a etapa de idealização desse jogo e conseqüentemente seu projeto. Na definição do guia PMBOK (PMI, 2017, p. 4), o “projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único”.

No pensamento de Cheng & Melo Filho (2007), a respeito da qualidade do produto:

“[...]no sentido amplo, pode ser entendida como a satisfação do cliente. A garantia dessa qualidade pode ser realizada através de três enfoques complementares, que fazem parte do ciclo de vida do produto: garantia da qualidade de inspeção, garantia da qualidade pelo controle do processo e garantia da qualidade durante o desenvolvimento do produto.”

5.2 Ferramentas de Gestão e Qualidade

5.2.1 Matriz de Pugh

Criada por Stuart Pugh, na década de 1990, esse método faz a comparação de diversos conceitos de um produto que se pretende trabalhar que são elencados através de uma matriz, assim é possível classificar e avaliar entre as opções de escolha. Basicamente consiste em estabelecer um conjunto de critérios de importância previamente definidos para aquele produto ou serviço, com isso pode-se fazer a comparação com a análise dos pontos positivos e pontos negativos e definir entre as opções de escolha qual é a mais viável. É frequentemente usado na engenharia para a tomada de decisão de *design*, mas também é utilizado em outros cenários, como opções de fornecedores, investimentos, dentre outros.

