




Universidade Federal de Pernambuco

Graduação em Ciência da Computação
Centro de Informática

2006.1



Um Modelo Preditivo para Desenvolvimento de Jogos de Computador

Trabalho de Graduação

Aluno – Gustavo Henrique Porto de Carvalho, ghpc@cin.ufpe.br.

Orientador – Hermano Perrelli de Moura, hermano@cin.ufpe.br.

Outubro de 2006

*We must do a few things to survive.
Everything else is entertainment*
Marvin Minsky

RESUMO

O desenvolvimento de jogos eletrônicos mudou bastante desde suas origens até os dias atuais. É preciso lidar com times heterogêneos e, em algumas circunstâncias, com projetos de duração superior a alguns anos. Cada vez é mais importante utilizar modelos de desenvolvimento que minimizem problemas durante a criação de um jogo. Os modelos de desenvolvimento de software existentes não são facilmente aplicáveis à realidade da indústria de jogos. Estes não levam em consideração etapas, atividades, papéis e artefatos específicos do mercado de jogos. Este trabalho de graduação propõe um modelo preditivo de desenvolvimento de jogos de computador inspirado na essência do *Rational Unified Process* (RUP). Este modelo, denominado MDJ-PC, pode ser utilizado por pessoas, academias e organizações, desde que estes o adaptem as suas realidades. De forma análoga ao RUP, o MDJ-PC é composto por 4 fases (Concepção, Pré-Produção, Produção e Pós-Produção) e um conjunto de 12 disciplinas: 9 de criação e 3 de apoio.

Palavras-chave: entretenimento digital, jogos de computador, engenharia de *software* e modelos de desenvolvimento de *software*.

ABSTRACT

The development of electronic games markedly changed from its beginnings to nowadays. It is now necessary to deal with heterogeneous teams and, under some circumstances, with projects lasting years. Each time it is more important to employ development models that minimize problems during the game creation. The existing software development models are not easily applicable to the actual state of game industry. They do not take in consideration steps, activities, roles and specific artifacts in the game market. This work proposes a predictive model for developing computer games essentially based on Rational Unified Process (RUP). This model, called MDJ-PC, can be used for people, academies and organizations, provided that they adapt it to their realities. Similarly to RUP, the MDJ-PC is composed of 4 phases (Conception, Pre-Production, Production and Post-Production) and a set of 12 disciplines: 9 of creation and 3 of support.

Key words: digital entertainment, computer games, software engineering and software development models.

Dedicatória

*Dedico este trabalho à memória
de um exemplo de determinação e luta,
Vovó Carmen.*

Agradecimentos

*A Ione e Luiz, meus pais,
pelo apoio e amor incondicional.*

*A Aninha, minha namorada,
por me fazer uma pessoa tão feliz.*

*A todos os meus colegas, amigos e amigas,
pelas dificuldades e alegrias compartilhadas.*

*A Hermano, meu orientador,
pelos ensinamentos durante a graduação.*

*A Katia, professora do Cin,
pela monitoria de Algoritmos, uma das
experiências mais gratificantes da graduação.*

*A Patrícia, professora do Cin,
pela confiança e orientação
durante minha iniciação científica.*

*A Fábio, professor do CIn,
pelo exemplo e inspiração de como
empreender pelos os seus sonhos.*

*Por fim, a todos os que formam
o Centro de Informática,
pela minha formação.*

SUMÁRIO

1. INTRODUÇÃO	3
1.1 CONTEXTO ATUAL DO DESENVOLVIMENTO DE JOGOS	3
1.2 DESAFIOS DO CENÁRIO ATUAL	3
1.3 OBJETIVOS DO TRABALHO DE GRADUAÇÃO.....	3
1.3.1 <i>Segmento PC</i>	3
1.3.2 <i>Adaptativo vs. Preditivo</i>	3
1.4 RELEVÂNCIA DO TRABALHO	3
1.5 METODOLOGIA APLICADA AO TRABALHO DE GRADUAÇÃO.....	3
1.6 ORGANIZAÇÃO DO TRABALHO	3
2. JOGOS ELETRÔNICOS	3
2.1 INTRODUÇÃO.....	3
2.1.1 <i>Gêneros de Jogos Digitais</i>	3
2.2 MÍDIAS DE JOGOS DIGITAIS	3
2.3 EVOLUÇÃO DOS JOGOS DIGITAIS	3
2.4 O MERCADO DE JOGOS DIGITAIS.....	3
2.4.1 <i>Internacional</i>	3
2.4.2 <i>Nacional</i>	3
2.5 MUDANÇAS NO DESENVOLVIMENTO DE JOGOS.....	3
3. CICLO DE PRODUÇÃO DE JOGOS PARA PC	3
3.1 INTRODUÇÃO.....	3
3.1.1 <i>Cadeia de Valor de Jogos para Plataforma PC</i>	3
3.2 CICLO DE DESENVOLVIMENTO DE JOGOS PARA PC	3
3.2.1 <i>Concepção</i>	3
3.2.2 <i>Pré-Produção</i>	3
3.2.3 <i>Produção</i>	3
3.2.4 <i>Pós-Produção</i>	3
3.2.5 <i>Pós-Lançamento</i>	3
3.3 PRINCIPAIS PAPÉIS NO DESENVOLVIMENTO DE JOGOS	3
3.4 PRINCIPAIS ATIVIDADES DESEMPENHADAS.....	3
3.4.1 <i>Produtor</i>	3
3.4.2 <i>Artista</i>	3
3.4.3 <i>Game Designer</i>	3
3.4.4 <i>Programador</i>	3
3.4.5 <i>Engenheiro de Áudio</i>	3
3.4.6 <i>Engenheiro de Qualidade</i>	3
3.5 DESAFIOS DO CICLO DE PRODUÇÃO DE JOGOS DE COMPUTADOR.....	3
4. METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE	3
4.1 INTRODUÇÃO.....	3
4.1.1 <i>Correntes de Modelos de Desenvolvimento de Software</i>	3
4.2 METODOLOGIAS ÁGEIS	3
4.2.1 <i>Extreme Programming (XP)</i>	3
4.2.2 <i>Extreme Game Development (XGD)</i>	3
4.3 METODOLOGIAS PREDITIVAS	3
4.3.1 <i>Game Waterfall Process (GWP)</i>	3
4.3.2 <i>Rational Unified Process (RUP)</i>	3
4.3.3 <i>Game Unified Process (GUP)</i>	3
4.4 ANÁLISE COMPARATIVA	3
5. MODELO PREDITIVO DE DESENVOLVIMENTO DE JOGOS PC	3

5.1	INTRODUÇÃO.....	3
5.1.1	<i>Práticas.....</i>	3
5.1.2	<i>Características</i>	3
5.2	VISÃO GERAL.....	3
5.2.1	<i>Especificação.....</i>	3
5.3	ESTRUTURA DINÂMICA – FASES.....	3
5.3.1	<i>Concepção</i>	3
5.3.2	<i>Pré-Produção</i>	3
5.3.3	<i>Produção</i>	3
5.3.4	<i>Pós-Produção</i>	3
5.4	ESTRUTURA ESTÁTICA – DISCIPLINAS.....	3
5.4.1	<i>Modelagem de Mercado</i>	3
5.4.2	<i>Game Design</i>	3
5.4.3	<i>Technical Design</i>	3
5.4.4	<i>Implementação</i>	3
5.4.5	<i>Criação de Arte 2D.....</i>	3
5.4.6	<i>Criação de Arte 3D.....</i>	3
5.4.7	<i>Criação de Áudio.....</i>	3
5.4.8	<i>Teste.....</i>	3
5.4.9	<i>Distribuição.....</i>	3
5.4.10	<i>Gerência de Configuração e Mudanças.....</i>	3
5.4.11	<i>Gerência de Projeto.....</i>	3
5.4.12	<i>Ambiente</i>	3
6.	CONCLUSÕES.....	3
6.1	CONTRIBUIÇÕES	3
6.2	TRABALHOS FUTUROS.....	3
6.3	CONSIDERAÇÕES FINAIS	3
	REFERÊNCIAS	3

LISTA DE FIGURAS

Figura 1 – <i>Halo 2</i>	3
Figura 2 – <i>Indigo Prophecy</i>	3
Figura 3 – <i>Rayman 3</i>	3
Figura 4 – <i>Starcraft Ghost</i>	3
Figura 5 – <i>Super Mario 3</i>	3
Figura 6 – <i>Tekken 5</i>	3
Figura 7 – <i>Ghost Recon 3</i>	3
Figura 8 – <i>Warcraft 3</i>	3
Figura 9 – <i>Civilization IV</i>	3
Figura 10 – <i>Final Fantasy XIII</i>	3
Figura 11 – <i>World of Warcraft</i>	3
Figura 12 – <i>Splinter Cell Double Agent</i>	3
Figura 13 – <i>Silent Hill 3</i>	3
Figura 14 – <i>Flight Simulator 2004</i>	3
Figura 15 – <i>Project Gotham Race 3</i>	3
Figura 16 – <i>FIFA 2006</i>	3
Figura 17 – <i>DDR</i>	3
Figura 18 – <i>Bejeweled</i>	3
Figura 19 – <i>Mario Party 7</i>	3
Figura 20 – <i>Chessmaster 10000</i>	3
Figura 21 – <i>Carmen Sandiego</i>	3
Figura 22 – <i>Worky's Challenge</i>	3
Figura 23 – <i>Clio Xtrem Racer</i>	3
Figura 24 – William Higinbotham	3
Figura 25 – <i>Tennis for Two</i>	3
Figura 26 – <i>Spacewar</i>	3
Figura 27 – Ralph Baer	3
Figura 28 – <i>Magnovox Odyssey</i>	3
Figura 29 – <i>Pong</i>	3
Figura 30 – <i>Atari 2600</i>	3
Figura 31 – <i>Pac-Man</i>	3
Figura 32 – <i>Donkey Kong</i>	3
Figura 33 – <i>NES</i>	3
Figura 34 – <i>Mario</i>	3
Figura 35 – <i>Super Mario Brothers</i>	3
Figura 36 – <i>Sega Master System</i>	3
Figura 37 – <i>Alex Kid in the Miracle World</i>	3
Figura 38 – <i>Game Boy</i>	3
Figura 39 – <i>SNES</i>	3
Figura 40 – <i>Super Mario World</i>	3
Figura 41 – <i>Playstation</i>	3
Figura 42 – <i>Nintendo 64</i>	3
Figura 43 – <i>Dreamcast</i>	3
Figura 44 – <i>Playstation II</i>	3
Figura 45 – <i>Xbox</i>	3
Figura 46 – <i>Nintendo DS</i>	3
Figura 47 – <i>PSP</i>	3
Figura 48 – <i>Playstation III (Sony)</i>	3
Figura 49 – <i>Xbox 360 (Microsoft)</i>	3
Figura 50 – <i>Wii (Nintendo)</i>	3

Figura 51 – <i>Assassin's Creed (Playstation III)</i>	3
Figura 52 – <i>Test Drive Unlimited (Xbox 360)</i>	3
Figura 53 – <i>Battlefield 2 (PC)</i>	3
Figura 54 – Mercado Mundial de Jogos por Segmento.....	3
Figura 55 – Mercado Mundial de Jogos para Consoles por Grupo de Países.....	3
Figura 56 – Mercado Mundial de Jogos para PCs por Grupo de Países	3
Figura 57 – Atores e Etapas da Cadeia de Valor Clássica de Jogos para PC.....	3
Figura 58 – Cadeia de Valor de Jogos para PC	3
Figura 59 – Alternativa 1 para Cadeia de Valor Clássica de Jogos para PC.....	3
Figura 60 – Alternativa 2 para Cadeia de Valor Clássica de Jogos para PC.....	3
Figura 61 – Mark Cerny	3
Figura 62 – Etapas e Papéis do Ciclo de Desenvolvimento de Jogos.....	3
Figura 63 – Interface Gráfica do Jogo Insignia.....	3
Figura 64 – <i>Storyboard</i> do Jogo Insígnia.....	3
Figura 65 – Arte Conceitual do Jogo SWAT3	3
Figura 66 – Modelo 3D de Personagem Presente na <i>Unreal Engine 3</i>	3
Figura 67 – Modelo 3D de Personagem com Textura Presente na <i>Unreal Engine 3</i>	3
Figura 68 – Personagem Sendo Animado no <i>3D Studio Max</i>	3
Figura 69 – <i>Storyboard</i> do Jogo Oz – <i>The Magical Adventure</i>	3
Figura 70 – Arte Final de Cena do Jogo Oz – <i>The Magical Adventure</i>	3
Figura 71 – <i>Screenshot</i> da <i>Crytek Cryengine</i>	3
Figura 72 – Dimensões das Metodologias de Desenvolvimento de <i>Software</i>	3
Figura 73 – Práticas Primárias e Corolárias do XP	3
Figura 74 – Fases do <i>Game Waterfall Process</i>	3
Figura 75 – Arquitetura do <i>Rational Unified Process</i>	3
Figura 76 – Conceito de Disciplinas, Iterações, Fases e <i>Milestones</i>	3
Figura 77 – Estrutura do MDJ-PC	3
Figura 78 – Essência do MDJ-PC	3
Figura 79 – Fases e <i>Milestones</i> do MDJ-PC	3
Figura 80 – Visão Geral da Disciplina Modelagem de Mercado.....	3
Figura 81 – Atividades da Disciplina Modelagem de Mercado	3
Figura 82 – Visão Geral da Disciplina <i>Game Design</i>	3
Figura 83 – Atividades da Disciplina <i>Game Design</i>	3
Figura 84 – Visão Geral da Disciplina <i>Technical Design</i>	3
Figura 85 – Atividades da Disciplina <i>Technical Design</i>	3
Figura 86 – Visão Geral da Disciplina Implementação	3
Figura 87 – Atividades da Disciplina Implementação	3
Figura 88 – Visão Geral da Disciplina Criação de Arte 2D	3
Figura 89 – Atividades da Disciplina Criação de Arte 2D.....	3
Figura 90 – Visão Geral da Disciplina Criação de Arte 3D	3
Figura 91 – Atividades da Disciplina Criação de Arte 3D.....	3
Figura 92 – Visão Geral da Disciplina Criação de Áudio	3
Figura 93 – Atividades da Disciplina Criação de Áudio	3
Figura 94 – Visão Geral da Disciplina Teste	3
Figura 95 – Atividades da Disciplina Teste	3
Figura 96 – Visão Geral da Disciplina Distribuição	3
Figura 97 – Atividades da Disciplina Distribuição.....	3
Figura 98 – Visão Geral da Disciplina Gerência de Configuração e Mudanças	3
Figura 99 – Atividades da Disciplina Gerência de Configuração e Mudanças.....	3
Figura 100 – Visão Geral da Disciplina Gerência de Projeto	3
Figura 101 – Atividades da Disciplina Gerência de Projeto.....	3
Figura 102 – Visão Geral da Disciplina Ambiente	3
Figura 103 – Atividades da Disciplina Ambiente	3

LISTA DE TABELAS

Tabela 1 – Modelo Adaptativo x Modelo Preditivo	3
Tabela 2 – Gêneros de Jogos Digitais	3
Tabela 3 – Gêneros de Jogos Digitais	3
Tabela 4 – Gêneros de Jogos Digitais	3
Tabela 5 – Gêneros de Jogos Digitais	3
Tabela 6 – Gêneros de Jogos Digitais	3
Tabela 7 – Atores da Cadeia de Valor de Jogos para <i>PC</i>	3
Tabela 8 – <i>Conceptual Document</i>	3
Tabela 9 – <i>Game Design Document</i>	3
Tabela 10 – <i>Technical Design Document</i>	3
Tabela 11 – <i>Art List Document</i>	3
Tabela 12 – <i>Sound List Document</i>	3
Tabela 13 – <i>Test Plan</i>	3
Tabela 14 – <i>Post Mortem Document</i>	3
Tabela 15 – Papéis no Desenvolvimento de Jogos	3
Tabela 16 – Papéis no Desenvolvimento de Jogos	3
Tabela 17 – Principais Características do Ciclo de Desenvolvimento de Jogos	3
Tabela 18 – Principais Características do Ciclo de Desenvolvimento de Jogos	3
Tabela 19 – Principais Características do Ciclo de Desenvolvimento de Jogos	3
Tabela 20 – Valores do XP	3
Tabela 21 – Valores do XP	3
Tabela 22 – Práticas do XP	3
Tabela 23 – Fases do GWP	3
Tabela 24 – Melhores Práticas do Desenvolvimento de <i>Software</i>	3
Tabela 25 – Melhores Práticas do Desenvolvimento de <i>Software</i>	3
Tabela 26 – Características do RUP	3
Tabela 27 – Fases do RUP	3
Tabela 28 – Disciplinas do RUP	3
Tabela 29 – Disciplinas do RUP	3
Tabela 30 – Comparativo RUP vs. XP no Desenvolvimento de Jogos para <i>PC</i>	3
Tabela 31 – Comparativo RUP vs. XP no Desenvolvimento de Jogos para <i>PC</i>	3
Tabela 32 – Práticas do MDJ-PC	3
Tabela 33 – Características do MDJ-PC	3
Tabela 34 – Estereótipos do SPEM	3
Tabela 35 – Estereótipos do SPEM	3
Tabela 36 – Fase de Concepção	3
Tabela 37 – Fase de Pré-Produção	3
Tabela 38 – Fase de Produção	3
Tabela 39 – Fase de Pós-Produção	3

1. Introdução

Estudos recentes da *Price Waterhouse Coopers* (PWC) mostram que a indústria de entretenimento encontra-se em um impressionante estado de crescimento. As análises indicam que este mercado irá crescer a uma taxa bastante acelerada na segunda metade desta década, chegando a movimentar US\$ 1,8 trilhões em vendas anuais em 2010 [1].

Dentro desta indústria, merece destaque o mercado de jogos digitais por apresentar o maior crescimento dentre todos os mercados de entretenimento.

Em 2003, o mercado mundial de jogos digitais movimentou US\$ 22,3 bilhões [2]. Sem incluir os gastos com hardware e acessórios, o que elevaria este valor para US\$ 50 bilhões. É importante destacar que esta indústria já superou a cinematográfica que movimentou em 2003 US\$ 19 bilhões [3].

A indústria de jogos é uma das que mais se expande no mundo, tendo crescido 12,5% em 2002 e estima-se que chegue a 20,1% em 2008, ano que estaria movimentando cerca de US\$ 55,6 bilhões [2].

Prevê-se também uma aceleração do crescimento do mercado a partir do final de 2005 pela introdução de uma nova geração de consoles, o que costuma elevar as taxas de crescimento da indústria.

O crescimento desta indústria acompanha o aumento de adeptos aos jogos digitais. Este crescimento pode ser constatado pelo dado de que aproximadamente 43% da população dos EUA são consumidores de jogos de computadores [4, 5].

A evidente expansão deste mercado traz repercussões em todos os níveis da cadeia de valor do setor. É preciso inovar, é preciso produzir conceitos e jogos criativos capazes de entreter o usuário através de novas experiências e mecanismos de interatividade. Portanto, a evolução do mercado exige mudanças desde o processo de vendas até os processos de produção e desenvolvimento de jogos digitais.

Atualmente, a criação de alguns jogos é comparável às grandes produções cinematográficas. Em 2004, a Microsoft lançou o jogo *Halo 2* (Figura 1), fruto de um projeto que envolveu mais de 190 pessoas, demorou 3 anos para ser concluído e teve um custo de desenvolvimento total de US\$ 40 milhões.

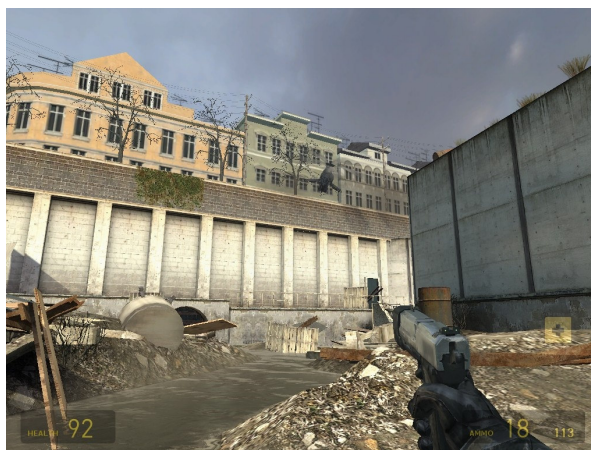


Figura 1 – Halo 2

1.1 Contexto Atual do Desenvolvimento de Jogos

O segmento de jogos eletrônicos mudou bastante desde suas origens até os dias atuais. O processo de desenvolvimento acompanha estas mudanças. No passado, os jogos eram desenvolvidos por pequenos times, constituídos essencialmente de programadores e em pouco tempo eram concluídos. No entanto, esta realidade contrasta com a dos dias atuais.

O desenvolvimento de jogos atualmente segue padrões bem distintos. É preciso lidar com times heterogêneos e, em algumas circunstâncias, com projetos de duração superior a alguns anos. Esta realidade é exigida principalmente pela maior complexidade, sob vários aspectos, dos jogos eletrônicos atuais.

Por exemplo, o jogo *Indigo Prophecy* (Figura 2), lançado em setembro de 2005, demorou 2 anos para ser concluído e a equipe de desenvolvimento era composta de aproximadamente 80 pessoas trabalhando em regime de tempo integral [7].



Figura 2 – *Indigo Prophecy*

A maior complexidade dos projetos de desenvolvimento de jogos faz necessário o uso de técnicas e modelos de desenvolvimento para tentar minimizar os problemas naturais que podem surgir ao longo do projeto.