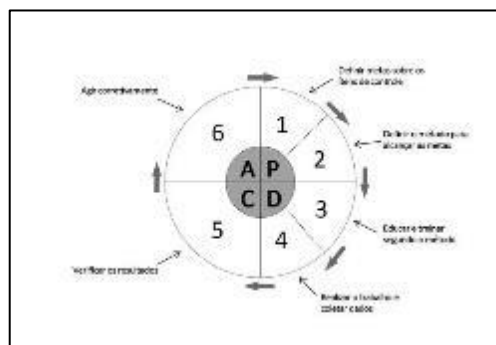
Figura 9 – Modelo de Matriz Pugh

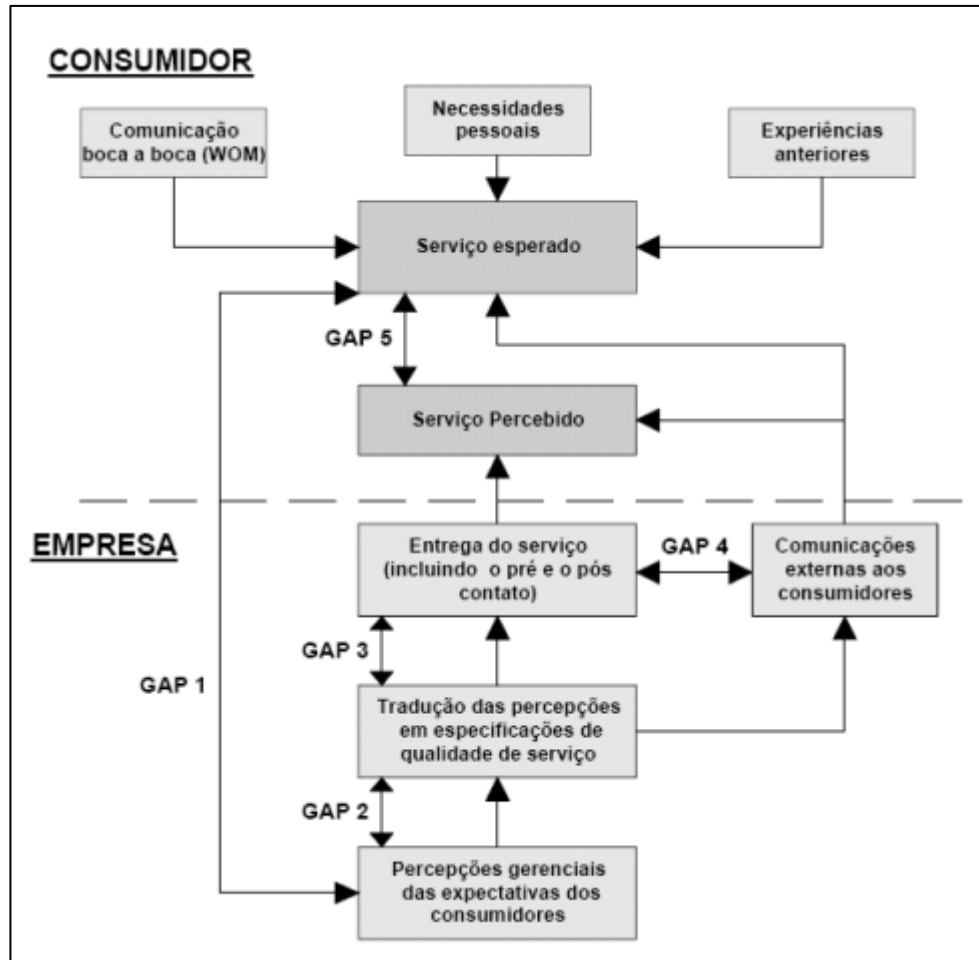


Fonte: Laboratório de Gestão UFSM (2020)

Existe outro modelo que é a evolução da primeira matriz elaborada pelo Stuart, onde temos um complexidade um pouco maior em relação a primeira matriz apresentada. Nesta matriz temos opção da aplicação do peso e nota para a lista de critérios a serem avaliadas, quanto maior é a relevância do critério, maior deverá ser o valor do seu peso.

Como visto na Figura 8 – Atividades de um game teste temoas as etapas do procedimento de análise do testador de QA, a similaridade se faz presente em uma ferramenta da área de gestão de qualidade o ciclo PDCA. O ciclo PDCA consiste na busca de melhorias contínuas aos processos de uma empresa ou de um projeto. A ferramenta é sistêmica, em ciclos de análise

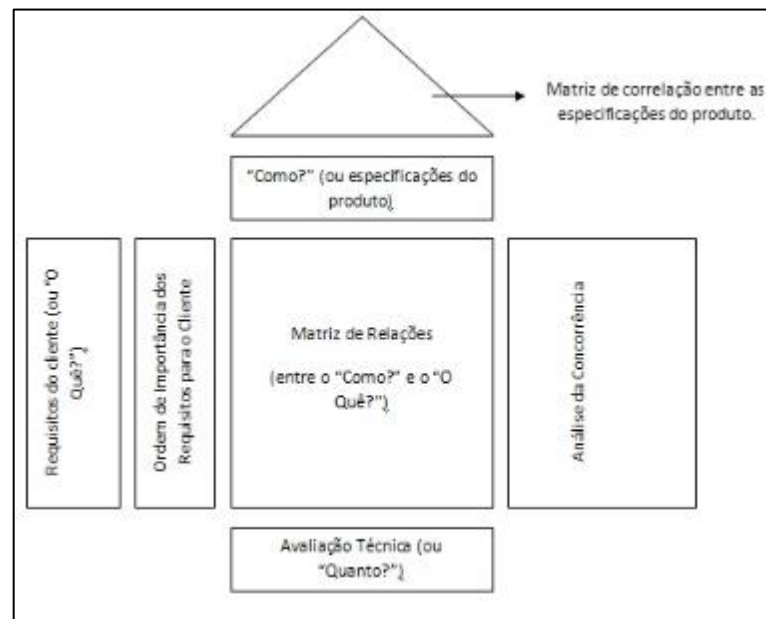




5.2.2 Matriz Quality Function Deployment (QFD)

Essa matriz foi desenvolvida como uma ferramenta de qualidade que permite que a equipe de desenvolvimento do produto compreenda as reais necessidades do cliente. Tem como objetivo organizar as ideias e com isso planejar os requisitos para melhoria do produto na forma de especificações técnicas.

Figura 10 – Modelo da Matriz QFD



Fonte: Infoescola (2020)

Nessa seção pretende-se abordar outras ferramentas de gestão e qualidade como por exemplo a gestão Lean, o indicador de desempenho KPI e o Seis Sigma.