Esta necessidade tende a aumentar com a nova geração de *consoles*. Daglow [8] aponta que o desenvolvimento de jogos para a nova geração trará maior complexidade, uma vez que haverá mudanças nas ferramentas de criação. Desenvolver protótipos será ainda mais crítico e será preciso misturar elementos da indústria cinematográfica com a de jogos. Estas mudanças contribuirão para o aumento da necessidade de modelos de desenvolvimento mais elaborados.

1.2 Desafios do Cenário Atual

Vários modelos de desenvolvimento de software foram concebidos nos últimos anos [9]; contudo, eles não são facilmente aplicáveis ao desenvolvimento de jogos. Estes modelos consideram cenários que nem sempre são aplicáveis à indústria de jogos. Por exemplo: em mercados de jogos consolidados, como o da Inglaterra e o dos Estados Unidos, normalmente, existem dois artistas para cada programador [10]; realidade que contrasta com a do desenvolvimento de software em geral.

Atualmente, há algumas iniciativas em criar modelos específicos para o desenvolvimento de jogos. No entanto, estas geralmente são superficiais e nem sempre correspondem à realidade do mercado e das empresas. Algumas destas iniciativas são: o *Extreme Game Development* (XGD) [11] e o *Game Unified Process* (GUP) [12].

1.3 Objetivos do Trabalho de Graduação

O objetivo do Trabalho de Graduação em questão é propor um modelo preditivo de desenvolvimento de jogos para *PC* que envolvam um longo período de desenvolvimento e grandes equipes heterogêneas. Fazem parte do objetivo deste estudo os seguintes aspectos:

1. Analisar o mercado de jogos eletrônicos, com maior ênfase no segmento *PC*;
2. Identificar as características do desenvolvimento de jogos para *PC*;
3. Reconhecer as necessidades emergentes do desenvolvimento de jogos para *PC*;
4. Propor um modelo preditivo para o desenvolvimento de jogos para *PC*.

Os argumentos que nortearam a definição dos objetivos e do escopo deste trabalho são apresentados a seguir.

1.3.1 Segmento *PC*

Difícilmente um único modelo poderia ser adotado nos vários segmentos do mercado de jogos em função de diferenças existentes. Por exemplo: atualmente os segmentos *wireless* e *on-line* possuem projetos com ciclo de desenvolvimento e equipes consideravelmente menores do que os projetos dos segmentos *PC* e *console*.

Logo, o escopo do modelo a ser desenvolvido neste trabalho será reduzido e focado ao segmento de jogos para *PCs*. O principal motivo por optar pelo segmento *PC* é que os processos de desenvolvimento nesta área costumam ser mais complexos do que os dos segmentos *on-line* e *wireless*, o que exige um modelo de desenvolvimento melhor concebido e estruturado; no entanto, nem tão complexos quanto os do segmento *console*.

1.3.2 Adaptativo vs. Preditivo

Outra decisão importante é a de basear este modelo de desenvolvimento de jogos para *PC* nos princípios, que forem aplicáveis, dos modelos de desenvolvimento de software mais preditivos.

Barry Boehm e Richar Turner sugerem usar uma análise de riscos para escolher entre usar um modelo de desenvolvimento adaptativo (ágil) ou preditivo [13]. Veja na Tabela 1 alguns pontos que eles sugerem analisar ao escolher que tipo modelo utilizar.

Tabela 1 – Modelo Adaptativo x Modelo Preditivo

Modelo Adaptativo	Modelo Preditivo
Desenvolvedores muito experientes	Desenvolvedores pouco experientes
Pequeno número de desenvolvedores	Grande número de desenvolvedores
Cultura que demanda menor ordem	Cultura que demanda maior ordem

O processo de desenvolvimento de jogos para *PC* possui mais semelhanças com as características do modelo preditivo: é comum um grande número de desenvolvedores e acontecem em uma cultura que demanda muita ordem. Por conseguinte, o modelo de desenvolvimento a ser proposto fará uso de princípios de modelos mais preditivos.

1.4 Relevância do Trabalho

É crescente o desejo de modelos de desenvolvimento melhor definidos que atendam as necessidades atuais do desenvolvimento de jogos digitais e, simultaneamente, sejam flexíveis à realidade de cada empresa.

Há vários relatos na indústria de jogos digitais de insatisfação com a manutenção dos custos estimados de um projeto, com o esforço necessário para atingir os níveis de qualidade almejados e com a incapacidade de construir cronogramas reais de desenvolvimento [14]. Ou seja, é um desejo comum, de várias empresas do setor, ter modelos de desenvolvimento que permitam às empresas uma melhor habilidade em planejar, conduzir e controlar o desenvolvimento de um jogo.

Portanto, os objetivos deste trabalho de graduação estão alinhados com um claro desejo da área de produção de jogos digitais. Pode-se melhor evidenciar este fato observando que pelo menos quatro apresentações foram realizadas na *Game Developers Conference* 2006 sobre modelos de desenvolvimento de jogos [15].

1.5 Metodologia Aplicada ao Trabalho de Graduação

A metodologia de desenvolvimento deste trabalho de graduação é composta pelas seguintes fases:

1. Estudo do mercado de entretenimento digital.

O objetivo da primeira fase é entender o cenário atual, assim como as perspectivas futuras, do promissor mercado de entretenimento digital. É preciso saber quais são as principais necessidades e variáveis deste setor.

2. Estudo do processo de desenvolvimento de jogos para plataforma PC.

O objetivo da segunda fase é identificar e compreender as principais características presentes no desenvolvimento de jogos digitais para plataforma PC. É preciso saber quais são as principais deficiências existentes no cenário atual.

3. Estudo dos modelos de desenvolvimento de software.

O objetivo da terceira fase é compreender as principais características presentes nos modelos de desenvolvimento de *software*. É preciso entender que aspectos do desenvolvimento de jogos para PC estes modelos não conseguem atender de uma forma satisfatória.

4. Estudo detalhado do *Rational Unified Process*.

O objetivo da quarta fase é estudar de forma detalhada o RUP. É preciso entender como este modelo de desenvolvimento de software é estruturado e definido.

5. Proposta de um modelo de desenvolvimento de jogos de computador.

O objetivo da quinta e última fase é propor um modelo preditivo de desenvolvimento de jogos inspirado nos moldes do RUP; contudo, adequado às necessidades atuais do desenvolvimento de jogos digitais de computador.

1.6 Organização do Trabalho

Este trabalho de graduação encontra-se estruturado em 6 capítulos. Segue abaixo uma breve descrição do conteúdo presente nos próximos capítulos.

Capítulo 2: Jogos Eletrônicos

O Capítulo 2 apresenta um panorama geral do mercado de jogos eletrônicos. Fala-se dos tipos de mídia e da evolução deste mercado. Também são apresentadas neste capítulo as principais mudanças ocorridas no desenvolvimento de jogos desde o início do setor até os tempos atuais.

Capítulo 3: Ciclo de Produção de Jogos para PC

O Capítulo 3 apresenta com mais detalhes as principais características, papéis, atividades e artefatos presentes no processo de produção de jogos para plataforma *PC*. Por fim, enumeram-se os principais aspectos pertinentes a este processo de produção.

Capítulo 4: Metodologias de Desenvolvimento de Software

O Capítulo 4 apresenta um levantamento das principais correntes de modelos de desenvolvimento de software. Dentro de cada corrente, descrevem-se as principais metodologias adotadas pelo mercado. Por fim, faz-se uma análise comparativa para identificar que metodologia melhor atende às características presentes no desenvolvimento de jogos para *PC*.

Capítulo 5: Modelo Preditivo de Desenvolvimento de Jogos PC

O Capítulo 5 apresenta os detalhes do modelo preditivo proposto para o desenvolvimento de jogos de computador.

Capítulo 6: Conclusões

O Capítulo 6 apresenta que conclusões podem ser obtidas a partir do trabalho desenvolvido, quais contribuições este trabalho deixa e, por fim, sugestões de atividades futuras relacionadas com o tema aqui abordado.

2. Jogos Eletrônicos

Este capítulo tem como objetivo apresentar um panorama geral do mercado de jogos digitais. As cinco seções contidas neste capítulo abordarão os seguintes temas:

Seção 1: Introdução

Explica-se o fenômeno crescente de consumo de entretenimento digital e quais são os principais gêneros de jogos digitais existentes atualmente.

Seção 2: Mídias de Jogos Digitais

Enumeram-se as principais mídias utilizadas por jogos eletrônicos.

Seção 3: Evolução dos Jogos Digitais

Apresenta-se a evolução dos jogos digitais desde a sua origem até os tempos modernos.

Seção 4: O Mercado de Jogos Digitais

Desenha-se a evolução do cenário mercadológico no qual está inserido o entretenimento digital.

Seção 5: Mudanças no Desenvolvimento de Jogos

Descrevem-se as principais mudanças no desenvolvimento de jogos desde o início do setor até os dias atuais.

2.1 Introdução

Por que o consumo de jogos digitais é cada vez maior? O que justifica a crescente demanda deste setor? Por que a indústria de entretenimento digital é uma das que mais cresce?

Uma das muitas respostas para estas perguntas reside na definição básica do que vem a ser entretenimento. De acordo com o *Houaiss*, entretenimento é ocupar-se de maneira prazerosa, é distrair-se, é recrear-se. [16]. Recreação esta que se pode dar de forma passiva ou ativa.

O entretenimento é, portanto, uma realidade presente na vida cotidiana. Como já dizia Minsky, um especialista em inteligência artificial: “*We must do a few things to survive. Everything else is entertainment.*” [17]. Em português: devemos fazer algumas coisas para sobreviver, tudo mais é entretenimento.

Continuando a linha de raciocínio aqui delineada, o *Wikipedia* define jogo como uma atividade, que envolve uma ou mais pessoas, com um objetivo a ser perseguido. Associado a este objetivo, tem-se um conjunto de regras que determinam o que pode ou não ser feito no jogo. A definição de jogo pelo *Wikipedia* ainda afirma que jogar é uma maneira de entretenimento [18].

Logo, se as pessoas buscam entretenimento constantemente e jogar é uma maneira de conseguir este divertimento almejado; então, pode-se concluir que muitas pessoas jogam com o objetivo de se entreter.

Portanto, a busca pelo entretenimento é natural do ser humano. Os avanços tecnológicos trouxeram novas oportunidades de entretenimento; criando, assim, a indústria de entretenimento digital.

De acordo com Malone [19], as pessoas jogam por três principais razões: fantasia, desafio e curiosidade. Ou seja, quer seja para se transportar mentalmente para um universo fantástico, quer seja para solucionar um complexo enigma ou descobrir o que acontece com o personagem do jogo ao entrar em certo cenário.

A teoria *Natural Funativity* desenvolvida por Falstein detalha ainda mais a influência dos jogos digitais sobre as pessoas [20]. De acordo com ele, jogos proporcionam quatro formas principais de divertimento: físico, social, mental e misto. O primeiro pode ser alcançado em jogos que estimulam, por exemplo, instintos de sobrevivência. O segundo é alvo de jogos que permitem a formação de grupos virtuais. O divertimento mental acontece geralmente porque é preciso manipular padrões. O quarto e último tipo é nada mais do que a sinergia dos outros três tipos de diversão.




2.1.1 Gêneros de Jogos Digitais

Há várias classificações de gêneros possíveis para jogos digitais; contudo, uma das mais aceitas é a classificação de Bakie [21] pela sua completude.

De acordo com Bakie, os jogos podem ser classificados nos seguintes gêneros: aventura, ação, plataforma, luta, tiro em primeira pessoa (FPS), estratégia em tempo real (RTS), estratégia por turnos, *role playing game* (RPG), *massive multiplayer on-line role playing game* (MMORPG), *stealth*, horror, simulação, corrida, esporte, rítmico, *puzzle*, *mini-game*, tradicional, educacional, sério e *advergames*¹.

A Tabela 2-6 descreve detalhadamente cada um dos gêneros, em conjunto com imagens de jogos representativos de cada um. É importante mencionar que normalmente um jogo não pertence a um único gênero; ou seja, é comum um jogo combinar aspectos de vários gêneros.

Tabela 2 – Gêneros de Jogos Digitais

Gênero	Descrição	Imagem
Aventura	É caracterizado pela exploração de ambientes e interação com demais personagens. Foca na narrativa e não em desafios baseados em reflexos [22].	 <p>Figura 3 – Rayman 3</p>
Ação	É um gênero bastante combinado com outros. Possui como característica principal elementos rápidos de combate e movimentação [21].	 <p>Figura 4 – Starcraft Ghost</p>
Plataforma	Os jogos originais de plataforma envolviam o personagem correndo e pulando em um cenário bidimensional composto por plataformas [21].	 <p>Figura 5 – Super Mario 3</p>

¹ Algumas classificações foram mantidas em inglês por serem termos globalmente utilizados.

Tabela 3 – Gêneros de Jogos Digitais


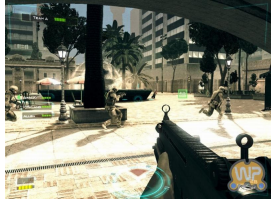



Gênero	Descrição	Imagem
Luta	Em jogos deste gênero, o jogador luta contra outros jogadores ou contra o computador utilizando, normalmente, artes marciais e espadas [21].	 <p>Figura 6 – Tekken 5</p>
FPS	É um tipo de jogo de ação que coloca o jogador em uma perspectiva de visão de primeira pessoa [21].	 <p>Figura 7 – Ghost Recon 3</p>
RTS	Em RTS típicos, o objetivo do jogador é coletar recursos, construir um exército e controlar suas unidades para atacar os inimigos [21].	 <p>Figura 8 – Warcraft 3</p>
Estratégia por Turnos	É um gênero similar ao RTS; contudo, as ações dos jogadores são realizadas em turnos alternados [21].	 <p>Figura 9 – Civilization IV</p>
RPG	É a versão digital do estilo presente em RPG clássicos como: <i>Dungeons & Dragons</i> [21].	 <p>Figura 10 – Final Fantasy XIII</p>

Tabela 4 – Gêneros de Jogos Digitais


Gênero	Descrição	Imagem
MMORPG	É um RPG digital presente em um mundo virtual on-line persistente e povoado simultaneamente por centenas de jogadores [21].	 <p>Figura 11 – World of Warcraft</p>
<i>Stealth</i>	Costumam ser parecidos com jogos de tiro em primeira ou terceira pessoa; contudo, possuem um conceito de jogo não tão apegado à ação. Tendem a ser mais metódicos [21].	 <p>Figura 12 – Splinter Cell Double Agent</p>
Horror	Tipicamente envolvem exploração de ambientes povoados por monstros e criaturas imaginárias [21].	 <p>Figura 13 – Silent Hill 3</p>
Simulação	Buscam simular com alto grau de detalhes ambientes próximos do real [21].	 <p>Figura 14 – Flight Simulator 2004</p>
Corrida	O principal objetivo é disputar corrida em veículos que vão desde carros de corrida até bicicletas. Tenta recriar com fidelidade aspectos do mundo real pertinentes ao contexto do jogo [21].	 <p>Figura 15 – Project Gotham Race 3</p>

Tabela 5 – Gêneros de Jogos Digitais




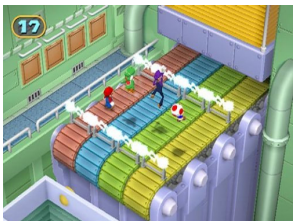
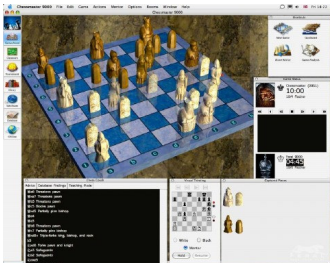



Gênero	Descrição	Imagem
Esporte	Cobre a gama de jogos que simulam a experiência esportiva. Tenta também recriar com fidelidade aspectos do mundo real pertinentes ao contexto do jogo [21].	 <p>Figura 16 – FIFA 2006</p>
Rítmico	É um gênero que explora a habilidade do jogador em sincronizar comandos com o tempo rítmico de uma música [21].	 <p>Figura 17 – DDR</p>
Puzzle	Combina elementos de identificação de padrões, lógica, estratégia e sorte. Geralmente em conjunto com o elemento tempo [21].	 <p>Figura 18 – Bejeweled</p>
Mini-game	É um gênero que envolve jogos geralmente curtos e simples. Às vezes são utilizados como um desafio curto em jogos maiores [21].	 <p>Figura 19 – Mario Party 7</p>
Tradicional	Engloba versões digitais de jogos tradicionais de carta e tabuleiro [21].	 <p>Figura 20 – Chessmaster 10000</p>

Tabela 6 – Gêneros de Jogos Digitais

Gênero	Descrição	Imagem
Educacional	É um gênero comprometido com o ensino de conceitos educacionais para um público-alvo específico [21].	 <p>Figura 21 – Carmen Sandiego</p>
Sério	É um gênero relativamente novo que nasceu como um mecanismo mais barato e divertido de ensinar aspectos do mundo real para adultos [21].	 <p>Figura 22 – Worky's Challenge</p>
Advergames	Busca usar jogos digitais para promover e divulgar um produto, organização ou ponto de vista [23].	 <p>Figura 23 – Clio Xtrem Racer</p>

2.2 Mídias de Jogos Digitais

Atualmente, há praticamente sete tipos principais de mídias em jogos digitais: *consoles*, *PCs*, *celulares*, *handhelds*, *consoles* portáteis, *TV* digital e *multiplataforma* [2].

Consoles podem ser vistos como um equipamento digital dotado de um conector para o aparelho de TV e um encaixe para algum tipo de unidade de armazenamento contendo programas e/ou dados de um jogo em particular. Os consoles com maior penetração no mercado atualmente são: o *Playstation 2* (Sony), o *Xbox* e o *Xbox 360* (Microsoft) e o *Game Cube* (Nintendo).

PCs englobam os jogos que geralmente são vendidos fisicamente em grandes redes varejistas, obtidos on-line via assinatura ou jogados pela internet utilizando uma conexão de banda larga.

Celulares constituem uma nova oportunidade visivelmente crescente para jogos digitais [2].

Handhelds e *consoles* portáteis são plataformas que possuem telas de maior tamanho e melhor qualidade que celulares, possibilitando o acesso a jogos de melhor resolução gráfica. *Consoles* portáteis que têm atingido bons índices de venda são o *Playstation Portable* (Sony) e o *Nintendo DS* (Nintendo).

TV digital é uma nova possibilidade, ainda incipiente, de jogos que podem ser rodados em aparelhos de TV digital como forma de conteúdo interativo.

Multipataforma engloba jogos criados para atingir diferentes nichos tecnológicos.

Além das sete mídias principais relacionadas acima, tem-se ainda o *Arcade*. São dispositivos de entretenimento geralmente operados por moedas ou cartões magnéticos. Costumam estar localizados em locais públicos como restaurantes, *shoppings*, etc.

2.3 Evolução dos Jogos Digitais

Pode-se dizer que a história dos jogos digitais começou com William Higinbotham [21]. Na época, 1958, era interesse do governo americano que a população dos EUA tivesse acesso a unidades do *United States Department of Energy* com o objetivo de criar uma imagem amigável para o poder atômico em desenvolvimento.

Contudo, Higinbotham (Figura 24) notou que o interesse dos visitantes era baixo e que eles sentiam falta de alguma atividade interativa. Foi então que Higinbotham teve a idéia de criar um jogo (*Tennis for Two*, Figura 25) para entreter o público visitante. O jogo era executado através de um computador analógico em conjunto com um osciloscópio.

Entretanto, a idéia não teve uma grande repercussão. O jogo foi desativado e substituído por outra atração no ano seguinte.



Figura 24 – William Higinbotham

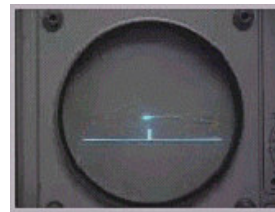


Figura 25 – Tennis for Two

Nos anos seguintes, em 1961, um aluno do MIT, Steve Russel [21], desenvolveu um jogo como projeto de uma disciplina da graduação: *Spacewar* (Figura 26). Projeto no qual dedicou 6 meses, totalizando um esforço de aproximadamente 200 horas. Este jogo era executado em um DEC PDP-1. Apesar da iniciativa, o projeto acadêmico não tomou projeções comerciais.

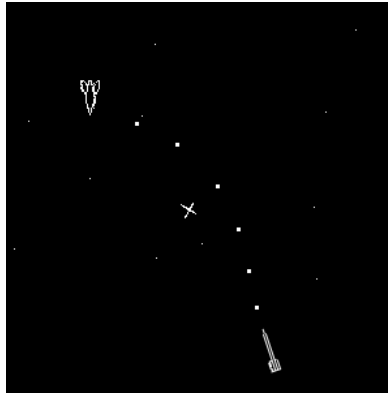


Figura 26 – *Spacewar*

Em 1972, os jogos atingiram as massas. Ralph Baer [21] (Figura 27) desenvolveu um *console* que podia ser acoplado a um televisor: *Magnavox Odyssey* (Figura 28). Cerca de 100 mil unidades foram vendidas.



Figura 27 – Ralph Baer



Figura 28 – *Magnovox Odyssey*

No entanto, o primeiro grande visionário foi Nolan Bushnell [21] que com a ajuda de vários estudantes criou 7 jogos para computadores e recriou uma versão de *Spacewar* para *Arcade*. Após alguns outros projetos, teve a idéia de criar uma empresa para produzir jogos: *Atari*, fundada em 1972. O primeiro grande sucesso da empresa foi o jogo *Pong* (Figura 29).

Em 1977, a *Atari* lançou o *Atari Video Computer System* (VCS) que depois foi nomeado *Atari 2600*: o primeiro *console* com a noção de cartucho e *joystick* (Figura 30).

No mesmo ano, a *Nintendo*, uma empresa fundada em 1889 que produzia jogos japoneses de cartas, lançou o seu primeiro *console*: *TV Game 6*.



Figura 29 – *Pong*



Figura 30 – *Atari 2600*

Em 1980, a *Namco* lança *Pac-Man* (Figura 31): o primeiro jogo cujo personagem controlado pelo jogador era animado. No ano seguinte, a *Nintendo* lançou *Donkey Kong* (Figura 32).



Figura 31 – Pac-Man



Figura 32 – Donkey Kong

Em seguida, entre 1982-1983, a indústria de *consoles* teve um grande declínio. Grandes empresas de *hardware* e de entretenimento faliram: *Apollo*, *US Games*, *Telesys*, *Data Age*, *Spectravision* entre outras.

Alguns especialistas apontam como principais razões da crise uma série de jogos para o *Atari 2600* que não agradou os jogadores e também a difusão de computadores pessoais a custos mais acessíveis, como o *Commodore 64*. Com a evolução dos computadores pessoais, cresceu a cultura de jogos para esta mídia.

Em 1985, a *Nintendo* lançou o *console Nintendo Entertainment System* (NES, Figura 33). A indústria de *consoles* voltou a se erguer nos anos seguintes com Shigeru Miyamoto [9], da *Nintendo*, que criou um dos personagens mais inesquecíveis da história dos jogos: *Mario* (Figura 34 e 35).



Figura 33 – NES



Figura 34 – Mario



Figura 35 – Super Mario Brothers

Em 1986, a *Sega* entra no mercado de *consoles* ao lançar o *Sega Master System* (Figura 36). O jogo *Alex Kid in the Miracle World* (Figura 37) foi um dos primeiros títulos para o *Master*.



Figura 36 – Sega Master System



Figura 37 – Alex Kid in the Miracle World

Em 1989, a *Nintendo* lançou o *console portátil Game Boy* (Figura 38). Em 1991, a *Nintendo* lançou o *Super Nintendo Entertainment System* (SNES, Figura 39).



Figura 38 – Game Boy



Figura 39 – SNES

A série *Mario* continuou no *SNES* com títulos que atraíram cada vez mais usuários para o mercado de *consoles* (Figura 40).



Figura 40 – Super Mario World

Em 1994 e 1995, a *Sony* entrou no mercado de *consoles*, até então dominado pela *Nintendo* e *Sega*, ao lançar o *Playstation* (Figura 41). Nos anos seguintes, impulsionados pela alta rentabilidade do setor, outros consoles foram desenvolvidos e lançados no mercado, como o *Nintendo 64* (Figura 42) e o *Dreamcast* (Figura 43).



Figura 41 – Playstation



Figura 42 – Nintendo 64



Figura 43 – Dreamcast

Em 2000, a *Sony* lança a nova versão do *Playstation*: o *Playstation II* (Figura 44), um dos consoles mais vendidos ainda nos dias atuais.



Figura 44 – Playstation II

2001 marcou a entrada da *Microsoft* no segmento de entretenimento digital com o lançamento do *Xbox* (Figura 45).



Figura 45 – Xbox

Os anos seguintes marcaram o lançamento de *consoles* portáteis com poder de processamento bastante superior aos dispositivos portáteis anteriores: o *Nintendo DS* (Figura 46) e o *Playstation Portable* (PSP, Figura 47).



Figura 46 – Nintendo DS



Figura 47 – PSP

A indústria de *consoles* voltou a se movimentar no final de 2005 e início de 2006 com o anúncio dos *consoles* de uma nova geração. Estes são assim chamados por representarem uma quebra no paradigma tecnológico de *consoles*. Veja fotos destes na Figura 48, 49 e 50.



Figura 48 – Playstation III (Sony)



Figura 49 – Xbox 360 (Microsoft)



Figura 50 – Wii (Nintendo)

Em menos de 50 anos, a sociedade presenciou uma evolução impressionante do poder tecnológico dos *consoles*. Evolução esta que impulsionou mudanças, de mesma grandeza, nos jogos desenvolvidos. O poder de processamento dos *consoles* é cada vez maior e tem impulsionado novos paradigmas gráficos, físicos e de programação [24].

O aumento do poder tecnológico dos *consoles* também foi acompanhado por uma evolução mais discreta da capacidade de processamento dos computadores pessoais. O que também provocou mudanças significativas nos jogos desenvolvidos para este

tipo de mídia. Veja na Figura 51, 52 e 53 imagens de jogos desenvolvidos para algumas das mídias atuais.



Figura 51 – Assassin's Creed (Playstation III)



Figura 52 – Test Drive Unlimited (Xbox 360)



Figura 53 – Battlefield 2 (PC)

2.4 O Mercado de Jogos Digitais

O mercado de jogos digitais encontra-se segmentado basicamente entre: jogos para *consoles* (73% do gasto mundial), *PCs* (17%), dispositivos *wireless* (3%) e jogos *on-line* (7%) [2]. Merece um destaque estes dois últimos segmentos pelas suas impressionantes taxas de crescimento (106,7% e 69,5% respectivamente). Apesar deste crescimento, o mercado de *consoles* ainda é o mais lucrativo em função dos custos envolvidos neste segmento. O mercado de *PCs* é o único para o qual não se prevê um crescimento nos próximos anos. Veja na Figura 54 a divisão do mercado de jogos em função dos seus segmentos.

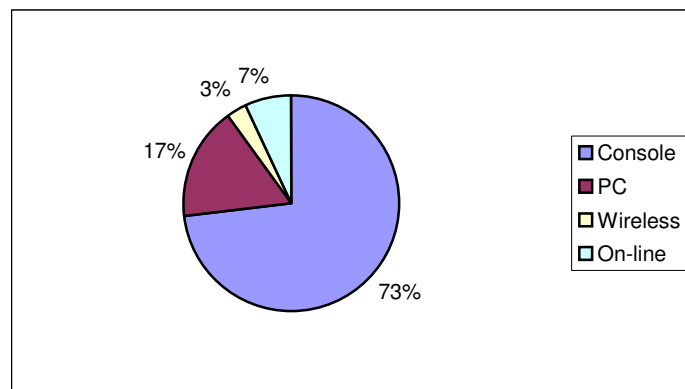


Figura 54 – Mercado Mundial de Jogos por Segmento

2.4.1 Internacional

Quanto à distribuição geográfica, o mercado encontra-se concentrado em 5 pólos: EUA, EMEA (Europa e África), Ásia/Pacífico, América Latina (AL) e Canadá que manipulam 34%, 24%, 38%, 2% e 3% dos gastos com jogos, respectivamente [2] (Figura 55).

No campo de *consoles*, os EUA encontram-se praticamente empatados com a Ásia/Pacífico: US\$ 5,8 bilhões e US\$ 5,9 bilhões do mercado mundial. Nos EUA a utilização de *consoles* é tão grande que cerca de metade das pessoas com mais de 6 anos jogam [25].

Um dado interessante é que nos EUA só 37% dos títulos vendidos são desenvolvidos no país. Já na Ásia/Pacífico, esta taxa é de 95% para títulos locais. A EMEA ocupa a terceira posição neste segmento com US\$ 3,6 bilhões do mercado mundial. A AL possui uma participação ínfima (US\$ 431 milhões) em função de altos índices de pirataria (cerca de 70% na região).

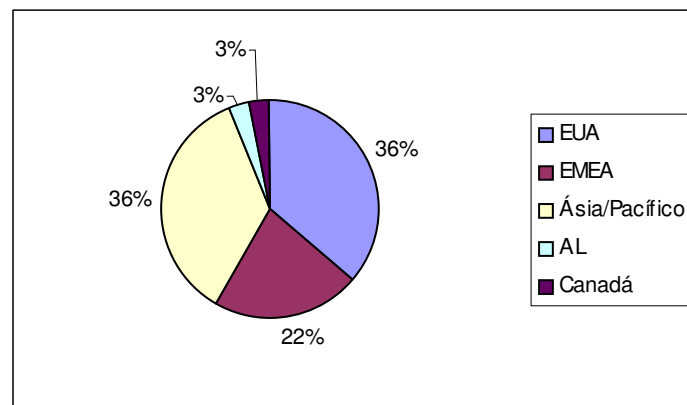


Figura 55 – Mercado Mundial de Jogos para Consoles por Grupo de Países

O mercado de jogos *on-line* é um dos que mais cresce. Os jogadores possuem em média 18 a 35 anos e jogam aproximadamente 13 horas por semana. Este segmento é dominado pela Ásia/Pacífico (42% do segmento), uma vez que no país as taxas mensais de assinatura de jogos *on-line* são mais baratas (US\$ 8,75 contra a média de US\$ 12,50) e banda larga é uma tecnologia difundida [2]. Os EUA ocupam a segunda posição neste segmento. O caráter promissor deste segmento pode ser constatado pelo recente feito do jogo *World of Warcraft* da Blizzard que atingiu a impressionante marca de 5,5 milhões de unidades vendidas em todo o mundo [26].

O mercado de *wireless* aposta nos estudos que mostram 4 de 5 usuários de dispositivos móveis em 2005 jogando uma média de 4,4 horas por semana [27]. A Ásia/Pacífico também dominam este segmento em função da grande difusão de telefones celulares na região. A EMEA ocupa o segundo lugar.

O mercado de *PCs* é o único para o qual não se prevê um grande crescimento (Figura 56). Contudo, estudos mostram que jogos de computadores são mais usados que processadores de textos. A Ásia/Pacífico é o principal ator neste segmento, atuando em um mercado extremamente fechado (consome basicamente títulos locais). Os EUA ocupam a segunda posição, acompanhados pela EMEA. Na EMEA se destaca a Alemanha que, contra a média mundial, investe cerca de 66% dos seus gastos na área com jogos para *PC*.

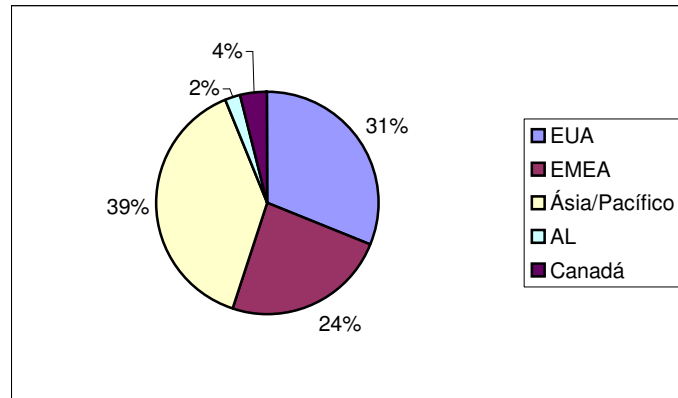


Figura 56 – Mercado Mundial de Jogos para PCs por Grupo de Países

2.4.2 Nacional

Desde os anos 80 existem no Brasil estudos para promover o desenvolvimento de jogos. Contudo, somente em 1995 foi fundada a primeira empresa exclusiva para o mercado de jogos: Perceptum. Nos anos seguintes, outras empresas foram fundadas como a SouthLogic (1996) e a Continuum (1998).

Em 2001, no Paraná, foi criada a Gamenet: uma rede para fomentar empresas desenvolvedoras de jogos. Esta iniciativa teve conseqüências nacionais: em 2002, a quantidade de empresas de jogos fundadas em todo o país foi igual a 7.

Em 2004, foi criada a Associação Brasileira de Desenvolvedoras de Jogos (ABRAGAMES) que passou a atuar junto ao governo e outras empresas para promover o mercado de jogos no país. De acordo com levantamento desta, em 2005 o Brasil contava com um quadro de 55 empresas atuando no ramo. A maior parte (77%) das empresas brasileiras é pequena e contam com não mais de 19 funcionários [2].

O faturamento em 2003 foi de R\$ 2,39 milhões. Em 2004, este valor subiu para R\$ 4,2 milhões; ou seja, apresentando um crescimento de 79% em relação ano anterior, mostrando a potencialidade do mercado nacional.

A maioria destas empresas tem como cliente publicadoras que distribuem os seus jogos (nacional ou internacionalmente). Em segundo lugar, têm-se como clientes empresas de telefonia; seguidas, em terceiro lugar, por empresas de publicidade.

A maior parte das empresas desenvolvedoras de jogos nacionais encontra-se no Paraná (33%). Em segundo lugar, tem-se São Paulo, com 30%. Pernambuco ocupa o quarto lugar com 9% [10].

Contudo, Pernambuco representa 16% do faturamento do país (a terceira maior parcela, ficando atrás do Rio de Janeiro e São Paulo), equivalente ao faturamento do Paraná. Outro dado importante é que 66% das empresas trabalham em cima do segmento PC e 23% com celulares.

Um dos grandes entraves para um maior crescimento do mercado nacional é o alto índice de pirataria. Dificilmente uma empresa nacional consegue vender jogos exclusivamente para o mercado brasileiro e conseguir uma rentabilidade satisfatória. A conseqüência é o investimento no mercado internacional como alternativa.

Entretanto, a situação no exterior também não é muito propícia para os produtos brasileiros; uma vez que, há dificuldades em ter acesso a boas publicadoras, os *royalties* dos produtos ficam com os agentes e publicadoras internacionais e o desenvolvedor é remunerado só pelo desenvolvimento. Os principais destinos no exterior dos jogos brasileiros são: EUA e Europa (Alemanha).

Atualmente, as principais empresas brasileiras atuantes no ramo são: *Continuum*, *Espaço Informática*, *GreenLand Studios*, *Ignis*, *Oniria Entertainment*, *Southlogic Studios*, *Jynx Playware*, *Meantime* e *Preloud Electronic Entertainment Brazil*.

2.5 Mudanças no Desenvolvimento de Jogos

Como mencionado acima, o mercado de entretenimento digital cresceu e sofreu grandes mudanças ao longo dos seus aproximados 50 anos de existência. Atualmente, este é um mercado com demanda crescente e com um número de adeptos cada vez maior.

No início, as pessoas se entretinham com um quadrado branco e duas barras retangulares brancas em cada lado da tela (*Pong*). Atualmente, os jogadores querem mundos tridimensionais realísticos em alta resolução. Ou seja, o sentimento desejado pelo jogador mudou drasticamente [6].

No início dos anos 80, o desenvolvimento de jogos era realizado por poucos programadores a trabalhar em um título por algumas poucas semanas. O trabalho de codificar, criar a arte e o áudio do jogo era desempenhado pela mesma pessoa. Esta realidade contrasta com a atual. Os jogos desenvolvidos para as mídias com maior poder tecnológico, *consoles* e *PCs*, chegam a contar com times de centenas de pessoas envolvendo programadores, artistas, *designers* e engenheiros de áudio.

Além dos papéis mais clássicos, há uma preocupação crescente com a garantia da qualidade, gerência, suporte e marketing. De acordo com as perspectivas, este quadro tende a crescer; ou seja, serão desenvolvidos jogos mais complexos para mídias mais poderosas utilizando equipes de desenvolvimento ainda maiores.

A viabilidade das tendências de crescimento foge ao escopo deste trabalho. Há questões ainda não claras em relação a este tema. Por exemplo: os custos de desenvolvimento são cada vez maiores, tornando os jogos mais caros. Até onde o usuário está disposto a pagar por um dado nível de qualidade é um dos muitos questionamentos em aberto.

Apesar das incertezas, tem-se observado uma preocupação crescente do mercado de entretenimento digital em formalizar e automatizar processos realizados no passado de forma quase que artesanal. Procuram-se, portanto, modelos e mecanismos com o objetivo de aperfeiçoar o processo de produção de jogos.

Contudo, dificilmente um único modelo poderia ser adotado nos vários segmentos do mercado de jogos em função de diferenças existentes. Por exemplo: atualmente os segmentos *wireless* e *on-line* possuem projetos com ciclo de desenvolvimento e equipes consideravelmente menores do que os projetos dos segmentos *PC* e *console*.

Este trabalho irá focar no desenvolvimento de jogos para plataforma *PC*. O principal motivo por optar por este segmento é que os processos de desenvolvimento nesta área costumam ser mais complexos do que os dos segmentos *on-line* e *wireless*, o que exige um modelo de desenvolvimento melhor concebido e estruturado; no entanto, nem tão complexos quanto os do segmento *console*.

3. Ciclo de Produção de Jogos para PC

Este capítulo tem como objetivo detalhar os principais aspectos relevantes ao processo de produção de jogos para *PC*. As cinco seções contidas neste capítulo abordarão os seguintes temas:

Seção 1: Introdução

Apresenta-se a cadeia de valor na qual a produção de jogos está inserida.

Seção 2: Ciclo de Produção de Jogos para PC

Apresentam-se as principais fases presentes no desenvolvimento de jogos para plataforma *PC*. Fala-se também sobre os principais artefatos produzidos ao longo do processo.

Seção 3: Principais Papéis no Desenvolvimento de Jogos

Descrevem-se os principais papéis, presentes no processo de desenvolvimento de jogos para *PC*; assim como, as funções-chave de responsabilidade dos representantes de cada papel.

Seção 4: Principais Atividades Desempenhadas

Detalham-se as principais atividades referentes ao escopo das funções-chave de cada papel presente no ciclo de produção de jogos para *PC*.

Seção 5: Desafios do Ciclo de Produção de Jogos para Computador

Descrevem-se as principais dificuldades e características existentes na realidade atual do ciclo de produção de jogos para *PC*.

3.1 Introdução

Conceber, planejar, desenvolver e entregar um jogo, para plataforma *PC*, ao consumidor final é um processo complexo, geralmente longo e custoso [29]. São vários os atores envolvidos, com interesses e papéis distintos. Ou seja, os jogos para *PC* tendem a ser desenvolvidos por grandes times. São projetos de desenvolvimento complexos, caros e longos [28].




Para compreender bem o ciclo de produção de jogos para *PC* é preciso conhecer os principais atores e seus respectivos papéis na cadeia de valor de desenvolvimento de jogos.

3.1.1 Cadeia de Valor de Jogos para Plataforma PC

A cadeia de valor no mercado de jogos depende substancialmente de qual segmento de mercado está sendo considerado. Por exemplo: o segmento de jogos móveis costuma envolver operadoras, um ator não presente em segmentos como jogos para *console*.

Conforme já discutido previamente, este trabalho estará analisando em específico o segmento de jogos para plataforma *PC*. Neste mercado, classicamente, há cinco atores: o desenvolvedor, a publicadora, o distribuidor, o varejista e o consumidor [29]. Veja na Tabela 7, uma breve descrição de cada ator.

Tabela 7 – Atores da Cadeia de Valor de Jogos para PC

Ator	Descrição
 Desenvolvedor	Os desenvolvedores são os responsáveis pela criação de jogos, que transmitam ao consumidor final uma experiência imersiva, que inspirem este a buscar entretenimento em mundos virtuais; passando dos limites da realidade para os da fantasia.
 Publicadora	As publicadoras são articuladores responsáveis por levar os conceitos criados pelos desenvolvedores para o consumidor final. Buscam recursos necessários aos desenvolvedores e orientam estes a criarem conteúdo que atende às necessidades do público-alvo.
 Distribuidor	Os distribuidores representam um elo entre as publicadoras e os varejistas. São os responsáveis pelas atividades de logística que levem a uma melhor distribuição do jogo a partir de uma rede de varejistas.
 Varejista	Os varejistas são responsáveis pela venda direta do jogo ao consumidor final. É o ator da cadeia de valor mais visível ao consumidor final. Possuem um papel importante, uma vez que influenciam as opções de escolha a que o consumidor final tem acesso.
 Consumidor	Os consumidores finais são os públicos-alvo do jogo. Possuem um papel crucial, pois guiam as ações dos demais atores da cadeia de valor.

É importante destacar que algumas grandes empresas atuam na cadeia de valor como mais de um ator. Por exemplo, a *Electronic Arts* (www.ea.com) e a *Ubi Soft Entertainment* (www.ubi.com) atuam tanto como desenvolvedoras, como publicadoras.

Da concepção de um novo título até este chegar ao consumidor final, há geralmente três etapas principais: 1) a etapa de criação e publicação de um jogo; 2) a etapa de distribuição e venda, e 3) a etapa final de consumo.

Participam mais ativamente da primeira etapa os seguintes atores: desenvolvedor e publicadora. O processo de concepção de um novo jogo geralmente parte do desenvolvedor. Neste caso, o desenvolvedor, de posse de uma idéia que ele acredita ser viável e economicamente interessante, procura publicadoras que acreditem no potencial comercial do título. Contudo, também costuma ocorrer o processo inverso: uma publicadora procura desenvolvedores interessados em produzir certa idéia.

Participam mais ativamente da segunda etapa os seguintes atores: distribuidor e varejista. Esta etapa envolve o processo de logística que leva o jogo das publicadoras até as prateleiras de várias lojas varejistas.

Da última etapa, participa mais ativamente o ator consumidor. É neste momento que o consumidor final compra o jogo. A receita gerada pela venda é então repassada retroativamente para o varejista, o distribuidor, a publicadora e, por fim, para o desenvolvedor. Este processo clássico pode ser visualizado na Figura 57.

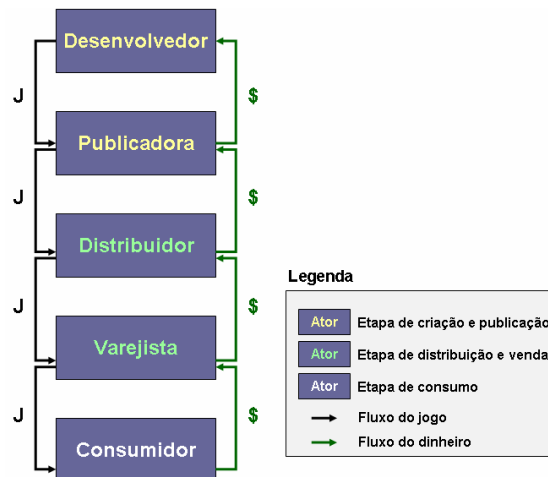


Figura 57 – Atores e Etapas da Cadeia de Valor Clássica de Jogos para PC

Uma visão mais completa da cadeia de valor apresentada na Figura 62 envolve a entrada de dois novos atores: as agências de publicidade e a mídia. Atualmente, antecede e acompanha o lançamento dos principais títulos, uma intensa e maciça campanha de marketing.