5.3 Proposta de ferramenta de avaliação

Durante a disciplina de Análise e Modelagem de Jogos Digitais, houve o conhecimento das especificidades da Engenharia de Software, técnicas de elicitação de requisitos e modelagem com linguagem UML (*Unified Modeling Language*) aplicadas a área de jogos digitais com a utilização das metodologias ativas, no caso o PBL ¹⁵(*Problem Based Learning*). Onde tivemos a oportunidade estudar uma situação hipotética de uma *startup* com objetivo de desenvolver um jogo educacional para o Ensino Médio. Durante os encontros foram elaborados a concepção, arte, prototipagem do jogo embasada nos conceitos de modelagem de software alinhado

¹⁵ **PBL** – É um metodologia onde o aprendizado é ativo centrado no aluno e tem como estratégia educacional a busca da solução de problemas, é uma forma de aprendizado que estimula a proatividade e o aprimoramento pessoal em um grupo acadêmico por meio de discussões profundas de casos interdisciplinares. (Disponível em: <https://periodicos.fclar.unesp.br/iberoamericana/article/view/11437>. Acessado: 10/10/2020)

ao método MDA. A cada artefato produzido durante o semestre foi possível conceber a documentação GDD¹⁶ (*Game Design Document*).

Diante ao exposto, durante os encontros um dos tópicos estudados foi a questão do *level design*. Com o emprego das metodologias ativas durante o semestre acadêmico, foi realizada a proposição de uma matriz de tomada de decisão aliados aos conceitos de *game design*.

Para Byrne (2005, p.6), o termo *level design* é originário do RPG (*Role-playing Game*) e antigas máquinas de *Arcade* sendo definido como o mapa, missão ou nível composto de eventos com incrementos de dificuldade que suceder-se-ão no jogo durante a partida.

Nessa etapa são constituídos os elementos da metodologia MDA e o *game designer* pode valer-se do método na construção desse mundo, nesse quesito o desenvolvimento dos jogos digitais tem se pautado em novas mecânicas, jogabilidade e narrativas que torna a experiência do usuário de uma forma mais imersiva, interativa e com uma melhor assimilação das regras estabelecidas na composição do jogo.

Para Braga (2018), esses elementos “[...] espaço, seus limites e as mecânicas que são executadas no seu interior dão forma ao *Level* do jogo”. E ainda, o jogador realiza essa ação dentro do jogo com o intuito de atingir um resultado específico e desejado.

Tratando-se da construção do *level* do jogo, temos os artigos de Dan Taylor¹⁷, onde ele apresenta os dez princípios para composição desse *level* e norteadores para o *level design*.

¹⁶ **GDD** - É o registro de todos os elementos que engendraram o projeto de um game. É o documento onde encontramos todo o detalhamento da equipe de desenvolvimento (artistas, programadores, designers) que vai desde a arte conceitual (temática, mecânicas, jogabilidade, trilha sonora) e a parte tecnológica (arquitetura do sistema, hardware, software), onde guiará as equipes na produção do jogo.

¹⁷ Disponível em: <https://www.gamasutra.com/blogs/author/DanTaylor/543613/> - Acessado em 04/10/2020

[...]

- 1) **É divertido de explorar** - use uma linguagem visual clara para guiar o jogador ao longo do caminho principal e crie interesse através da verticalidade, caminhos secundários, áreas escondidas e elementos de labirinto.
- 2) **Não depender de palavras para contar uma história** - Além da narrativa explícita produzida pela história e pelos objetivos, o bom *level design* oferece uma narrativa implícita através do ambiente e fornece aos jogadores uma mecânica de escolhas para que possam criar sua própria narrativa emergente.
- 3) **Diz ao jogador o que fazer, mas não como fazê-lo** - Garante que os objetivos da missão sejam claramente comunicados, mas permite que os jogadores os concluam da maneira que quiserem e, quando viável, em qualquer ordem.
- 4) **Constantemente ensina ao jogador algo novo** - Mantêm o jogador engajado, introduzindo novas mecânicas durante o jogo, e impede que as mecânicas já aprendidas se tornem obsoletas ao aplicar modificadores ou reutilizá-los de maneiras inusitadas.
- 5) **Ser surpreendente** – Um bom *level design* não tem medo de assumir riscos com o ritmo, a estética, localidade e outros elementos para criar uma nova experiência.
- 6) **Capacitar o jogador** - Videogames são escapistas e, como tal, devem abster-se do mundano. Além disso, um bom *level design* reforça a capacidade dos jogadores, permitindo que eles experimentem as consequências de suas ações, seja de forma imediata, de momento em momento, quanto a longo prazo, por meio do design de todos os níveis.
- 7) **Permite ao jogador controlar a dificuldade** - Prepara o caminho principal para jogadores iniciantes, apresentando aos jogadores avançados com desafio opcional por meio de oportunidades claramente comunicadas de risco e recompensa.
- 8) **Ser eficiente** - Recursos são limitados. Um bom *level design* cria eficiência por meio da modularidade, jogabilidade bidirecional, objetivos integrados e bônus exploratórios que fazem uso de todo o espaço jogável.
- 9) **Cria emoção** – Comece pelo fim, com a resposta emocional desejada, e trabalhe no sentido reverso, selecionando as mecânicas apropriadas, métricas espaciais e dispositivos de narrativa para se obter essa resposta.
- 10) **É movido pela mecânica do jogo** - Acima de tudo, mostra a mecânica do jogo por meio dos níveis, para reforçar a natureza interativa única dos videogames (Dan Taylor, 2013, tradução nossa).