Neste cenário mais completo, a publicadora fornece recursos para o desenvolvedor entregar-lhe o jogo. Esta encaminha o produto final para o canal de distribuição (distribuidor e varejista), mas também se preocupa em fornecer materiais promocionais que estimulem a venda do produto final. Esta cadeia de valor mais elaborada pode ser visualizada na Figura 58.

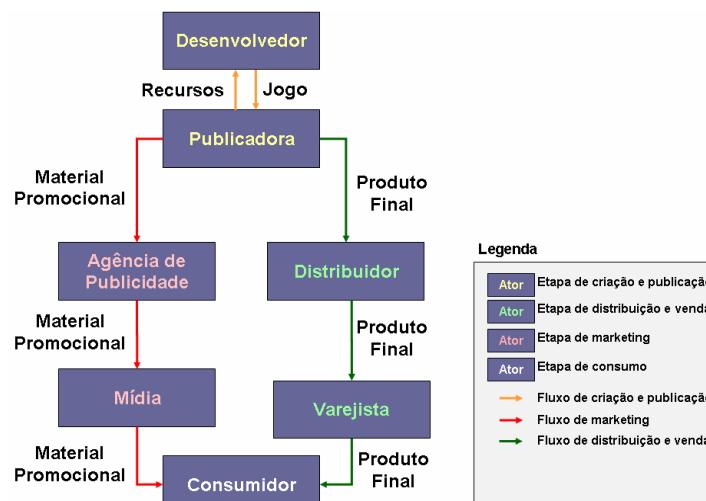


Figura 58 – Cadeia de Valor de Jogos para PC

Recentemente, com o advento da publicação *on-line*, tem-se destacado uma variação das etapas da cadeia de valor apresentada anteriormente. A idéia neste novo cenário é publicar o jogo de forma *on-line*, diretamente ao cliente final; eliminando, portanto, a intermediação do distribuidor e do varejista.

Desta forma, publicadoras *on-line*, como a TryGames (www.trygames.com), e desenvolvedores conseguem maiores margens de lucro pela eliminação da fatia consumida pelo distribuidor e varejista.

Em alguns casos, o próprio desenvolvedor publica o seu jogo de forma *on-line*. Chega-se, portanto, até a eliminar o papel da publicadora. Logo, nestes casos, a cadeia de valor apresentada na Figura 57 passa a ter uma forma mais próxima da presente na Figura 59 e 60.

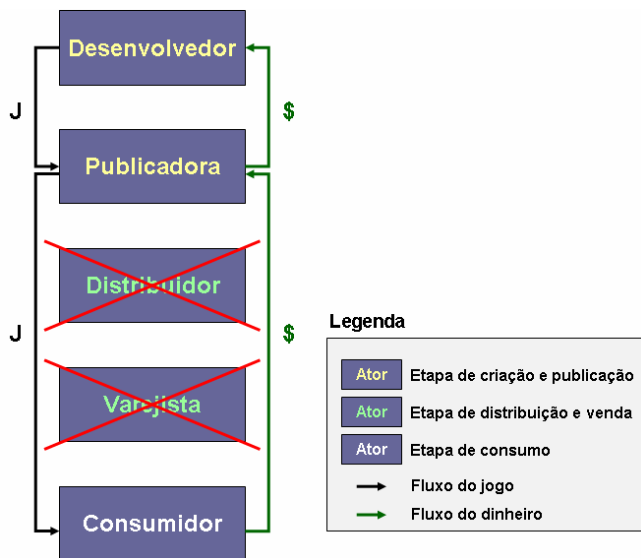


Figura 59 – Alternativa 1 para Cadeia de Valor Clássica de Jogos para PC

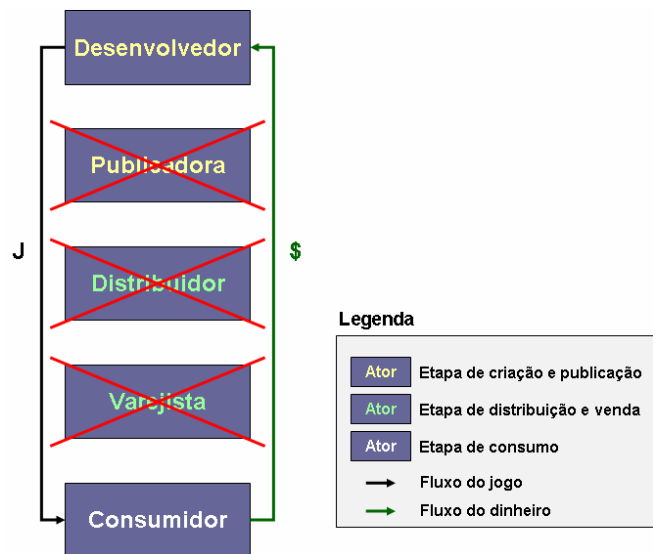


Figura 60 – Alternativa 2 para Cadeia de Valor Clássica de Jogos para PC

3.2 Ciclo de Desenvolvimento de Jogos para PC

Como foi visto anteriormente, o ciclo de desenvolvimento é parte da cadeia de valor de jogos. É neste estágio que as idéias de desenvolvedores e publicadoras são transformadas em algo mais concreto: o jogo.

De acordo com Sloper, há cinco principais fases no processo de desenvolvimento de um jogo: concepção, pré-produção, produção, pós-produção e pós-lançamento [28]. As próximas subseções trarão detalhes sobre cada uma destas fases.

3.2.1 Concepção


A primeira fase do desenvolvimento é a fase de concepção. O marco do seu início é a decisão, do desenvolvedor e/ou publicadora, de iniciar o desenvolvimento de um novo jogo.

É preciso haver um entendimento das partes envolvidas com o desenvolvimento e publicação do jogo a cerca da viabilidade operacional e comercial do produto [28].

Geralmente esta decisão encontra-se documentada em um artefato chamado *Game Proposal Document*².

Esta fase é guiada por decisões lógicas, comercialmente embasadas no conhecimento prévio do desenvolvedor e da publicadora. O objetivo principal é definir um conceito para o jogo que seja comercialmente interessante. O principal artefato produzido nesta fase é o *Conceptual Document* [28], detalhado na Tabela 8.

Tabela 8 – Conceptual Document

Documento	Características
 Conceptual Document	O propósito principal do <i>Conceptual Document</i> é comunicar uma visão unificada do jogo que todas as partes engajadas com o desenvolvimento concordem.
	O <i>Conceptual Document</i> deve descrever de forma sucinta e objetiva os principais elementos e personagens presentes no jogo.

O comum acordo das partes envolvidas com o desenvolvimento e publicação dos termos presentes no *Conceptual Document* marcam o início da produção propriamente dita. O processo de produção costuma ser dividido em três fases: pré-produção, produção e pós-produção.


3.2.2 Pré-Produção

Esta fase tem como objetivo reunir e definir os elementos necessários ao início efetivo da produção do jogo. É nesta fase que um time de desenvolvimento é definido e alocado ao projeto [28].

Termos de confidencialidade e não-divulgação também são assinados nesta etapa. Precisa estar claro nesta fase o gênero do jogo, o público-alvo e com quais outros títulos o jogo em questão irá competir.


O próximo passo é produzir os dois principais artefatos desta fase: o *Game Design Document* [30], também conhecido como GDD, e o *Technical Design Document* [28], também conhecido como TDD (Tabelas 9 e 10).

Tabela 9 – Game Design Document

Documento	Características
 Game Design Document	O principal propósito do GDD é apresentar uma visão completa e clara do jogo, descrever seus elementos e apresentar um plano de desenvolvimento.
	O GDD precisa descrever o jogo e seus elementos em um nível de detalhe que permita uma posterior especificação técnica do título.

² Os nomes dos artefatos estão em inglês por serem assim melhor conhecidos no mercado de jogos

Tabela 10 – Technical Design Document

Documento	Características
 Technical Design Document	O TDD é a especificação técnica da implementação das idéias apresentadas no GDD. Seus principais objetivos são: guiar a atividade de programação do jogo, identificar os desafios técnicos e determinar que ferramentas de desenvolvimento serão utilizadas.
	Os elementos descritos no GDD e tecnicamente detalhados no TDD ajudam na definição de uma lista de ações e <i>milestones</i> . Estas informações são utilizadas para definir o cronograma de desenvolvimento.

De posse do cronograma de desenvolvimento e da lista de ações, a produção propriamente dita do jogo pode ser iniciada.

3.2.3 Produção

Esta é a fase que concerne ao trabalho efetivamente realizado para produzir o código, a arte e o áudio do jogo. Vários são os artefatos produzidos e monitorados constantemente, por exemplo: o *Art List Document* e o *Sound List Document* [28]. Maiores detalhes destes documentos podem ser vistos na Tabela 11 e 12.

Tabela 11 – Art List Document



Documento	Características
 Art List Document	Este documento é criado a partir de informações contidas no GDD. Devem estar presentes neste documento informações sobre o que a equipe de arte deve criar. Detalhes dos personagens e dos cenários devem ser providos à equipe de arte.
	Este documento é constantemente atualizado e pode ser utilizado para controlar a criação da arte do jogo. Ou seja, este documento deve ser atualizado à medida que equipe de arte crie elementos do jogo.

Tabela 12 – Sound List Document

Documento	Características
 Sound List Document	Este documento é bastante similar ao <i>Art List Document</i> ; uma vez que fornece informações sobre o que a equipe de áudio deve criar. Detalhes dos efeitos e trilhas sonoras estão presentes neste documento.
	Os efeitos e trilhas sonoras do jogo estão diretamente associados com os eventos e ações que ocorrem no fluxo do jogo. Ou seja, este documento precisa estar coerente com os elementos do jogo definidos no GDD.

É importante deixar claro que o TDD também sofre atualizações e atinge um grau ainda maior de detalhes nesta fase.

Além das preocupações envolvidas com a criação do código, da arte e do áudio; fazem parte desta fase atividades clássicas presentes no dia-a-dia da gerência de projetos: identificar e antecipar problemas na produção, motivar o time de desenvolvimento, monitorar os custos do projeto, analisar e controlar as necessidades de mudanças e minimizar atrasos no cronograma.

Pode-se, portanto, observar que esta fase é uma das mais complexas do processo de produção como um todo. Há uma grande quantidade de pessoas envolvidas, realizando atividades complexas e com um alto grau de detalhes técnicos e interdependências.

Algumas empresas costumam estruturar a fase de produção na criação iterativa e evolutiva de vários protótipos. Ou seja, evoluir de um primeiro protótipo inicial até o jogo final. Esta abordagem na área de jogos é conhecida como *Cerny Method* em homenagem a Mark Cerny (Figura 61), presidente da *Cerny Games*.



Figura 61 – Mark Cerny

3.2.4 Pós-Produção

Após ter criado e integrado todos os elementos do jogo, inicia-se a fase de pós-produção. É neste momento que há uma maior preocupação do departamento de qualidade em testar o jogo, identificar *bugs* e locais que precisem de ajustes para melhorar a jogabilidade [28].

Em paralelo, o departamento de marketing da empresa preocupa-se em criar material promocional para acompanhar o lançamento do título.

Por fim, faz-se um *post mortem* do projeto de desenvolvimento do jogo com o objetivo de analisar os pontos positivos e negativos ocorridos ao longo da criação do título.

Maiores informações sobre os principais artefatos produzidos nesta fase podem ser vistas na Tabela 13 e 14.

Tabela 13 – Test Plan



Documento	Características
 Test Plan	Analisando as especificações contidas no GDD, no TDD e o estado atual do jogo, determinam-se quais testes precisam ser realizados, que técnicas e procedimentos serão utilizados e quanto tempo os testes irão durar.
	Costuma-se dividir os testes em duas categorias: <i>alpha</i> e <i>beta</i> . Os primeiros geralmente envolvem equipes de testes internas à empresa e à publicadora. Já os testes <i>beta</i> procuram envolver consumidores finais.

Tabela 14 – Post Mortem Document

Documento	Características
 Postmortem Document	Após a conclusão do desenvolvimento do jogo, cria-se um documento detalhado com três seções principais: pontos positivos, pontos negativos e lições aprendidas com o desenvolvimento do título em questão.
	O principal objetivo do <i>Postmortem Document</i> é documentar de forma clara e objetiva os acertos e erros do projeto. Pretende-se com esta prática, repetir e maximizar os acertos em projetos futuros; ao passo que minimizar e evitar os erros cometidos.

3.2.5 Pós-Lançamento

De acordo com Sloper, a fase de pós-lançamento ocorre após a fase de pós-produção. O jogo já foi lançado e encontra-se no mercado. Neste momento o desenvolvedor e, principalmente, a publicadora devem observar e monitorar a reação do consumidor ao título desenvolvido; pois, podem existir oportunidades para expansões do jogo e seqüências [31].

Sloper divide as ações realizadas nesta fase em cinco grupos:

1. Suporte às Vendas
O varejista pode ter questionamentos a cerca do jogo oriundas dos consumidores. Alguns destes questionamentos podem voltar para a publicadora e devem ser respondidos e analisados. Estes questionamentos podem indicar ameaças e oportunidades para o jogo e para o modelo de negócio.
2. Suporte ao Consumidor
É comum o consumidor final ter dúvidas sobre como usar o jogo. Podem ser dúvidas em relação a elementos de configuração e instalação. A publicadora deve criar canais para estas dúvidas fluírem dos consumidores até a publicadora. Estes questionamentos podem indicar, por exemplo, falhas na criação do manual do jogo.
3. Mais versões
A partir da reação do mercado ao título produzido, o desenvolvedor e a publicadora podem decidir produzir novas versões do jogo para outras línguas (localização) ou até mesmo para outras plataformas.
4. Expansões
Para alguns tipos específicos de jogos, o desenvolvedor e a publicadora podem aumentar o retorno financeiro de um título ao criar novas missões, cenários e personagens para este.
5. Seqüências
Caso o jogo faça um grande sucesso com o público-alvo, talvez seja interessante explorar a idéia do jogo em seqüências. Um exemplo clássico é o jogo *Final Fantasy* que possui mais de 12 seqüências.

Portanto, entende-se por pós-lançamento, uma última fase do ciclo de desenvolvimento de um jogo preocupada em coletar e analisar o *feedback* do mercado para título lançado. Esta fase pode até incentivar o desenvolvimento de um novo

título; retornando, assim, ao início do ciclo. As informações coletadas nesta fase podem ser documentadas em um artefato chamado de *Market Feedback Document*.

3.3 Principais Papéis no Desenvolvimento de Jogos

Na seção anterior apresentou-se uma visão geral do ciclo clássico de desenvolvimento de jogos, em específico, para plataforma *PC*.

As funções e atividades, presentes em cada uma das fases deste processo, são normalmente desempenhadas por times grandes e heterogêneos. Nesta seção, serão apresentados os principais papéis presentes neste ciclo, focando no ator desenvolvedor.





A equipe do *Gamasutra*, parte do *CMP Game Group*, faz periodicamente uma análise dos salários praticados na indústria de jogos digitais. O estudo mais recente [32] classifica os cargos neste mercado em seis categorias.

Logo, estas categorias serão consideradas como os mais representativos papéis dentro do ciclo de criação de jogos. Estes são: produtor, artista, *game designer*, programador, engenheiro de áudio e engenheiro de qualidade. A Tabela 15 e 16 apresenta informações sobre as funções destes papéis.

Tabela 15 – Papéis no Desenvolvimento de Jogos

Papel	Descrição
 <p>Produtor</p>	O produtor é a pessoa responsável por planejar e controlar o desenvolvimento do jogo. Possui uma visão macro do projeto e envolve-se com todos os aspectos da criação deste [33].
	Suas responsabilidades são parecidas com as de um gerente de projeto. É responsável por garantir que o jogo seja desenvolvido conforme a especificação, o cronograma, o orçamento e com um nível de qualidade aceitável [33].
	O salário médio do produtor, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 65.000 anuais [32].
 <p>Artista</p>	O artista é a pessoa responsável por produzir o conteúdo de arte do jogo. Há várias especializações de artista dentro da equipe de desenvolvimento. Por exemplo: artista 2D e artista 3D.
	Dependendo da especialidade do artista, desempenha funções variadas como: modelagem tridimensional, criação de texturas, animação de personagens, etc.
	O salário médio do artista, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 50.000 anuais [32].

Tabela 16 – Papéis no Desenvolvimento de Jogos

Papel	Descrição
 Game Designer	<p>O papel do <i>game designer</i> é guiar, desenvolver e documentar o fluxo do jogo. É um papel que exige habilidades diversas como alto grau de abstração e criatividade, entre outras.</p>
	<p>Realiza atividades como: definir os elementos presente no jogo, a mecânica do fluxo do jogo, criar elementos centrais da história do jogo, etc.</p>
	<p>O salário médio do <i>game designer</i>, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 60.000 anuais [32].</p>
 Programador	<p>O programador é a pessoa responsável por codificar os elementos necessários para viabilizar a execução do jogo. Há várias especialidades de programadores dentro da equipe de desenvolvimento.</p>
	<p>Dependendo da especialidade do programador, desempenha funções variadas como: codificação do sistema gráfico, da arquitetura de comunicação, do fluxo do jogo, etc.</p>
	<p>O salário médio do programador, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 70.000 anuais [32].</p>
 Engenheiro de Áudio	<p>O engenheiro de áudio é a pessoa responsável por desenvolver os elementos sonoros do jogo: efeitos sonoros e trilhas sonoras.</p>
	<p>Desempenha atividades como adequar a trilha sonora ao fluxo e à história do jogo, determinar e criar os efeitos sonoros, etc.</p>
	<p>O salário médio do engenheiro de áudio, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 42.000 anuais [32].</p>
 Engenheiro de Qualidade	<p>O engenheiro de qualidade é o responsável por políticas que levem o jogo a ter um nível de qualidade de acordo com os anseios do mercado.</p>
	<p>Desempenha atividades como planejar, monitorar e controlar testes de codificação, de arte, de jogabilidade, de usabilidade, etc.</p>
	<p>O salário médio do engenheiro de qualidade, de acordo com pesquisa feita pelo <i>Gamasutra</i> em 2003, é de aproximadamente US\$ 62.000 anuais [32].</p>

Os papéis acima listados devem ser vistos como os principais dentro do ciclo de desenvolvimento de um jogo. Contudo, há outros papéis também presentes neste processo; papéis estes, às vezes, especializações dos acima. Por exemplo: o líder de arte, o *level designer*, o escritor, etc.

As informações chaves referentes às etapas de desenvolvimento de um jogo e os principais papéis envolvidos neste processo encontram-se resumidas na Figura 62.

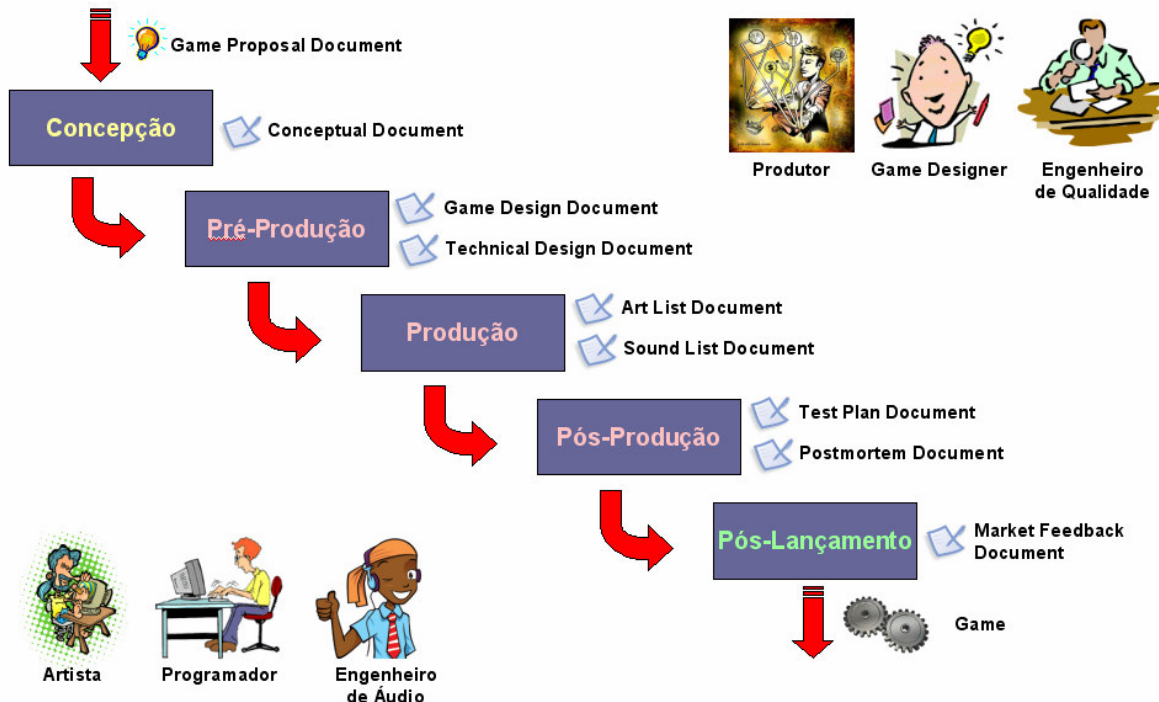


Figura 62 – Etapas e Papéis do Ciclo de Desenvolvimento de Jogos

Na próxima seção, serão detalhadas as principais funções desempenhadas pelos papéis aqui descritos.

3.4 Principais Atividades Desempenhadas

As atividades desempenhadas, pelos papéis apresentados na seção anterior, variam consideravelmente dependendo do tamanho do projeto e da empresa. Por exemplo: em certas circunstâncias, o projeto pode envolver atividades como captura de movimento; o que provavelmente não ocorrerá na produção de um pequeno jogo.

As atividades que serão apresentadas neste capítulo são as consideradas mais representativas das responsabilidades de cada papel envolvido no ciclo de desenvolvimento de um jogo.

Tomou-se como base aqui as atividades enumeradas pelos principais representantes de empresas de jogos do estado de Pernambuco, em encontro realizado no início de 2006 [34].

3.4.1 Produtor

Como já foi dito anteriormente, o produtor é a pessoa responsável por planejar e controlar o desenvolvimento do jogo. Logo, suas atividades são semelhantes com as de um gerente de projeto.

O produtor precisa estar atento às novidades e tendências do mercado de jogos. Precisa conhecer e saber lidar com as variáveis que podem levar um título a ser bem aceito pelo consumidor. Portanto, faz parte das suas atribuições analisar as propostas de jogos que surgem e chegam à empresa quanto a sua viabilidade técnica e comercial.

Também faz parte das suas funções participar ativamente da construção do *Conceptual Document*; uma vez que este precisa fornecer uma visão clara e unificada do jogo para todas as partes envolvidas com a sua produção.

Como um gerente de projeto, também atua na elaboração de cronogramas de projeto, previsão de recursos de infra-estrutura e humanos necessários ao desenvolvimento de um jogo, assim como na construção do orçamento do projeto.

O produtor também está presente durante a fase de produção: faz parte das suas atribuições gerenciar o desenvolvimento do jogo. Portanto, preocupa-se em administrar e alocar os recursos do projeto e garantir o fluxo de comunicação entre os membros do time de desenvolvimento.

Dependendo do tamanho do projeto e da equipe de desenvolvimento, pode haver um cenário no qual exista um produtor líder gerenciando outros produtos; estes, com funções mais gerenciais e acompanhando mais de perto a execução do projeto.

Por fim, também faz parte do papel do produtor analisar o *feedback* do mercado em relação aos jogos desenvolvidos pela empresa. Desta forma, ele poderá direcionar os próximos projetos para serem melhor aceitos no mercado.

3.4.2 Artista

O time de desenvolvimento de jogos para *PC* é composto, na maioria dos casos, por uma quantidade bem maior de artistas do que de quaisquer outros papéis. A tendência da indústria de jogos é cada vez mais especializar as funções destes. Logo, vários são os profissionais considerados artistas. Por exemplo: o responsável por modelar objetos tridimensionais, criar texturas, animar os personagens, etc.

Apesar das diferenças existentes em cada especialização, faz parte do trabalho dos artistas, de uma forma geral, adequar o material artístico produzido por este com o ambiente do jogo definido no *Game Design Document*.

Antes de iniciar o processo de criação de conteúdo artístico, os artistas projetam a arte do jogo tecnicamente. Isto envolve atividades como projetar a interface gráfica do jogo (Figura 63), criar *storyboards*³ (Figura 64) e arte conceitual⁴ (Figura 65).



Figura 63 – Interface Gráfica do Jogo Insignia

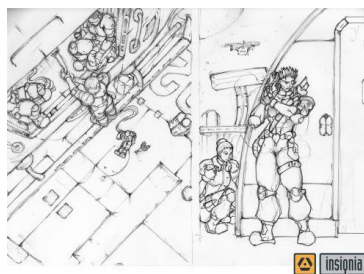


Figura 64 – Storyboard do Jogo Insignia



Figura 65 – Arte Conceitual do Jogo SWAT3

³ Sequência de quadros desenhados com o objetivo de passar a idéia da história do jogo.

⁴ Esboços de cenários e personagens que serão criados posteriormente.

A criação da arte propriamente dita pode ser dividida em cinco grandes áreas: a criação da arte tridimensional (3D), da arte bidimensional (2D), das animações 3D, das animações 2D e da interface gráfica.

A criação da arte 3D envolve criar o conteúdo tridimensional presente no jogo: desde os cenários, os objetos presentes nestes até os personagens (Figura 66).



Figura 66 – Modelo 3D de Personagem Presente na *Unreal Engine 3*

A criação da arte 2D envolve atividades como a criação das ilustrações presentes no jogo como também a criação das texturas que irão compor os cenário, os objetos e os personagens. Veja na Figura 67 o modelo 3D da Figura 66 com texturas aplicadas e efeitos de iluminação.

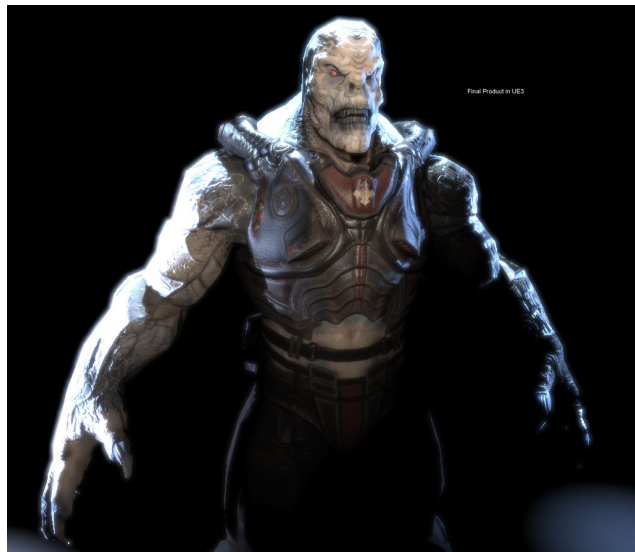


Figura 67 – Modelo 3D de Personagem com Textura Presente na *Unreal Engine 3*

A criação de animações bidimensionais ou tridimensionais utiliza diversas técnicas para trazer animação para os elementos do jogo. Estas técnicas podem ser

das mais simples, como desenho de quadros seqüenciais, até técnicas complexas, como captura de movimento. Veja na Figura 68 o *screenshot* de um personagem sendo animado através do *software 3D Studio Max*.

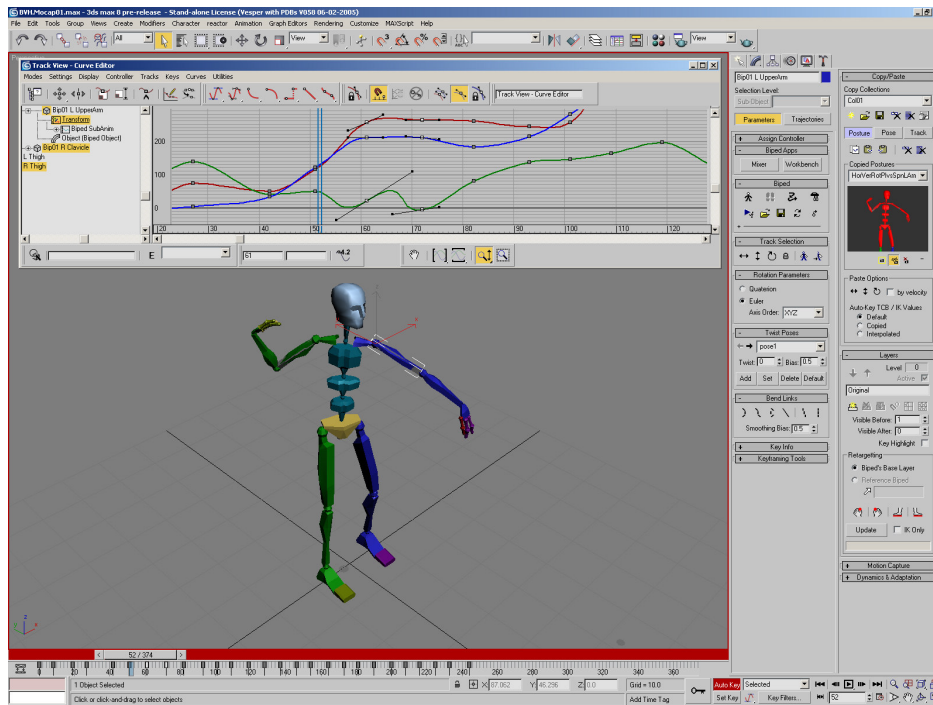


Figura 68 – Personagem Sendo Animado no 3D Studio Max

Por fim, a equipe de arte também é responsável por projetar e criar os elementos que constituem a interface gráfica do jogo. O conjunto de todas as atividades aqui listadas, e outras, é que permite a arte do jogo sair do *storyboard* para o seu estado final. Veja na Figura 69 e 70 o *storyboard* e a arte final, respectivamente, de uma cena.



Figura 69 – Storyboard do Jogo Oz – The Magical Adventure



Figura 70 – Arte Final de Cena do Jogo Oz – The Magical Adventure

3.4.3 Game Designer

O *game designer* tem como principal preocupação fazer com que o jogo, no final, tenha os elementos certos dosados de forma coerente; para, desta forma,

conseguir conquistar o interesse do consumidor em imergir no ambiente criado virtualmente.

Para atingir tal objetivo normalmente complexo, o *game designer* interage diretamente com todo o time de desenvolvimento: artistas, produtores, programadores, etc.

Uma das principais atividades do *game designer* é a criação, o controle e a manutenção do GDD. Este é um documento que precisa ser muito bem definido na fase de pré-produção para permitir um melhor desempenho da fase de produção. Também é importante atualizar este documento com eventuais mudanças ocorridas ao longo do desenvolvimento e comunicar e manter todo o time ciente das atualizações do GDD.

A criação do GDD envolve um conjunto de tarefas como: pesquisar referências conceituais para o jogo, desenvolver o roteiro do jogo, pensar nos elementos presentes no fluxo do jogo, definir os cenários e fases, etc.

O *game designer* também tem papel fundamental no momento de implementação do *level design*: a integração do trabalho de arte com o de programação para montar as fases do jogo.

Por fim, o *game designer* também acompanha de perto os testes, principalmente os de usabilidade e jogabilidade, para verificar a conformidade do jogo com as premissas e expectativas definidas no GDD.

Em grandes projetos, costuma-se dividir as muitas atividades do *game designer* em cargos mais especializados e focados. Por exemplo, é comum haver um *level designer* que pode ser visto como um *game designer* mais focado na definição e integração das fases do jogo.

3.4.4 Programador

O programador é um papel que engloba todas as atividades relacionadas com a implementação técnica do jogo. Ou seja, estão dentro do escopo do papel programador todos os membros do time envolvidos com áreas como: programação de baixo e alto nível, engenharia de comunicação e sistemas, gerenciamento de bancos de dados, etc.

As atividades de um programador variam bastante dependendo da sua especialidade. Por exemplo: programadores especializados na criação de ferramentas de apoio ao desenvolvimento do jogo e no desenvolvimento de *game engines*⁵ (Figura 71).

Também faz parte do papel do programador participar da definição dos requisitos técnicos do jogo; como também, do projeto dos componentes de entrada e saída, de rede, gráficos, de áudio e físicos. Informações estas que são devidamente documentadas no *Technical Design Document*.

Outra importante responsabilidade atribuída ao programador é permitir a integração do conteúdo produzido por outros papéis (artistas e engenheiros de áudio) em torno de uma arquitetura única. Esta é uma atividade crítica no ciclo de desenvolvimento e se não bem planejada e dimensionada, pode desestruturar por completo o planejamento do projeto.

⁵ É uma plataforma de desenvolvimento para jogos. Normalmente cuida do sistema gráfico e de outros sistemas como o de colisão, inteligência artificial, rede, edição de terreno e cenário, etc.

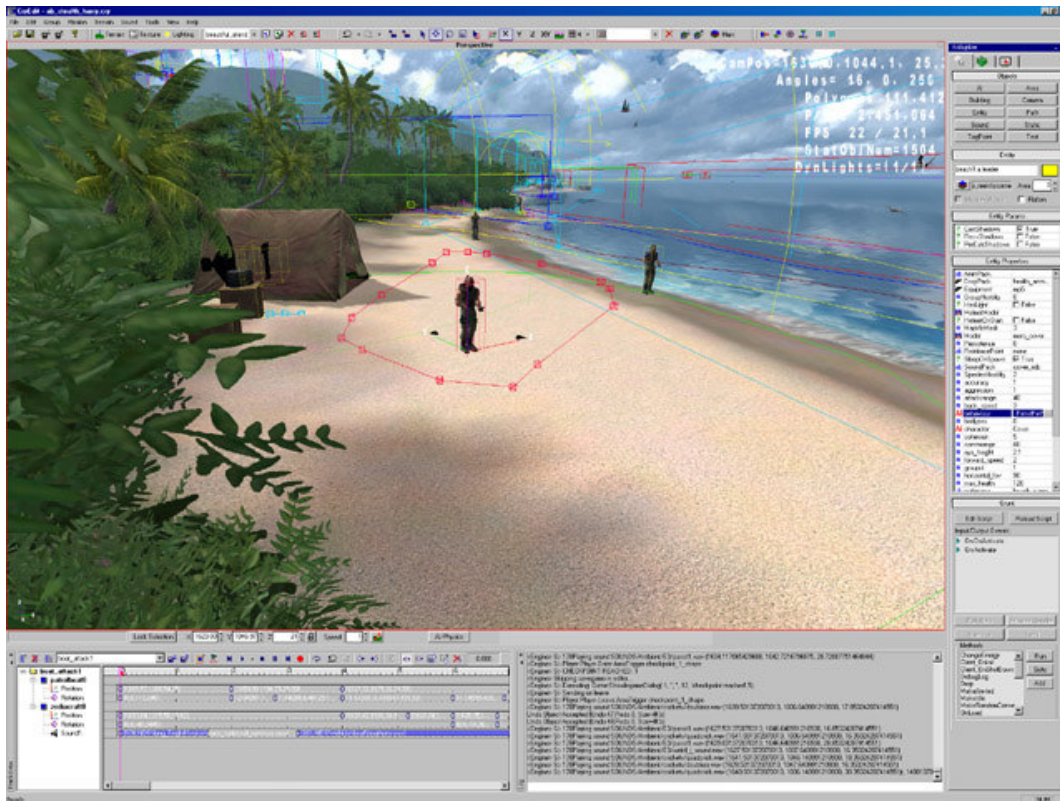


Figura 71 – Screenshot da Crytek Cryengine

3.4.5 Engenheiro de Áudio

Como o nome do papel sugere, o engenheiro de áudio é o responsável pelas atividades de áudio do jogo. Geralmente são especialistas formados em música e áreas afins que se preocupam com a criação dos efeitos sonoros do jogo, composição das trilhas musicais, gravação e edição das vozes de personagens, etc.

Este é um papel no ciclo de desenvolvimento de jogos que costumava ser bastante terceirizado. Atualmente, com o crescimento do mercado de entretenimento digital e a existência de grandes empresas desenvolvendo vários projetos em paralelo, alguns grandes desenvolvedores estão começando a optar por manter uma equipe de áudio vinculada tempo integral à empresa.

É importante frisar que o trabalho do engenheiro de áudio anda muitas vezes em paralelo com o da equipe de arte. Por exemplo: o efeito de áudio criado para o latido de um cachorro pode guiar a criação da animação do cachorro latindo, para esta casar com a duração do efeito.

Além deste aspecto, uma boa composição da arte do jogo com efeitos e trilhas sonoras bem concebidas pode levar o consumidor a se envolver mais com o ambiente virtual criado. Por exemplo: um cenário, cuja arte retrata um ambiente escuro e misterioso, em harmonia com uma trilha sonora, que cria um clima de tensão, pode estimular uma sensação de tensão grande no jogador.

3.4.6 Engenheiro de Qualidade

O papel do engenheiro de qualidade tem crescido significativamente nos últimos anos na indústria de entretenimento digital. No início, as atividades deste papel se

resumiam ao planejamento e execução das atividades de teste em parceria com representantes de outros papéis.

Contudo, acompanhando a tendência da produção de software, há uma preocupação crescente com os aspectos de qualidade envolvidos com o jogo e com a sua produção.

Atualmente, este papel é mais visto com a atribuição de fazer com que o jogo atenda aos anseios da publicadora e do consumidor, minimizando os custos com falhas. Ou seja, o papel evoluiu e hoje engloba atividades mais complexas do que simples testes. Por exemplo: o foco em usabilidade, testes mais elaborados e elementos de jogabilidade.

Dependendo das necessidades do projeto, este papel pode ser especializado em outros papéis como: equipe focada em realizar testes, em documentar e otimizar o processo de desenvolvimento, etc.

3.5 Desafios do Ciclo de Produção de Jogos de Computador

De acordo com McConnell, *softwares* como sistemas comerciais, aplicações para internet e jogos demandam práticas diferentes dependendo das prioridades do seu mercado alvo. Por exemplo, estes *softwares* diferem em variáveis como tempo de produção, consequência dos erros, número de disciplinas (arte, música e código) envolvidas e trabalho de aperfeiçoamento necessário [35].

Quase 100% dos projetos apresentam atrasos no seu cronograma, cerca de um quarto dos projetos são cancelados e somente percentual similar é entregue dentro do prazo. As razões para estes problemas são muitas e complicadas. Contudo, podem-se generalizar estas causas como sendo ineficiências reais e persistentes decorrentes de projetos realizados de maneira ineficiente [35].

O ciclo de produção de jogos para computador, como parte da realidade de desenvolvimento de *software*, também possui os mesmos problemas. São evidentes as grandes mudanças ocorridas no mercado de entretenimento digital nos últimos anos; contudo, os processos de desenvolvimento não acompanharam estas mudanças no mesmo ritmo. Logo, criou-se um cenário de dificuldades crescentes.

É nesta conjuntura que nasce a necessidade de modelos que construam e disseminem padrões de desenvolvimento. Padrões que criem uma estrutura mais sólida e melhor organizada para lidar e resistir às dificuldades emergentes da produção de jogos para computador.

Portanto, os modelos de desenvolvimento de jogos para computador precisam considerar, gerenciar e controlar as principais características presentes nesta vertente do desenvolvimento de *software*. As principais características do ciclo de produção de jogos para computador são apresentadas na Tabela 17, 18 e 19.

Tabela 17 – Principais Características do Ciclo de Desenvolvimento de Jogos


Característica	Descrição
 Tamanho da Equipe	O time envolvido no desenvolvimento de um jogo para plataforma <i>PC</i> costuma envolver um grande número de colaboradores.
	Dependendo do tamanho do jogo, há relatos de times de desenvolvimento com mais de 200 pessoas.

Tabela 18 – Principais Características do Ciclo de Desenvolvimento de Jogos

Característica	Descrição
 <p>Composição da Equipe</p>	<p>São muitos os papéis envolvidos no desenvolvimento de um jogo.</p> <p>Com o aumento da complexidade dos títulos produzidos, também cresce o grau de especialização. Logo, o desenvolvimento de jogos costuma envolver times que são bastante heterogêneos.</p>
 <p>Tempo de Desenvolvimento</p>	<p>Dependendo da complexidade do jogo e de outros fatores, o tempo de desenvolvimento de um título pode variar de alguns poucos meses até vários anos.</p>
 <p>Caráter Iterativo e Incremental</p>	<p>O ciclo de desenvolvimento do jogo costuma ser realizado de forma iterativa e incremental. Ou seja, produz-se com certa frequência novas versões do jogo que evoluem progressivamente até o produto final.</p> <p>Esta característica é essencial; pois, permite que a publicadora, por exemplo, possa acompanhar e avaliar um título ao longo do seu desenvolvimento.</p>
 <p>Ocorrência de Mudanças</p>	<p>Como consequência do próprio caráter evolutivo do ciclo de desenvolvimento de um jogo, é comum ocorrerem mudanças em aspectos do <i>Game Design</i> ao longo da produção.</p>
 <p>Proporção dos Custos</p>	<p>Já que o processo de produção envolve equipes grandes, colaboradores especializados, longos períodos de desenvolvimento; os custos de desenvolvimento podem ser bastante altos.</p> <p>Há relatos de jogos que custaram mais de U\$ 40 milhões para serem desenvolvidos.</p>
 <p>Necessidade de Documentação</p>	<p>Documentar as etapas de produção é essencial para comunicar uma imagem comum do projeto para todos os colaboradores.</p> <p>O aspecto documentação também é importante para identificar aspectos positivos e negativos, no processo de desenvolvimento, que devem ser maximizados e minimizados, respectivamente, em projetos futuros.</p>

Tabela 19 – Principais Características do Ciclo de Desenvolvimento de Jogos

Característica	Descrição
 <p>Influência do <i>Game Design Document</i></p>	<p>O <i>Game Design Document</i> é o elemento central do processo de desenvolvimento. Ele guia as atividades das mais diversas áreas presentes na produção do jogo: arte, código, áudio, testes, etc.</p>
 <p>Necessidade de Integração</p>	<p>A atividade de integração é um fator crítico no processo de desenvolvimento. Além da integração dos vários componentes de código (IA, E/S, Gráfico, etc.), precisa-se também integrar o conteúdo de arte e áudio para formar um todo coerente.</p>
 <p>Necessidade de Reuso e Repetição</p>	<p>Como importante aspecto redutor dos custos de desenvolvimento, há uma preocupação crescente em reusar componentes e repetir boas práticas presentes em projetos passados.</p>

4. Metodologias de Desenvolvimento de Software

Este capítulo tem como objetivo apresentar as principais correntes de modelos de desenvolvimento de software. As quatro seções contidas neste capítulo abordarão os seguintes temas:

Seção 1: Introdução

Apresentam-se os principais motivos que levaram a necessidade de usar metodologias de desenvolvimento de *software*. Descrevem-se também as principais correntes de modelos de desenvolvimento de *software*.