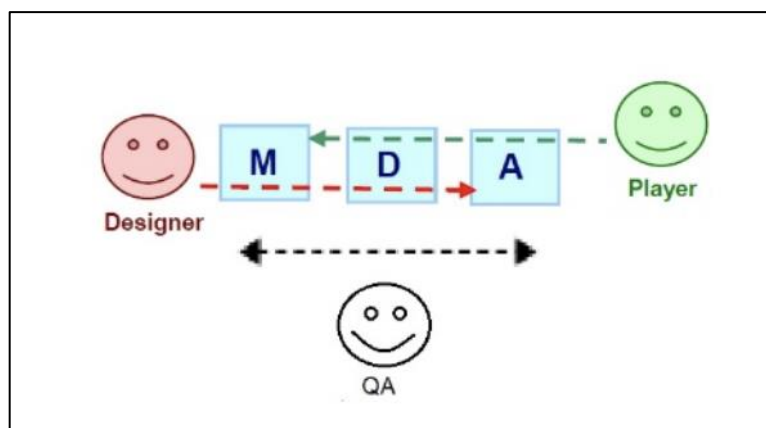
Conforme abordado no processo de modelagem MDA, temos as perspectivas do ponto de vista do designer e do ponto de vista do jogador, e se faz necessário analisar as perspectivas desses dois elementos para melhorar o desenvolvimento do jogo, a fim de um resultado promissor na entrega do produto final.

Nesse contexto, podemos acrescentar uma nova perspectiva do terceiro elemento, o ponto de vista do departamento de Garantia de Qualidade (*Quality Assurance* – QA). Segundo Chandler (2012), a principal responsabilidade do departamento de QA é criar o plano de testes para o jogo e validação do mesmo. A função primordial do QA é executar o *playtest* do jogo e dar *feedback* aos desenvolvedores dos possíveis *bugs* encontrados durante o processo de análise.

É importante ressaltar que a equipe de QA pode oferecer às equipes de *design* e desenvolvimento uma nova perspectiva sobre um jogo, através dos olhos de quem não está familiarizado com a mecânica ou história do jogo. Eles agem como uma defesa pré-lançamento contra as frustrações e confusões que poderiam ser encontradas pelos consumidores pós-lançamento (RUGGILL, MCALLISTER, NICHOLS & KAUFMAN, 2017, p. 214, tradução nossa).¹⁸

O QA consegue avaliar e apontar as necessidades relevantes para melhoria no desenvolvimento do projeto de uma forma técnica concomitantemente com o olhar subjetivo de um jogador, e nisso, temos o refinamento do sistema antes da produção do jogo.

Figura 11 – Perspectiva do QA em relação ao Framework MDA



Fonte: Modificado Hunicke, LeBlanc & Zubek (2004, p. 2)

¹⁸ “Importantly, QA personnel can offer the design and development teams a new perspective on a game, through the eyes of those unfamiliar with the game’s mechanics or story. They act as a pre-release defense against the frustrations and confusions that might otherwise be encountered by post-release consumers.”