Seção 2: Metodologias Ágeis

Descrevem-se as principais metodologias ágeis de desenvolvimento de software.

Seção 3: Metodologias Preditivas

Descrevem-se as principais metodologias preditivas de desenvolvimento de *software*.

Seção 4: Análise Comparativa

Faz-se uma análise comparativa para identificar que corrente e metodologia melhor atende às características presentes no desenvolvimento de jogos para PC.

4.1 Introdução

O mundo atual é extremamente dependente da tecnologia da informação. Cada vez mais, as organizações, os governos e as sociedades fazem uso e se beneficiam de sistemas computacionais. Logo, o desenvolvimento de *software* deixou de ser uma realidade restrita a resolver problemas dos programas militares para melhorar o dia-a-dia das pessoas.

É imprescindível neste cenário atual a produção de *software* confiável, correto e entregue dentro dos prazos e custos planejados. Um *software* com erros, entregue fora do prazo e dos custos planejados, pode trazer grandes prejuízos à sociedade.

Dados de 1995, investigando uma base de 8.380 projetos, indicam que apenas 16,2% dos projetos foram entregues respeitando os prazos e custos e com todas as funcionalidades previstas. Cerca de 31% dos projetos foram cancelados antes da sua conclusão, e dos projetos entregues, fora dos prazos e custos especificados, a média de atrasos foi de 222% e a média de custo foi de 189% a mais do que o previsto [36]. Além disso, muitos dos projetos entregues possuíam qualidade abaixo do esperado.

Portanto, pode-se dizer que o desenvolvimento de *software* é uma atividade com um alto risco associado. Esta atividade já gerou grandes prejuízos no passado e continua gerando no presente.

É neste cenário que nasce a necessidade de utilizar metodologias de desenvolvimento de software. O objetivo principal de uma metodologia de desenvolvimento de *software* é diminuir as falhas mais recorrentes presentes no processo de produção [37].

Então, a utilização de uma metodologia tem como objetivo produzir sistemas de alta qualidade e confiabilidade, aumentar a produtividade durante o desenvolvimento e ter controle sobre os custos, prazos, escopo e valor agregado para o cliente; além de, coletar e armazenar informações sobre como ocorreu o desenvolvimento para auxiliar no planejamento de projetos futuros. Para tanto, uma metodologia de

desenvolvimento de *software* define uma estrutura de processos envolvidos no desenvolvimento de um produtor de *software* [38].

Atualmente, existe uma grande variedade de metodologias de desenvolvimento de *software*. Cada uma destas possui características próprias que atendem melhor a cenários e mercados específicos. Apesar da grande variedade, elas podem ser agrupadas em algumas correntes. A próxima subseção apresentará quais correntes são estas.

4.1.1 Correntes de Modelos de Desenvolvimento de Software

As metodologias de desenvolvimento de *software* atuais podem ser agrupadas em duas principais vertentes: preditivas e adaptativas. A primeira vê a produção de *software* como uma atividade que tem fases bem definidas e delimitadas, assim como também acredita ser importante haver uma maior formalidade quanto à documentação do desenvolvimento. Já a segunda, tem como característica não ter fases isoladas e nem ser considerada tão formal qual à documentação.

Além desta dimensão (maior formalidade vs. menor formalidade), as metodologias podem adotar um modelo cascata ou podem ser mais iterativas. As metodologias que seguem o primeiro modelo costumam lidar com o desenvolvimento de sistemas que possuem poucos riscos. Desta forma, o desenvolvimento é linear e a integração e os testes são realizados somente ao final.

Já as metodologias mais iterativas, costumam lidar com o desenvolvimento de sistemas que o grau de risco associado é alto. Para minimizar os riscos, a integração de componentes e a realização de testes são realizadas de forma contínua.

Portanto, podem-se comparar as metodologias de desenvolvimento de *software* em duas dimensões: maior formalidade vs. menor formalidade e modelo cascata vs. modelo iterativo [39]. Veja na Figura 72 os quatro quadrantes formados por estas duas dimensões e quais metodologias são as principais representantes de cada um destes.

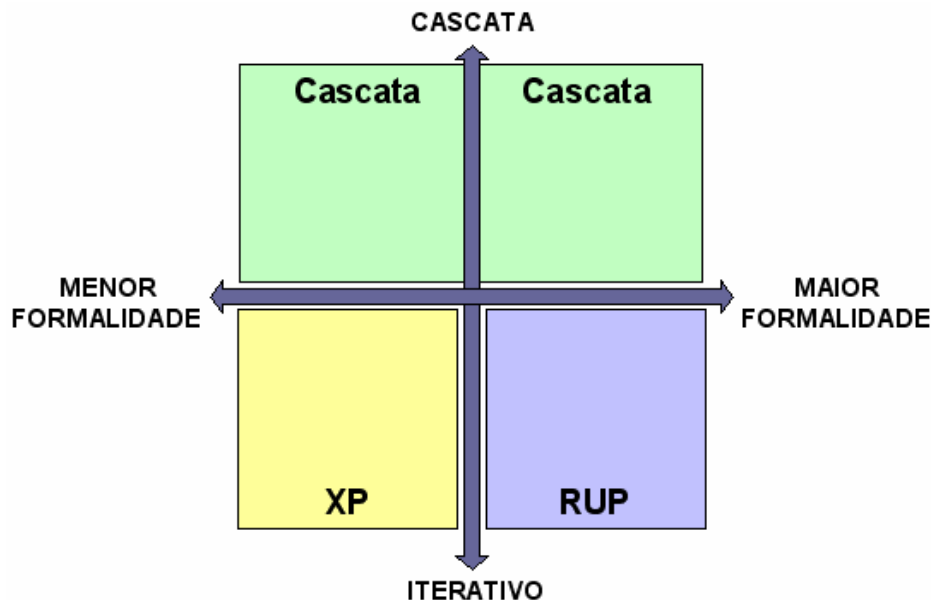


Figura 72 – Dimensões das Metodologias de Desenvolvimento de Software

Na Figura 72 acima, associou-se a cada quadrante uma metodologia: Cascata, XP e RUP.

O modelo cascata é um modelo que vê o desenvolvimento de *software* como a execução sequencial das seguintes fases: análise de requisitos, *design*, implementação, validação, integração e manutenção [40]. Ele pode ser instanciado tanto com alta ou baixa formalidade. Depende da necessidade do projeto de desenvolvimento considerado.

O RUP é um processo iterativo, criado pela *Rational Software Corporation*, preocupado com a documentação do desenvolvimento de *software* [41]. Contudo, há outros modelos de desenvolvimento considerados mais formais do que ele.

Já o XP é uma metodologia de desenvolvimento iterativa muito mais preocupada com a adaptabilidade do que com a preditividade, em detrimento da formalidade. Ou seja, é uma metodologia que possui um grau de formalidade menor.

Nas próximas seções será apresentada uma metodologia chave representativa dos modelos adaptativos ou ágeis (XP) e dos preditivos (RUP). Também serão descritas adaptações realizadas no XP e no RUP para o cenário de desenvolvimento de jogos: XGD e GUP, respectivamente. Também será vista uma adaptação do modelo cascata para o mercado de jogos (GWP).

4.2 Metodologias Ágeis

As metodologias ágeis surgiram como uma reação às metodologias excessivamente preditivas que eram estruturadas em torno de práticas extremamente regulamentadas, amarradas e inspiradas no modelo cascata [42].

Estas metodologias excessivamente preditivas ficaram conhecidas como metodologias pesadas. Foram assim denominadas por serem muitas vezes burocráticas.

No início, as metodologias ágeis eram conhecidas como metodologias leves. Em 2001, criou-se o *Agile Manifesto* e estas foram denominadas de metodologias ágeis. Posteriormente, formou-se a *Agile Alliance*, uma organização sem fins lucrativos com o objetivo de promover o desenvolvimento das metodologias ágeis.

A maioria das metodologias ágeis busca minimizar os riscos desenvolvendo o *software* em pequenos intervalos de tempo chamados de iteração. Cada iteração costuma durar entre 1 e 4 semanas. Cada iteração é como um subprojeto de desenvolvimento de *software* e, no final de cada, uma nova versão do *software* deve estar concluída. O próximo passo antes de começar uma nova iteração é reavaliar as prioridades do projeto.

As metodologias ágeis preferem a comunicação freqüente das informações pessoalmente a documentos escritos. Estas metodologias também frisam que o processo de desenvolvimento deve-se concentrar principalmente na produção efetiva do *software* e não em documentação. Portanto, é característica destas a pequena produção de documentos escritos.

Portanto, estas metodologias são recomendadas para situações em que ajam mudanças rápidas e freqüentes nos requisitos, desenvolvedores experientes e pequeno número de desenvolvedores [13]. Por estas características, estas metodologias são consideradas mais adaptativas.

Existem várias metodologias ágeis (*Scrum*, *Adaptive Software Development*, *Crystal Clear*, *Dynamic Systems Development Method*, *Feature Driven Development*, *Lean Software Development*); contudo, o *Extreme Programming* (XP) foi a responsável pela popularização desta família de metodologias.

As próximas subseções irão mostrar as principais características do XP; assim como do XGD (*Extreme Game Development*), uma adaptação do XP para o mercado de jogos.

4.2.1 Extreme Programming (XP)

O XP é uma metodologia de desenvolvimento de *software* que, como as demais metodologias ágeis, difere por dar mais valor à capacidade de adaptar do que à de prever. Os defensores do XP vêem a mudança de requisitos como uma constante no processo de desenvolvimento de *software* [43].

Define-se, portanto, o XP como sendo uma metodologia ágil para equipes pequenas e médias desenvolvendo *software* com requisitos vagos e com mudança constante [44]. É um processo que procura assegurar que o cliente ganhe o máximo de valor para cada dia de trabalho da equipe de desenvolvimento.

No XP, as funcionalidades do sistema são especificadas através de histórias. Ou seja, histórias são descrições das necessidades do cliente descritas por este, com suas próprias palavras.

As histórias são implementadas durante o desenvolvimento do projeto que costuma ser dividido em *releases* e em iterações. *Releases* são módulos do sistema que englobam um conjunto representativo de histórias realizadas. Portanto, possuem um valor agregado bem definido para o cliente. As iterações são períodos curtos de tempo, na média em torno de duas semanas, no qual a equipe implementa um conjunto de funcionalidades mais importantes de acordo com o cliente.

O processo de desenvolvimento de *software* no XP é guiado por um conjunto de valores, princípios e práticas que atuam de forma harmônica e coesa para garantir que o cliente receba um alto retorno do investimento realizado no desenvolvimento do *software* em questão.

O XP possui cinco valores: comunicação, simplicidade, *feedback*, coragem e respeito. Veja na Tabela 20 e 21 uma breve descrição de cada um destes valores.

Tabela 20 – Valores do XP






Valor	Descrição
 <p>Comunicação</p>	Quando problemas aparecem durante o desenvolvimento, geralmente alguém da equipe sabe como solucioná-lo; contudo, este conhecimento nem sempre chega à pessoa que se encontra no dilema do problema. Comunicação é importante para criar o sentimento de time e cooperação [44].
 <p>Simplicidade</p>	A idéia é fazer o sistema simples o suficiente para conseguir resolver, de acordo com as expectativas, um dado problema. Simplicidade tem a ver com eliminar requisitos não necessários no presente e deixar para se preocupar com eles no futuro [44].
 <p>Feedback</p>	Mudanças é algo que ocorre durante o projeto. <i>Feedback</i> é o cliente comunicar ao desenvolvedor algo de novo que ele aprendeu sobre o problema, é o desenvolvedor comunicar ao cliente estimativas, riscos e melhorias do projeto. A idéia é que o quão mais cedo você sabe, o quanto antes você pode adaptar [44].

Tabela 21 – Valores do XP

Valor	Descrição
 Coragem	Coragem é ter ações efetivas para superar dificuldades. Quando combinada com outros valores, torna-se uma arma poderosa: coragem para comunicar, para manter simples e para ouvir o <i>feedback</i> [44].
 Respeito	Toda a equipe de desenvolvimento precisa ter respeito com os demais. É preciso que cada um se importe com o projeto; caso contrário, não terá como este não fracassar [44].

Os princípios do XP são baseados nos valores descritos acima e têm como objetivo guiar as decisões durante o desenvolvimento do *software*. Estes princípios buscam ser mais concretos que os valores e mais fáceis de serem traduzidos em sugestões de ações em situações práticas. O XP possui vários princípios como: princípio de benefício mútuo, de diversidade, de melhoramento, de oportunidade, de redundância, entre outros [44].

As práticas do XP dizem respeito a um conjunto de aspectos presentes no dia-a-dia da equipe de projeto. Estas são divididas em duas categorias: primárias e corolárias. As primeiras trazem melhorias imediatas à execução do projeto; entretanto, as segundas só costumam trazer benefícios após as primeiras terem sido alcançadas.

Algumas das principais práticas do XP podem ser vistas na Tabela 22.

Tabela 22 – Práticas do XP

Prática	Descrição
Cliente Presente	O cliente deve guiar o desenvolvimento com o seu <i>feedback</i> .
Stand Up Meeting	As reuniões de projeto devem ser rápidas.
Refactoring	O código deve ser revisto para ser suficientemente simples.
Código Coletivo	Todos os desenvolvedores podem modificar qualquer parte do código.
Ritmo Sustentável	Desenvolvedores não devem fazer horas-extras.
Integração Contínua	Assegurar que todo o sistema esteja sempre funcionando.
Releases Curtos	Gerar um fluxo contínuo de valor para o cliente.

Veja na Figura 73 uma imagem, retirada da referência [44], ilustrando as principais práticas primárias e corolárias do XP.

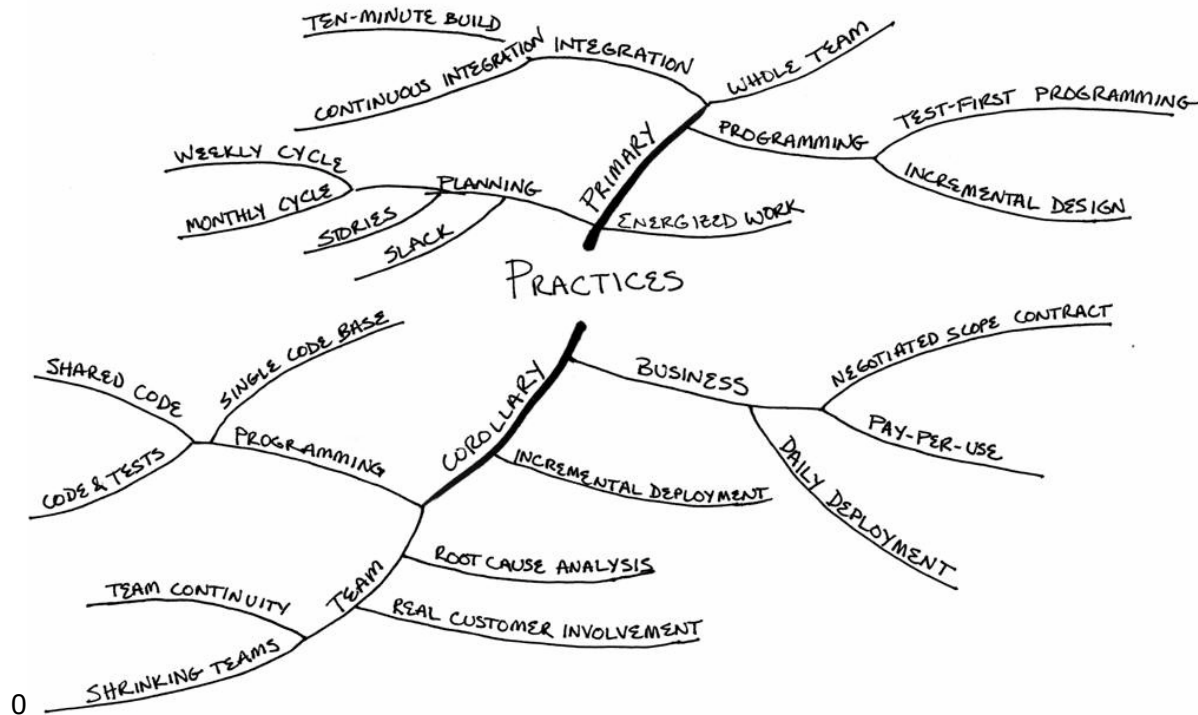


Figura 73 – Práticas Primárias e Corolárias do XP

Portanto, pode-se dizer que o XP é a junção harmoniosa dos valores, dos princípios e das práticas apresentadas no decorrer desta subseção.

4.2.2 Extreme Game Development (XGD)

Extreme Game Development (XGD) é uma metodologia ágil de desenvolvimento de jogos inspirada no *Extreme Programming* (XP). Assim como o XP, o XGD foca em criar uma versão executável do jogo o mais cedo possível [11].

A principal motivação que levou ao surgimento do XGD foram os constantes atrasos presentes no desenvolvimento de jogos em conjunto com as altas penalidades impostas pelas publicadoras sobre os atrasos [45]. Isto faz com que a equipe de desenvolvimento trabalhe sobre grande pressão e aumenta as chances da entrega de *milestones* instáveis.

Como o XGD é baseado na essência do XP, ele possui práticas semelhantes. Por exemplo: o XGD estimula a presença da publicadora (cliente) como parte integrante da equipe de desenvolvimento, iterações rápidas para que os problemas possam ser identificados rapidamente e integração contínua.

Logo, o acompanhamento mais perto da publicadora permite identificar antecipadamente problemas e indicar alternativas; o que leva, por sua vez, a uma diminuição das chances do projeto atrasar.

No entanto, ao contrário do desenvolvimento de *software*, a criação de um jogo envolve atividades que não são de programação. Como já foi visto anteriormente, há atividades de produção de conteúdo de arte e de áudio, por exemplo. Portanto, é preciso adaptar os elementos do XP para lidar com estes aspectos.

As principais críticas ao XGD dizem respeito a um suporte ineficiente à documentação necessária ao desenvolvimento de jogos. Ou seja, o modelo em questão não produz naturalmente a documentação necessária, no nível de detalhes comumente exigido.

4.3 Metodologias Preditivas

Como já foi dito anteriormente, as metodologias preditivas possuem fases de desenvolvimento bem definidas e delimitadas e, diferentemente das metodologias ágeis, também acreditam ser importante haver uma maior formalidade quanto à documentação.

As próximas subseções irão mostrar as principais características do GWP (*Game Waterfall Process*), uma adaptação do modelo cascata para o desenvolvimento de jogos, do RUP e do GUP (*Game Unified Process*), uma adaptação do RUP para o mercado de jogos.

4.3.1 Game Waterfall Process (GWP)

O *Game Waterfall Process* (GWP) é uma adaptação do modelo cascata para o desenvolvimento de jogos. Apesar das críticas, este modelo ainda é bastante utilizado [12].

Similar ao modelo cascata, o GWP é composto por um conjunto bem definido de fases que precisam ser concluídas em uma determinada ordem. Depois de concluída uma fase, a equipe de desenvolvimento não volta a trabalhar nesta.

O GWP pode acontecer desde os mais projetos simples até os mais complexos. Dependendo do caso, mais ou menos fases o modelo pode ter. Contudo, é comum haver um conjunto básico de 10 fases (Figura 74).

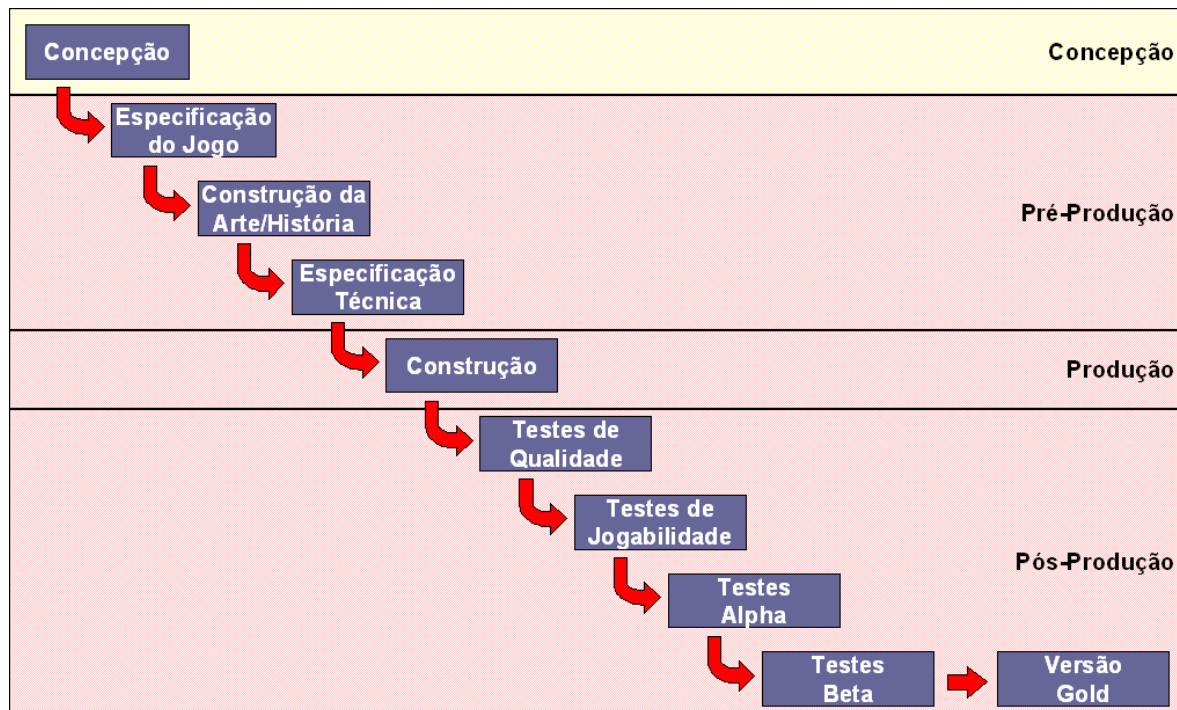


Figura 74 – Fases do Game Waterfall Process

Observe que as 10 fases do GWP se enquadram na perspectiva, apresentada anteriormente, que vê o desenvolvimento de jogos como um conjunto de atividades separadas em grandes etapas: Concepção, Pré-Produção, Produção e Pós-Produção.