Com esse entendimento, emerge a proposição da elaboração de uma matriz de avaliação como ferramenta de apoio na análise de cada *level*, com a finalidade de se ter um parâmetro norteador na melhoria continuada para o desenvolvimento do jogo. Para elaboração dessa matriz utilizamos a planilha eletrônica em Excel, a sua concepção foi realizada conforme apresentado na Tabela 3, a saber:

Tabela 4 – Matriz de avaliação (DMC)

1	2	3	4 FASE				5
			1ª	2ª	3ª	4ª	
		PESO	NOTA	NOTA	NOTA	NOTA	PONTUAÇÃO MÁXIMA
Dinâmicas	Emoção	1	5	5	5	1	7
	Narrativa	3	4	3	2	6	21
	Progressão	1	5	2	1	2	7
	Relacionamento	1	6	3	6	5	7
	Restrições	1	7	2	5	7	7
Mecânicas	Aquisição de recursos	3	5	4	3	2	21
	Feedback	5	5	3	4	5	35
	Competição	3	7	2	2	1	21
	Desafios	1	6	5	3	6	7
	Recompensas	3	5	4	2	5	21
	Transações	1	4	5	5	3	7
	Turnos	1	6	1	1	2	7
Componentes	Boss	5	6	2	1	2	35
	Itens	3	4	6	2	4	21
	Combate	5	3	2	3	3	35
	Missão	5	5	3	5	2	35
	Level Up	3	4	1	2	1	21
	Formação de Times	1	3	3	4	5	7
	Música	3	2	4	3	4	21
	Efeitos sonoros	3	3	3	4	3	21
	Interface	3	4	3	3	6	21
6		Soma	251	166	164	187	385
		%	65,19%	43,12%	42,60%	48,57%	

Fonte: Produzido pelo autor

- 1) Na matriz dividimos em três grupos para avaliação, sendo: Dinâmicas, Mecânicas e Componentes do jogo para que seja feita a análise da etapa de desenvolvimento.
- 2) Elencamos nas linhas os elementos do jogo definido para cada grupo.
- 3) Na coluna PESO são atribuídos os valores de peso (0,1,3 ou 5), sendo: 0 – Não se aplica; 1 – Pouca importância; 3 – Relativamente importante; 5 –

Extremamente importante. Este valores são definidos pelos pares da equipe de desenvolvimento a partir do critério de importância, a ser atribuído a cada um dos itens da respectiva linha.

- 4) Na coluna da FASE, refere-se a cada um dos *level* em desenvolvimento, onde são atribuídos os valores da nota para cada um dos itens da respectiva linha em escala de 1 a 7, onde a nota 1 a menos significativa e nota 7 a maior nota.
- 5) Na coluna PONTUAÇÃO MÁXIMA, são os valores máximos alcançados para aquele item pela multiplicação do peso com a nota atribuída (Ex: Pontuação máxima = valor do peso x nota máxima do item).
- 6) Somatório das notas atribuídas de cada coluna pela multiplicação dos seus respectivos pesos. A porcentagem apresentada de cada coluna (*level*) em relação a pontuação máxima estabelecida.

Tabela 5 – Valores dos pesos e nota a serem atribuídos na matriz de avaliação

PESOS	
0	Não se aplica (N/A)
1	Pouca importância
3	Relativamente importante
5	Extremamente importante

NOTA	
1 a 7	Valor atribuído para o item

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Fonte: Produzido pelo autor

Foi elaborada uma segunda matriz sob a perspectiva dos artigos de Dan Taylor “*Ten Principles of Good Level Design*”, onde elencamos na linhas os dez princípios de um bom *level design* e em cada coluna foram dispostos os itens (Peso, Fase, Pontuação Máxima) da mesma forma supracitada com a primeira matriz.

Tabela 6 – Matriz de avaliação princípios do bom level design (PGLD)

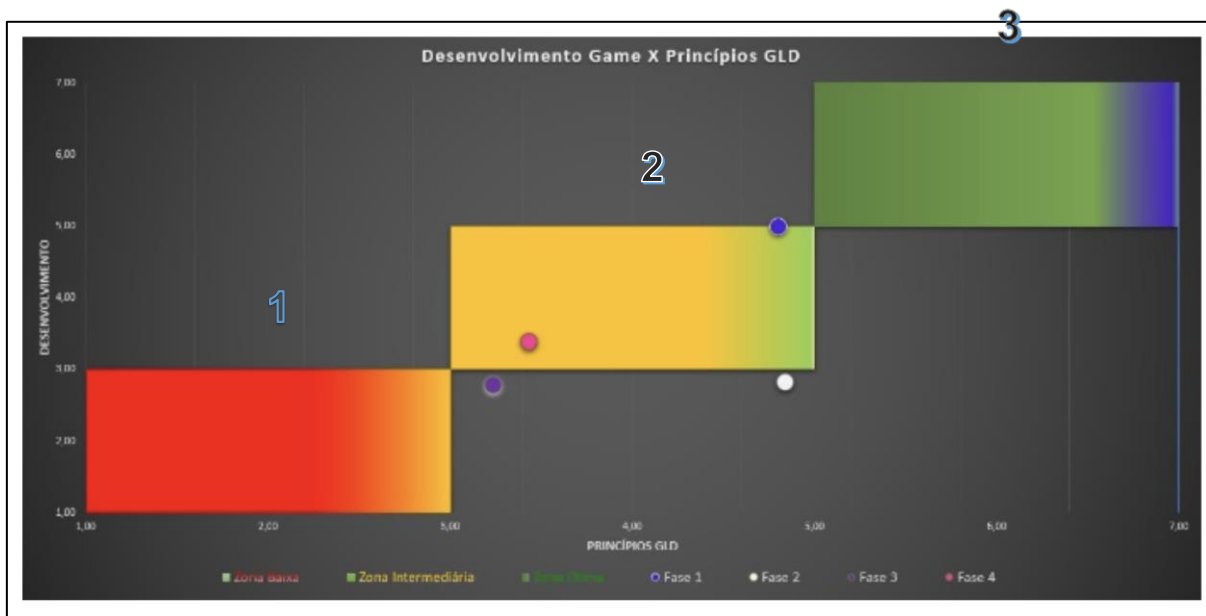
			FASE				PONTUAÇÃO O MÁXIMA
			1ª	2ª	3ª	4ª	
Princípios			PESO	NOTA	NOTA	NOTA	NOTA
1	É divertido de explorar	1	2	3	2	2	7
2	Não depender de palavras para contar uma história	3	3	5	3	3	21
3	Dizer ao jogador o que fazer, mas não como fazer	5	4	4	5	4	35
4	Ensinar constantemente algo novo ao jogador	3	5	6	6	5	21
5	Ser surpreendente	1	7	7	7	6	7
6	Capacitar o jogador	5	6	5	2	5	35
7	Permitir ao jogador controlar a dificuldade	3	5	3	4	4	21
8	Ser eficiente	1	3	4	3	3	7
9	Criar emoção	3	6	3	2	2	21
10	É movido pela mecânica do jogo	5	5	7	1	1	35
Soma			144	145	97	103	210
%			68,57%	69,05%	46,19%	49,05%	

Fonte: Produzido pelo autor

Como forma de contextualizar o processo, na matriz da tabela 3 distribuímos os pesos nos itens elencados nos elementos do jogo dos grupos Dinâmicas, Mecânicas e Componentes. Em seguida, solicitamos ao time de QA uma análise de cada um dos *levels* em desenvolvimento e com o critério de notas foram atribuídas notas de 1 a 7 para cada um dos itens. E na matriz da tabela 5, com base nos princípios do bom *level design*, foram atribuídos os pesos para cada um dos itens e solicitamos ao time de QA o mesmo procedimento para validação das notas.

Com as notas atribuídas conseguimos classificar e visualmente observar a avaliação do time de QA. Assim, verifica-se através de um critério quantitativo cada um dos itens de cada fase em desenvolvimento. Com os critérios avaliados, foi gerado um gráfico do produto cartesiano entre a matriz DMC e a matriz PGLD, com a escala das notas de 1 a 7 nos eixos da abscissa e ordenada. E determinamos três áreas de categoria desse gráfico, sendo a primeira zona baixa, a segunda zona intermediária e a terceira zona ótima. Para sabermos em qual faixa aquele *level* se encontra, foi feita a média ponderada de cada uma das fases de ambas as matrizes e com isso encontramos a localização do ponto que representa aquele *level* analisado.

Figura 12 – Gráfico DMC x PGLD



Fonte: Produzido pelo autor

Assim, podemos visualizar e auxiliar na tomada de decisão os pontos a serem melhorados, tanto pelas tabelas visualmente com a escala de cor e a nota atribuída, como pelo gráfico com os pontos dentro de cada área mencionada (zona baixa, zona intermediária, zona ótima). Outra forma de utilização desta ferramenta é a tomada de decisão para a escolha entre protótipos de alta fidelidade.