Vea na Tabela 23 uma breve descrição de cada uma das 10 fases do GWP.

Tabela 23 – Fases do GWP

Fase	Descrição
Concepção	O produtor, alta gerência do desenvolvedor, assim como a publicadora discutem os principais aspectos do jogo. Entre outros pontos, eles discutem: o público-alvo, plataforma do jogo, tempo de desenvolvimento estimado e principais elementos do jogo.
Especificação do Jogo	Neste momento inicia-se a especificação do jogo: descreve-se como será a interação do jogador com o jogo e as características do fluxo do jogo.
Construção da Arte/História	Em seguida detalham-se os elementos de arte e de história presentes no jogo.
Especificação Técnica	Após detalhada a arte e a história, descrevem-se em detalhes os elementos técnicos relacionados com a codificação dos componentes do jogo.
Construção	Inicia-se o desenvolvimento do jogo, guiado pela documentação produzida até então.
Testes de Qualidade	Nesta fase verifica-se e garante-se a conformidade do jogo perante a documentação produzida antes do início da construção do jogo.
Testes de Jogabilidade	Aqui se investiga e balanceiam-se os elementos de jogabilidade do jogo com o objetivo de satisfazer as necessidades de entretenimento do público-alvo.
Testes Alpha	Os testes <i>alpha</i> consistem em testar o jogo com um público não restrito à equipe de desenvolvimento. Costuma-se submeter o jogo a testes por parte da publicadora.
Testes Beta	Os testes <i>beta</i> envolvem um público ainda mais aberto. Neste momento costuma-se testar o jogo com um grupo restrito de consumidores finais.
Versão Gold	Após todas as fases anteriores, o jogo é então lançado para o público geral.

Apesar de bem estruturado, este modelo é passível das mesmas críticas aplicadas ao modelo cascata. Por exemplo: como os testes são realizados somente após a etapa de construção do jogo, caso alguma grande não conformidade seja detectada, as necessidades de mudanças podem ter consequências drásticas no escopo, cronograma e custos do projeto.

As mudanças de grande impacto podem fazer com que o jogo seja enviado retroativamente para uma fase que já fora supostamente concluída. Neste cenário, o desenvolvimento se torna caótico uma vez que o princípio de desenvolvimento cascata foi quebrado.

Na prática, o que se tem visto é que o modelo cascata, apesar de sua estrutura sequencial bem definida, costuma transformar problemas, pequenos se antecipados, em bolas de neve.

4.3.2 Rational Unified Process (RUP)

O *Rational Unified Process*, criado pela *Rational Software Corporation*, agora uma divisão da *IBM*, é um processo de engenharia de *software* que provê uma metodologia disciplinada para realizar o trabalho de desenvolvimento de *software*. Seu principal objetivo é garantir a produção de *software* com alto grau de qualidade e que

atenda as necessidades dos seus usuários finais dentro de um cronograma e orçamento previsto [46].

O RUP procura considerar as melhores práticas do desenvolvimento atual de *software* presentes e aplicáveis em uma grande gama de projetos de desenvolvimento e organizações. Logo, pode-se dizer que o RUP é baseado em um conjunto de melhores práticas do desenvolvimento de *software*. Veja quais são estes na Tabela 24 e 25.

Tabela 24 – Melhores Práticas do Desenvolvimento de *Software*

Princípios	Descrição
Desenvolvimento Iterativo e Incremental	É comum a mudança nos requisitos ao longo do desenvolvimento do <i>software</i> , quer seja por restrições da arquitetura, necessidades do cliente ou por mau entendimento do problema original.
	O desenvolvimento iterativo permite que o projeto seja desenvolvido a partir de uma abordagem de refinamento sucessivo. O que diminui os riscos do projeto, permite um maior <i>feedback</i> por parte do cliente e mantém os desenvolvedores focados.
	O RUP utiliza o desenvolvimento iterativo incremental para principalmente: analisar e implementar partes do sistema progressivamente, facilitar a integração de componentes, permitir reuso de componentes e acomodar mudanças nos requisitos.
Gerenciamento dos Requisitos	O gerenciamento de requisitos no RUP se preocupa em atender as necessidades do cliente identificando e especificando o que ele quer, como também identificando quando estas necessidades mudam.
Arquitetura Baseada em Componentes	Desenvolver <i>softwares</i> cuja arquitetura é baseada em componentes faz com que eles sejam mais facilmente extensíveis, mais fáceis de serem compreendidos e promove reuso de código.
	O RUP foca na produção de uma arquitetura básica já no início do desenvolvimento. Esta arquitetura inicial torna-se um protótipo que guia o início do ciclo de desenvolvimento propriamente dito.
Software Modelado Visualmente	Criar abstrações visuais dos componentes do <i>software</i> ajuda a construir uma visão geral mais clara da solução; permitindo, assim, uma melhor codificação.
	Modelar visualmente um <i>software</i> facilita, inclusive, a identificação de componentes passíveis de reuso.
	O RUP recomenda a utilização da <i>Unified Modeling Language</i> (UML) para modelar visualmente aspectos do <i>software</i> . Por exemplo: casos de uso, diagramas de classes, diagramas de atividades, etc.
Qualidade do Software	A busca pela qualidade do <i>software</i> é uma atividade muitas vezes postergada para o final do projeto e realizada por uma equipe não necessariamente ligada ao desenvolvimento.
	O RUP procura realizar controle de qualidade durante todo o processo de desenvolvimento e envolvendo todos os membros do time.

Tabela 25 – Melhores Práticas do Desenvolvimento de Software

Princípios	Descrição
Controle de Mudanças	Mudanças é algo inevitável em um projeto de <i>software</i> .
	O RUP define métodos para controlar e monitorar mudanças.

Além das seis práticas de desenvolvimento de *software* descritas anteriormente, há três importantes características presentes na essência do RUP. Veja na Tabela 26 os detalhes destas.

Tabela 26 – Características do RUP

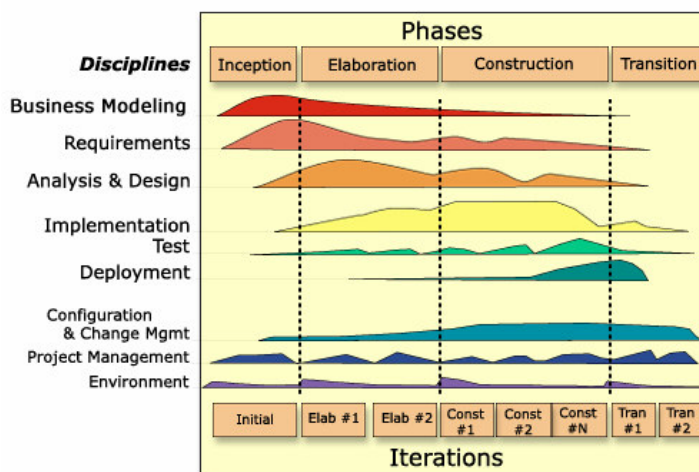
Característica	Descrição
Guiado por Casos de Uso	O RUP acredita que para entender como um sistema deve-se se comportar é preciso especificá-lo utilizando casos de uso.
	Diz-se, portanto, que o RUP é guiado por casos de uso, pois os casos de uso definidos para o sistema guiam todo o resto do processo de desenvolvimento.
Configurável	O RUP é um processo geral e compreensivo o suficiente para poder ser utilizado por pequenas e médias organizações.
	Porém, ele também pode ser adaptado para melhor atender as necessidades específicas de organizações de qualquer tamanho.
Suporte a Ferramentas	O RUP, com o objetivo de ser um processo efetivo, possui um conjunto grande de ferramentas para apoiar e automatizar várias atividades.

A junção das práticas acima guia a definição da arquitetura do RUP. Esta arquitetura possui duas dimensões (estruturas).

A primeira dimensão (eixo horizontal) representa o tempo e mostra os aspectos relacionados ao ciclo de desenvolvimento. Esta dimensão costuma ser vista como a estrutura dinâmica do RUP.

A segunda dimensão (eixo vertical) representa os fluxos de atividades, realizados no desenvolvimento de *software*, reunidos em grupos de atividades logicamente relacionadas. Esta dimensão costuma ser vista como a estrutura estática do RUP.

Veja na Figura 75 uma visão geral da arquitetura do RUP.

**Figura 75 – Arquitetura do Rational Unified Process**

A dimensão horizontal é composta por fases, iterações e *milestones*. O RUP possui quatro fases: Concepção, Elaboração, Construção e Transição. Estas, por sua vez, são divididas em uma ou mais iterações. O final de cada fase é marcado pela concretização de um *milestone*. Veja na Tabela 27 uma descrição de cada uma das fases do RUP.

Tabela 27 – Fases do RUP

Fase	Descrição
Concepção	Nesta fase devem-se estabelecer os elementos pertinentes ao escopo do projeto. Também são preocupações desta fase as previsões do projeto em termos de finanças e de cronograma.
	O <i>milestone</i> desta fase é o <i>Life-Cycle Objective</i> : o escopo, custo e tempo do projeto bem definidos.
Elaboração	Nesta fase deve-se analisar o domínio do problema; ou seja, identificar os requisitos do sistema, assim como chegar a uma arquitetura estável.
	O <i>milestone</i> desta fase é o <i>Life-Cycle Architecture</i> : o escopo e objetivos do sistema detalhados, como também uma arquitetura estável.
Construção	Nesta fase deve-se focar no desenvolvimento dos componentes do sistema. É nesta fase que a atividade de codificação tem maior atuação.
	O <i>milestone</i> desta fase é o <i>Initial Operational Capability</i> : a primeira versão externa executável do <i>software</i> .
Transição	Nesta última fase deve-se implantar o <i>software</i> no ambiente do usuário final.
	O <i>milestone</i> desta fase é o <i>Product Release Milestone</i> : o <i>software</i> executável atendendo às necessidades do cliente.

Dentro de cada iteração de uma fase, costuma-se realizar um conjunto de atividades. O RUP agrupa estas atividades em 6 disciplinas de engenharia (Modelagem do Negócio, Requisitos, Análise e Projeto, Implementação, Teste e Implantação) e 3 disciplinas de apoio (Gerência de Mudanças e Configuração, Gerência de Projetos e Ambiente). Veja na Tabela 28 e 29 uma breve descrição das disciplinas do RUP.

Tabela 28 – Disciplinas do RUP

Disciplina	Descrição
Modelagem do Negócio	O objetivo principal desta disciplina é estabelecer um bom nível de entendimento da estrutura, dos problemas e das possíveis melhorias da organização (cliente).
Requisitos	O objetivo principal desta disciplina é descrever com detalhes o que o sistema a ser desenvolvido deve fazer. Deve-se chegar a uma visão comum entre todas as partes envolvidas com o projeto.
Análise e Projeto	O objetivo principal desta disciplina é descrever com detalhes como o sistema será realizado no momento de sua implementação.

Tabela 29 – Disciplinas do RUP

Disciplina	Descrição
Implementação	O objetivo principal desta disciplina é organizar o código do sistema em termos da implementação de subsistemas organizados em camadas.
	Também faz parte desta disciplina a implementação propriamente dita dos componentes, os testes unitários para cada um dos componentes implementados e a integrações entre estes.
Teste	Os objetivos desta disciplina são verificar a interação entre os componentes do sistema, a correta integração entre as partes, a conformidade do código implementado com os requisitos estabelecidos e identificar problemas que devem ser corrigidos antes da implantação do <i>software</i> .
Implantação	O objetivo principal desta disciplina é produzir versões estáveis do produto e entregá-lo para os seus usuários finais.
	Também se preocupa com a implantação do produto no ambiente do cliente, como também com o treinamento dos usuários para que estes possam usar corretamente o sistema.
Gerência de Mudanças e Configuração	O objetivo principal desta disciplina é realizar a gerência da configuração do repositório no qual o projeto encontra-se.
	Esta disciplina também se preocupa com o monitoramento das de mudanças solicitadas e ocorridas.
Gerência de Projeto	Os objetivos desta disciplina são planejar e gerenciar o ciclo de desenvolvimento.
	A atividade de planejamento pode ser dividida em duas categorias: planejamento das fases, ocorre uma vez no projeto, e planejamento das iterações, ocorre uma vez em cada iteração do projeto.
Ambiente	O objetivo desta disciplina é apoiar o desenvolvimento do <i>software</i> com ferramentas e processos.

Cada disciplina é definida em termos de papéis (quem), artefatos (o quê), atividades (como) e fluxo de atividades (quando).

O papel define as competências e responsabilidades de quem vai participar da disciplina. Os artefatos representam o resultado proveniente da realização de alguma atividade. As atividades associam um dado trabalho a um papel que provê um resultado (artefato) bem definido.

O fluxo de atividades, por sua vez, define uma seqüência lógica entre atividades e a interação entre atores para produzir algum artefato de grande valor para o processo de desenvolvimento.

Por fim é importante entender que apesar do RUP ser aconselhado a projetos com times grandes de desenvolvimento, maior demanda de documentação e estruturação, ele também pode ser aplicado em outros contextos. Na verdade, o RUP é um modelo que busca uma maior predictividade do desenvolvimento, mas que pode e deve ser adaptado à realidade de cada projeto.

4.3.3 Game Unified Process (GUP)

O *Game Unified Process* (GUP) é um modelo concebido por Kevin Flood a partir das necessidades detectadas na execução de um projeto no qual trabalhava como produtor [12].

A empresa executora do projeto em questão utilizava um modelo inspirado nos moldes do GWP. Contudo, as dificuldades inerentes a este modelo tomaram grandes proporções e criaram a necessidade de adoção de um novo modelo.

O GUP é, portanto, um modelo inspirado principalmente no RUP; no entanto, com algumas características presentes no XGD. Apesar do aspecto promissor do modelo, não há relato sobre seus elementos e uso nos últimos anos.

4.4 Análise Comparativa

Nesta última seção, será realizada uma análise comparativa entre a metodologia ágil e preditiva mais representativa, XP e RUP respectivamente, com as características do ciclo de desenvolvimento de jogos (Tabela 30 e 31). As características que serão consideradas aqui são as listadas e discutidas no final do capítulo anterior.

Com esta análise pretende-se identificar que corrente e metodologia de desenvolvimento de *software* melhor atende às características presentes no desenvolvimento de jogos para PC.

Tabela 30 – Comparativo RUP vs. XP no Desenvolvimento de Jogos para PC

Característica	RUP	XP
Tamanho da Equipe	Satisfaz	Satisfaz Menos
	A estrutura do RUP lida bem com grandes equipes de desenvolvimentos.	Apesar de relatos de casos de sucesso, o XP é mais recomendado para projetos com pequenos times de desenvolvimento.
Composição da Equipe	Não Satisfaz	Não Satisfaz
	O RUP não considera, no desenvolvimento do <i>software</i> , o papel de artistas.	O XP não considera, no desenvolvimento do <i>software</i> , o papel de artistas.
Tempo de Desenvolvimento	Satisfaz	Satisfaz Menos
	No caso de desenvolvimento de jogos para PC, normalmente com ciclos de desenvolvimento mais longos, o RUP é mais indicado.	No caso de desenvolvimento de jogos para PC, normalmente com ciclos de desenvolvimento mais longos, o XP é menos indicado.
Caráter Iterativo e Incremental	Satisfaz	Satisfaz Menos
	O RUP é por definição iterativo e incremental e possui um modelo de planejamento, controle e execução de iterações explícitos.	Apesar de também ser iterativo e incremental, o XP não possui de forma explícita um modelo de planejamento, controle e execução das iterações.
Ocorrência de Mudanças	Satisfaz	Satisfaz
	O gerenciamento das mudanças faz parte da essência do RUP.	O gerenciamento das mudanças também faz parte da essência do XP.

Tabela 31 – Comparativo RUP vs. XP no Desenvolvimento de Jogos para PC

Característica	RUP	XP
Proporção dos Custos	Satisfaz	Satisfaz Menor
	O RUP possui uma disciplina preocupada especificamente com a gerência do projeto, o que envolve a gerência dos custos deste.	O XP não possui uma preocupação específica com a gerência dos custos do projeto.
Necessidade de Documentação	Satisfaz	Satisfaz Menos
	O RUP possui uma preocupação clara com a documentação do processo de desenvolvimento.	O XP possui uma preocupação menor em relação à documentação do processo de desenvolvimento.
Influência do <i>Game Design Document</i>	Não Satisfaz	Não Satisfaz
	O GDD não é um elemento presente no RUP.	O GDD não é um elemento presente no XP.
Necessidade de Integração	Satisfaz	Satisfaz
	O caráter iterativo e incremental do RUP gera a preocupação constante com integração.	O caráter iterativo e incremental do XP gera a preocupação constante com integração.
Necessidade de Reuso e Repetição	Satisfaz	Satisfaz Menos
	A preocupação do RUP em modelar visualmente a arquitetura permite identificar melhor pontos de reuso.	Apesar do XP se preocupar com práticas como refatoração, a preocupação menor em modelar visualmente a arquitetura diminui a capacidade de identificar pontos de reuso.

A partir da análise realizada nas tabelas anteriores, pode-se concluir que o RUP é uma metodologia de desenvolvimento de *software* que melhor satisfaz as características da produção de jogos para PC. Portanto, modelos mais preditivos são mais aconselhados para o cenário de criação de jogos para PC.

Contudo, é importante destacar que apesar de satisfazer parte considerável das características, este é um modelo de desenvolvimento que não possui preocupação com aspectos exclusivos da criação de jogos. Por exemplo: o RUP não considera a existência de papéis como artistas, engenheiros de áudio e *game designers*; logo, o RUP também não inclui atividades e artefatos relacionados com estes papéis.

Portanto, para melhor atender às características e necessidades do desenvolvimento de jogos para plataforma PC é preciso um modelo preditivo, inspirado na essência do RUP.

Um modelo preditivo, de autoria do autor deste trabalho, inspirado na essência do RUP e voltado para o desenvolvimento de jogos de computador, será apresentado e analisado no próximo capítulo.

5. Modelo Preditivo de Desenvolvimento de Jogos *PC*

Este capítulo tem como objetivo apresentar o modelo preditivo de desenvolvimento de jogos proposto. Inspirado na essência do RUP, o modelo foi criado com o objetivo de atender às características específicas da criação de jogos para *PC*. As quatro seções contidas neste capítulo abordarão os seguintes temas:

Seção 1: Introdução

Apresentam-se as práticas que norteiam o modelo, assim como as características-chaves deste.

Seção 2: Visão Geral

Descreve-se o modelo sob uma perspectiva macro: qual o papel e importância da estrutura dinâmica e estática do modelo.

Seção 3: Estrutura Dinâmica - Fases

Detalham-se as fases que compõem a estrutura dinâmica do modelo.

Seção 4: Estrutura Estática - Disciplinas

Detalham-se as disciplinas que compõem a estrutura estática do modelo.

5.1 Introdução

O **Modelo** preditivo de **Desenvolvimento de Jogos** para plataforma **PC** criado será, a partir deste ponto do documento, denominado simplesmente **MDJ-PC**.

O MDJ-PC é um mapeamento estruturado do processo de desenvolvimento de jogos para plataforma *PC*. É um modelo genérico que busca organizar as principais etapas e atividades presentes no ciclo de criação de jogos para a plataforma em questão.

O objetivo principal do MDJ-PC é fornecer um modelo estruturado que permita pessoas e empresas melhor planejar, gerenciar e executar a criação de jogos. Desta forma, cada pessoa e empresa poderá melhor estruturar o seu processo de desenvolvimento. Podendo, então, realizar economias de custos, sem perda de qualidade.

Ou seja, o MDJ-PC pode e deve ser utilizado como um instrumento para reduzir os altos e crescentes custos do desenvolvimento de jogos; uma vez que, processos melhor estruturados tendem a minimizar gastos desnecessários.

A essência do MDJ-PC é bem similar à essência do RUP. Nas próximas subseções, serão apresentadas e descritas as práticas e características deste novo modelo.

5.1.1 Práticas

As principais práticas do desenvolvimento de *software* apresentados no capítulo anterior, presentes no conceito do RUP, também fazem parte, em um contexto diferente, do MDJ-PC.

Estas práticas são: desenvolvimento iterativo e incremental, arquitetura baseada em componentes, *software* modelado visualmente, qualidade do *software* e controle de mudanças. Além destes, outras três práticas fazem parte do MDJ-PC: gerenciamento do *game design*, coerência áudio-visual e entretenimento apropriado. Veja na Tabela 32 a descrição de como estas práticas se aplicam ao desenvolvimento de jogos.