6. DISCUSSÃO DOS RESULTADOS

7. CONSIDERAÇÕES FINAIS

REFERÊNCIAS

- ADAMS, E. (2003). *Break into the Game Industry: How to Get A Job Making Video Games*. Emeryville,, CA, Estados Unidos da América: McGraw-Hill/Osborne.
- ADAMS, E. (2010). *FUNDAMENTALS OF GAME DESIGN* (2 ed.). CA, berkeley, Estados Unidos da América: New Riders.
- BERNE, E. (s.d.). *Games People Play - The psychology of human relationships*.
- BERTHKE, E. (2003). *Game development and production*. Plano, Texas, Estados Unidos da América: Wordware Publishing, Inc.
- BUG. (2020). In:Infopedia,Dicionário Porto Editora. Acesso em 04 de 11 de 2020, disponível em <https://www.infopedia.pt/>
- BUG. (2020). In:Michaelis, Dicionário Basrileiro da Língua Portuguesa. Acesso em 04 de 11 de 2020, disponível em <https://michaelis.uol.com.br/busca?r=0&f=0&t=0&palavra=bug>
- BURGUN, K. (2013). *Gane Design Theory: A New Philosophy for Understanding Games*. Boca Raton, FL, Estados Unidos: CRC Press.
- CHANDLER, H. M. (2012). *Manual de produção de jogos digitais* (2 ed.). Porto Alegre: Bookman.
- CHANDLER, H. M. (2020). *The Game Production Toolbox*. Boca Raton, FL, Estados Unidos da América: CRC Press.
- CHENG, L. C., & MELO FILHO, L. D. (2007). QFD na garantia da qualidade do produto durante seu desenvolvimento – caso em uma empresa de materiais. *Produção*, 604-624.
- COHEN, D., BUSTAMANTE, & S PARK, T. (2010). *Producing Games:From Business and Budgets to Creativity and Design*. Burlington, MA, Estados Unidos da América: Focal Press.
- CROCI, F. (25 de 10 de 2015). https://www.tilt.it/flipper_pinball/2015/10/happy-flipper-day/. Acesso em 11 de 09 de 2020, disponível em TILT.IT: https://www.tilt.it/flipper_pinball/2015/10/happy-flipper-day/
- DEMARIA, R. (2019). *High Score! Expanded The Illustrated History of Eletronic Games* (3ª ed.). Boca Raton, FL, Estados Unidos: CRC Press.

- DRESCH, A., LACERDA, D. P., & ANTUNES JUNIOR, J. A. (2015). *Design science research: método de pesquisa para avanço da ciência e tecnologia*. Porto Alegre, RS, Brasil: Bookman.
- FLYNT, J. P., & SALEM, O. (2005). *Software Engineering for Game Developers*. Boston, MA, Estados Unidos da América: Thomson/Course Technology.
- FULLERTON, T. (2019). *Game Design Workshop - A Playcentric Approach to Creating Innovative Games* (4 ed.).
- GIL, A. C. (2002). *Como elaborar projetos de pesquisa* (4ª ed.). São Paulo, SP, Brasil: Atlas.
- HIWILLER, Z. (2016). *Players Making Decisions: Game Design Essentials and the Art of Understanding Your Players*. New Riders.
- IRISH, D. (2005). *The Game Producer's Handbook*. Boston, MA, Estados Unidos da América: Thomson Course Technology.
- IUPPA, N. V., & BORST, T. (2010). *End-to-end game development: creating independent serious games and simulations from start to finish*. Burlington, MA, Estados Unidos da América: Focal Press.
- JUUL, J. (2005). *Half-Real: Video Games Between real rules*. Cambridge, MA: MIT Press.
- KANJI, G. k. (1996). *100 METHODS FOR TOTAL QUALITY MANAGEMENT*. Sage Publications.
- KEITH, C. (2010). *Agile game Development with Scrum*. Boston, MA, Estados Unidos: Pearson Education, Inc.
- LARSEN, K. R. (2020). Validity in Design Science Research. *International Conference on Design Science Research in Information Systems and Technology (DESRIST 2020)* (p. 11). ResearchGate.
- LEMES, D. (2019). *Estudos sobre o desenvolvimento de games*. São Bernardo do Campo, SP: C0D3S.
- LEMES, D. O. (2018). *"Level Design, Jogabilidade e Narrativa para Games-* (eBook ed.). São Bernardo do Campo: COD3S.
- LEVY, L., & NOVAK, J. (2010). *Game Development Essentials: Game QA & Testing*. Clifton Park, Nova York, Estados Unidos da América: Delmar.
- LOVELL, N. (2019). *The Pyramid of Game Design*. Boca Raton, FL, Estados Unidos: CRC Press.

- MENCHER, M. (2003). *Get in the Game! CAREERS IN THE GAME INDUSTRY*. Indianapolis, Indiana, Estados Unidos da América: New Riders Publishing.
- NOVAK, J. (2017). *Desenvolvimento de games* (2 ed.). (P. C. Conti, Trad.) São Paulo, SP, Brasil: Cengage Learning.
- ON LINE EDITORA. (2016). Guia Conheça a História. *A Vida no Antigo Egito*, 33-37.
- PERUCIA, A., & AL, E. (01 de 2011). Coordenação das atividades produtivas na indústria brasileira de jogos eletrônicos: hierarquia, mercado ou aliança? *Produção*, 21, pp. 64-75. doi:10.1590/S0103-65132010005000046
- PRODANOV, C. C., & FREITAS, E. C. (2013). *Metodologia do trabalho científico:métodos e técnicas da pesquisa e do trabalho acadêmico* (2ª ed.). Novo Hamburgo, RS, Brasil: Feevale.
- ROGERS, S. (2014). *Level Up!The guide to great video game design* (2 ed.). Wiley.
- RUGGILL, J. E., MCALLISTER, K. S., NICHOLS, R., & KAUFMAN, R. (2017). Nova York, NY, Estados Unidos da América: Routledge.
- SAKUDA, L. O., & FORTIM, I. (2018). *II Censo da Indústria Brasileira de Jogos Digitais*. Ministério da Cultura, Brasília. Fonte: <<http://www.tinyurl.com/censojogosdigitais>>
- SALEN, K., & ZIMMERMAN, E. (2004). *Rules of Play: Game Design Fundamentals*. Cambridge, MA: MIT Press.
- SCHELL, J. (2011). *A arte de game design*. (E. Furmankiewicz, Trad.) Rio de Janeiro, RJ, Brasil: Elsevier.
- SCHELL, J. (2020). *The art of game design : a book of lenses* (3ª ed.). Boca Raton, FL, Estados Unidos da América: CRC Press.
- SCHULTZ, C. P. (2012). *Game Testing All in One* (2 ed.). Boston, Massachusetts, Estados Unidos: MERCURY LEARNING AND INFORMATION.
- SCHULTZ, C. P., & BRYANT, R. D. (2012). *GAME TESTING: All In One* (2ª ed.). Dulles, VA, Estados Unidos da América: Mercury Learning and Informartion.
- SCHULTZ, C., BRYANT, R., & LANGDELL, T. (2005). *Game Testing All in One*. Boston, Massachusetts, Estados Unidos: Thomson Course Technology PTR.
- SYED, M. (2019). *Quality Control in Games Development*. Laurea University of Applied Sciences, Business Information Technology.
- WILLIAMS, A. (2017). *History of digital games: developments in art, design and interaction*. Boca Raton, FL, Estados Unidos: CRC Press.

ZANELLA, L. C. (2013). *Metodologia de pesquisa* (2ª ed.). Florianópolis, SC, Brasil: UFSC,.

SONY PS5 <https://www.playstation.com/pt-br/accessories/dualsense-wireless-controller/>

MICROSOFT <https://www.xbox.com/pt-BR/accessories/controllers/xbox-wireless-controller>

NINTENDO <https://www.nintendo.pt/Familia-Nintendo-Switch/Acessorios/Acessorios-1183144.html>

GLOSSÁRIO

ANEXOS

ANEXO – 01

Game Design Document

ANEXO – 02

Relação de Assets

ANEXO – 03

Modelagem UML

ANEXO – 04

Mapa de orientação

ANEXO – 05

Mapeamento bibliográfico

ANEXO – 06

Infográficos

Infográficos

ANEXO – 07

Mapas mentais

Índice de autores

Índice remissivo