Tabela 32 – Práticas do MDJ-PC

Princípios	Descrição
Desenvolvimento Iterativo e Incremental	O desenvolvimento de um jogo para <i>PC</i> de forma iterativa e incremental permite diminuir os riscos do projeto a partir de uma abordagem de refinamento sucessivo.
	Desenvolve-se o jogo progressivamente analisando e implementando partes do seu conceito com o objetivo de facilitar a integração de componentes e acomodar mudanças no <i>game design</i> .
Arquitetura Baseada em Componentes	Deve-se pensar no jogo com um conjunto de componentes atuando de forma cooperativa com o objetivo de entreter o jogador.
	Jogos desenvolvidos em cima de componentes fazem com que eles sejam mais facilmente extensíveis, mais fáceis de serem compreendidos e promove reuso de código.
	Esta arquitetura deve começar a ser delineada desde os primeiros protótipos desenvolvidos.
Software Modelado Visualmente	O jogo deve ser modelado visualmente.
	Não só o código, mas o fluxo do jogo, os conteúdos de arte e de áudio devem ser modelados visualmente.
	Para tanto se devem utilizar as técnicas mais apropriadas. Por exemplo: UML (para o código), <i>storyboards</i> (para o fluxo do jogo) e arte conceitual (para o conteúdo de arte).
Qualidade do Software	Deve haver uma preocupação constante com a garantia da qualidade do jogo como um todo (código, arte e áudio) em relação aos padrões estabelecidos.
Controle de Mudanças	A mudança é inevitável no processo de criação de jogos.
	Devem-se utilizar métodos objetivos e efetivos para controlar e monitorar as mudanças de todas as áreas presentes na criação de jogos.
Gerenciamento do Game Design	Similar ao gerenciamento dos requisitos, o gerenciamento do <i>game design</i> se preocupa em atender às necessidades de entretenimento do jogador identificando e especificando as características do jogo.
Coerência Áudio-Visual	O jogo deve ter coerência áudio-visual com o seu estilo.
	Por exemplo: em um jogo infantil, as músicas e o visual devem ser desenvolvidos pensando neste público-alvo.
Entretenimento Apropriado	O jogo deve passar o sentimento de entretenimento coerente com o seu estilo e público-alvo.
	Crianças e adultos se entretêm de formas diferentes. Logo, jogos para crianças e adultos devem ter experiências de entretenimento distintas.

5.1.2 Características

O MDJ-PC possui, além das práticas apresentadas na subseção anterior, duas importantes características presentes na sua essência. Estas são: guiado pelo *game design* e configurável. Veja na Tabela 33 os detalhes destas características.

Tabela 33 – Características do MDJ-PC

Característica	Descrição
Guiado pelo <i>Game Design</i>	Para entender bem como o jogo deve ser, é preciso tê-lo especificado, com um grau de detalhamento necessário, através de um <i>Game Design</i> .
	Diz-se, portanto, que todas as atividades envolvidas com a criação de um jogo devem ser, direta ou indiretamente, guiadas pelo seu <i>Game Design</i> .
Configurável	Assim como o RUP, o MDJ-PC é um modelo geral o suficiente para ser utilizado em contextos e realidades diferentes de pequenas, médias e grandes organizações.
	O MDJ-PC deve, portanto, ser adaptado e instanciado para as necessidades específicas de cada projeto e organização.

Idealmente, o MDJ-PC, assim como o RUP, também deveria ter como característica o suporte a ferramentas. Ou seja, o modelo em questão deveria oferecer um conjunto de ferramentas que auxiliassem e automatizassem atividades presentes no ciclo de produção de jogos.

Contudo, no estado atual, o MDJ-PC é constituído unicamente por um modelo teórico. Não possuindo, portanto, ferramentas que apóiem a instanciação deste modelo para uma realidade específica.

5.2 Visão Geral

Assim como o RUP, o MDJ-PC é composto por duas dimensões: estática (vertical) e dinâmica (horizontal).

A estrutura estática consiste em um conjunto de disciplinas. Ou seja, podemos ver o MDJ-PC como um grupo de disciplinas realizadas ao longo do tempo. Cada uma é definida em termos de quem (papéis), como (atividade), o quê (artefato) e quando (fluxo de atividades). As definições de papéis, atividades, artefatos e fluxos de atividades do MDJ-PC são similares às do RUP.

A estrutura dinâmica consiste em estruturar como o conjunto de disciplinas da estrutura estática se comporta ao longo da linha do tempo. Associados à estrutura dinâmica, têm-se os conceitos de fases, iterações e *milestones*.

A estrutura dinâmica do MDJ-PC, similar ao RUP, procura resolver os principais problemas do modelo cascata. O modelo cascata, como visto anteriormente, segue uma abordagem seqüencial para desenvolver *software*.

Pode-se dizer que os principais problemas do modelo cascata são dois. O primeiro é assumir que os requisitos, após coletados, não sofrerão mudanças e, portanto, pode-se analisar e implementar o sistema sem precisar rever os requisitos. O segundo é assumir que é possível analisar o *software* e saber perfeitamente o que vai ser preciso implementar, sem que esta visão mude ao longo do processo de desenvolvimento. Na prática, ocorre exatamente o oposto.

Com o objetivo de superar estas dificuldades surge a idéia de desenvolvimento iterativo e incremental. De forma breve, o conceito é quebrar o desenvolvimento de um projeto grande em pequenas etapas (iteraões) que produza incrementalmente o produto final.

Desta forma, em cada iteração tem-se a execução seqüencial, como o cascata, de um conjunto de atividades. Contudo, cada conjunto de atividades é executado novamente, em escopo e profundidade potencialmente diferente, na próxima iteração. O que guia a definição do escopo e profundidade das atividades em cada iteração é uma priorização baseada na redução dos riscos de fracassos do projeto.

No contexto do MDJ-PC, isto significa que em cada iteração o processo de desenvolvimento passa por todas as disciplinas do modelo; no entanto, em cada iteração as atividades realizadas diferem em termos de escopo e profundidade.

Entretanto, o conceito de iteração por se só não é suficiente, pois se pode terminar iterando excessivamente e o final do projeto nunca ser atingido. Para tanto, existe o conceito de fase e *milestones*.

Uma fase reúne uma ou mais iterações que possuem como objetivo maior atingir um marco (*milestone*). Ou seja, as fases e *milestones* são artifícios criados com o objetivo de oferecer um maior controle sobre o rumo do processo iterativo e incremental. Veja na Figura 76 um diagrama ilustrando o conceito de disciplinas, iterações, fases e *milestones*.

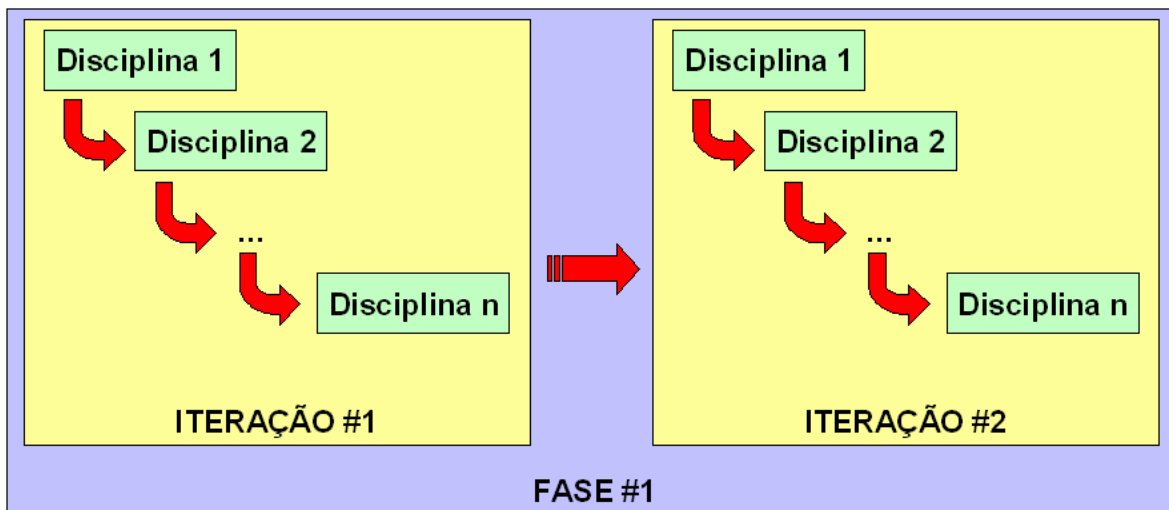


Figura 76 – Conceito de Disciplinas, Iterações, Fases e *Milestones*

O MDJ-PC possui um conjunto de quatro fases (Concepção, Pré-Produção, Produção e Pós-Produção) e 12 disciplinas, estas divididas em 9 disciplinas de criação (Modelagem do Mercado, *Game Design*, *Technical Design*, Implementação, Criação de Arte 2D, Criação de Arte 3D, Criação de Áudio, Teste e Distribuição) e 3 disciplinas de apoio (Gerência de Configuração e Mudanças, Gerência de Projeto e Ambiente). Veja na Figura 77 como estas disciplinas se comportam em relação às fases da estrutura dinâmica do MDJ-PC.

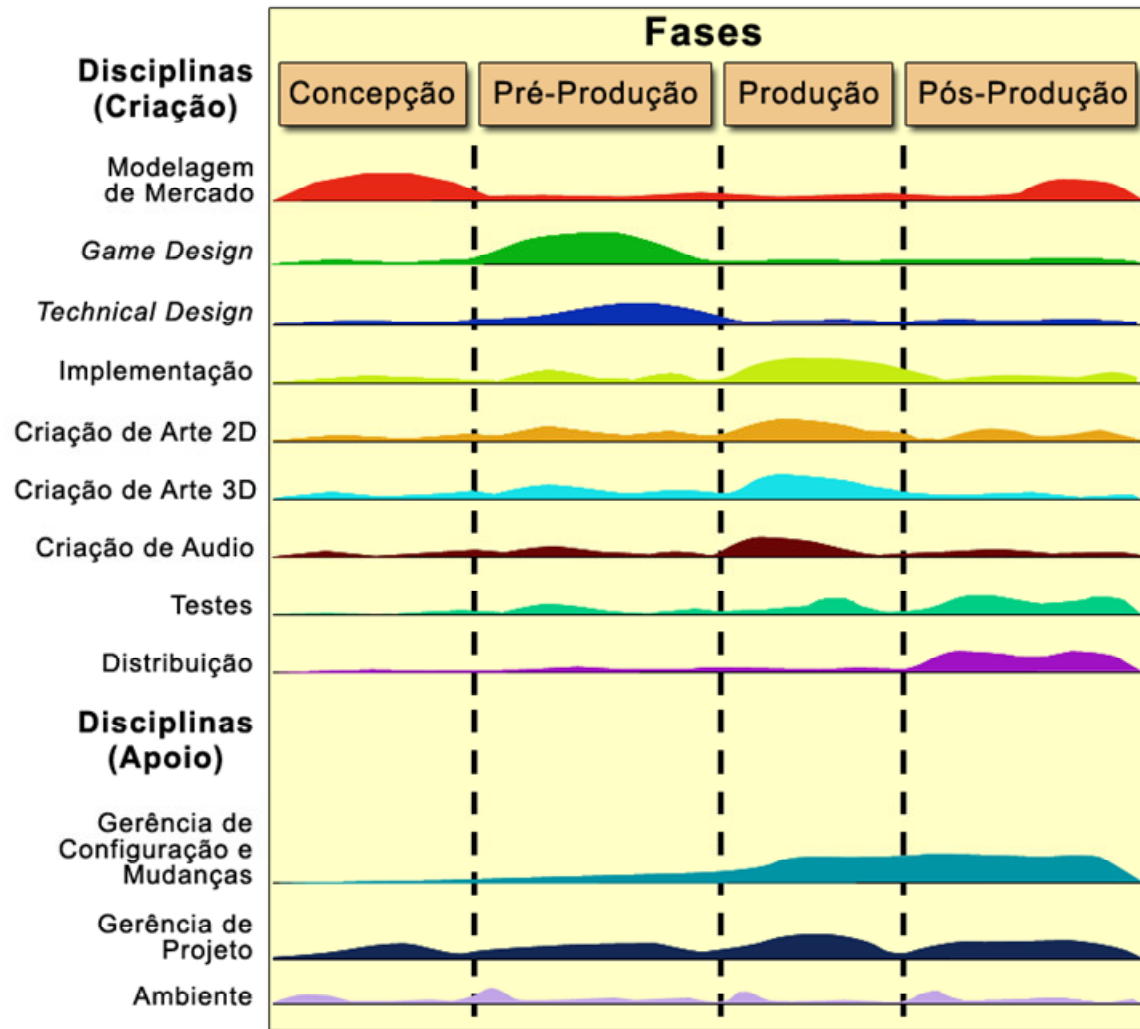


Figura 77 – Estrutura do MDJ-PC

A disciplina Modelagem de Mercado demanda um maior esforço na primeira fase do MDJ-PC (Concepção). É neste momento que se precisam identificar as necessidades em termos de entretenimento e jogabilidade do mercado (público-alvo). Nas fases de Pré-Produção e Produção mantém certa estabilidade. É responsável por monitorar, em intensidade menor, o mercado. Na última fase (Pós-Produção) há um maior esforço para lançar o produto explorando os pontos mais alinhados com o mercado de jogos no momento.

A disciplina *Game Design*, apesar de ter atividades em todas as fases, demanda um maior esforço na fase de Pré-Produção em que se definem com certa profundidade os elementos do jogo.

A disciplina *Technical Design*, similar à disciplina *Game Design*, demanda um maior esforço na fase de Pré-Produção com o objetivo de especificar-se tecnicamente o jogo. A intensidade desta disciplina é um pouco deslocada para a direita na linha do tempo em relação ao *Game Design*. O motivo é que as principais atividades desta disciplina dependem do resultado de atividades da disciplina de *Game Design*.

A disciplina Implementação possui um maior esforço na fase de Produção. Contudo, faz parte desta disciplina o desenvolvimento de protótipos, nas fases de

Concepção e Pré-Produção, como também a implementação de eventuais correções e melhorias na fase de Pós-Produção.

A disciplina Criação de Arte 2D possui também uma maior intensidade na fase de Produção. No entanto, também é responsável pela criação de arte conceitual e protótipos da interface gráfica nas fases de Concepção e Pré-Produção. Na fase de Pós-Produção realiza, principalmente, eventuais modificações na interface gráfica do jogo.

A disciplina Criação de Arte 3D possui uma maior intensidade na fase de Produção. No entanto, é responsável pela criação de protótipos de modelos tridimensionais nas fases de Concepção e Pré-Produção. Na fase de Pós-Produção desempenha atividades de melhorias em pequena escala, uma vez que o conteúdo gráfico tridimensional não costuma sofrer grandes alterações neste momento.

A disciplina Criação de Áudio segue a mesma idéia da disciplina Criação de Arte 3D. Grande esforço na Produção, esforço menor na Concepção e Pré-Produção e pequeno esforço na fase de Pós-Produção.

A disciplina Teste, apesar de algum esforço ao longo das fases de Concepção, Pré-Produção e Produção, possui sua intensidade concentrada na fase de Pós-Produção com o objetivo de realizar testes *alpha* e *beta*.

A disciplina Distribuição, semelhante à disciplina Teste, possui seu maior esforço concentrado na fase de Pós-Produção com o objetivo de finalizar e lançar o jogo.

A disciplina Gerência de Configuração e Mudanças possui um pequeno esforço no início do projeto (Concepção) e cresce ao longo do tempo (Pré-Produção) atingindo o seu ápice perto do fim do projeto (Produção e Pós-Produção).

A disciplina Gerência de Projetos possui um comportamento relativamente cíclico. No início de cada fase o esforço é moderado e cresce no final da fase para garantir que o *milestone* seja atingido.

A disciplina Ambiente também possui esforço cíclico ao longo de todo o projeto. Normalmente maior no início da fase e menor no final desta.

5.2.1 Especificação

O MDJ-PC será especificado utilizando a notação *Software Process Engineering Metamodel* (SPEM) do *Object Management Group* (OMG). O SPEM é um metamodelo utilizado para descrever modelos de desenvolvimento de *software* [47]. Veja na Tabela 34 e 35 um resumo dos estereótipos deste metamodelo utilizados na especificação do MDJ-PC.

Tabela 34 – Estereótipos do SPEM








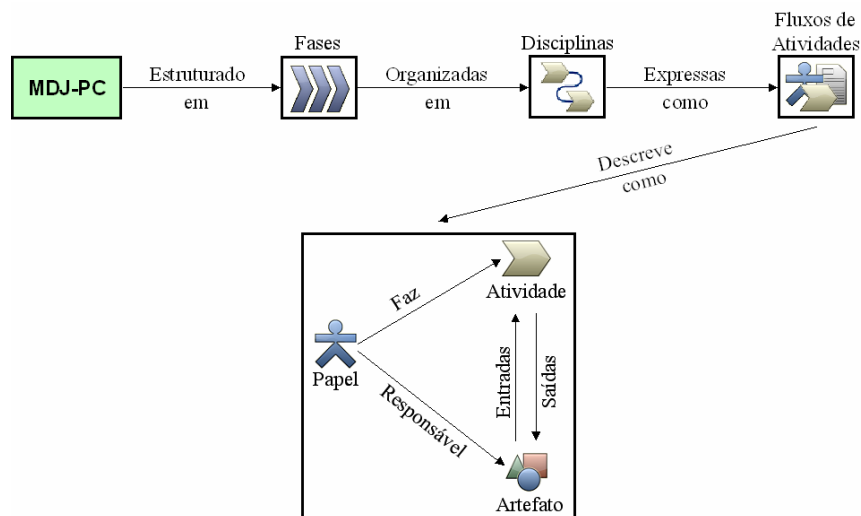
Estereótipo	Descrição
 <p>Fase</p>	A fase é um mecanismo de estruturar o conjunto de iterações do projeto com a intenção de atingir sequencialmente objetivos intermediários (<i>milestones</i>) que levem ao término do projeto.
 <p>Disciplina</p>	<p>A disciplina reúne um conjunto de atividades, papéis e artefatos logicamente relacionados com uma área de conhecimento do projeto.</p> <p>A disciplina é expressa por um fluxo de atividades.</p>

Tabela 35 – Estereótipos do SPEM

Esteriótipo	Descrição
 Fluxo de Atividades	O fluxo de atividades determina um relacionamento seqüencial lógico entre as atividades, papéis e artefatos envolvidos com uma dada disciplina.
 Papel	O papel define um conjunto de competências e responsabilidades que os atores, que irão realizar alguma atividade, precisam ter.
 Atividade	<p>A atividade é uma unidade de trabalho executado por um papel específico no projeto.</p> <p>Cada atividade possui um conjunto ordenado de passos, entradas e saídas.</p>
 Artefato	<p>O artefato é um produto do projeto.</p> <p>Artefatos são consumidos (entradas) e produzidos (saídas) pelas atividades do projeto.</p>
 Documento	<p>O documento é um tipo específico de artefato.</p> <p>O documento é um artefato composto por textos e imagens.</p>

Logo, utilizando a notação apresentada acima, a essência do MDJ-PC pode ser entendida como um conjunto de fases. Em cada fase, atividades relacionadas a um conjunto de disciplinas são realizadas. Cada disciplina é, por sua vez, descrita por um fluxo de atividades que engloba papéis, atividades, artefatos e documentos. Veja na Figura 78 um esquema gráfico desta idéia central.

**Figura 78 – Essência do MDJ-PC**

As próximas seções trarão mais detalhes sobre cada uma das fases e disciplinas do MDJ-PC.

5.3 Estrutura Dinâmica – Fases

O ciclo de desenvolvimento do MDJ-PC é composto por um conjunto de quatro grandes fases: Concepção, Pré-Produção, Produção e Pós-Produção. Veja na Figura 79 uma visão geral das fases do modelo e os respectivos *milestones*. As próximas subseções trarão mais detalhes sobre cada uma das fases.

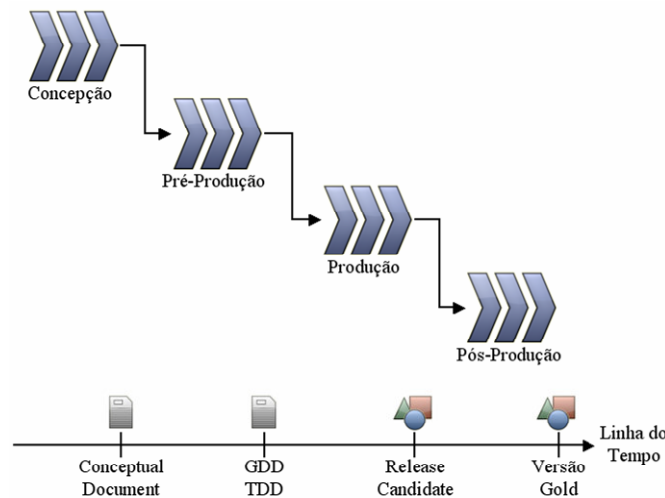



Figura 79 – Fases e *Milestones* do MDJ-PC

5.3.1 Concepção

A fase de Concepção é a primeira das quatro fases do MDJ-PC. Ela começa quando há uma intenção clara em desenvolver uma idéia de um jogo.

A partir de uma idéia, eventualmente descrita em um *Game Proposal Document*, uma ou mais entidades (desenvolvedores, publicadoras, etc.) tomam a decisão de evoluir esta idéia para um conceito melhor definido. Veja na Tabela 36 o objetivo principal e *milestone* desta fase.

Tabela 36 – Fase de Concepção


Fase de Concepção	
 Fase de Concepção	Objetivo
	O objetivo desta fase é desenvolver e comunicar uma visão unificada do conceito (elementos centrais) do jogo para todas as partes (desenvolvedores, publicadoras, etc.) engajadas com o desenvolvimento do mesmo.
	Milestone
	<i>Game Conceptual Document</i>

5.3.2 Pré-Produção

A fase de Pré-Produção é a segunda das quatro fases do MDJ-PC. Ela começa a partir do momento que se tem uma versão estável do *Game Conceptual Document*. Ou seja, as partes envolvidas definem os conceitos centrais do jogo e concordam que a

mesma é viável tecnicamente e comercialmente. Veja na Tabela 37 o objetivo principal e *milestones* desta fase.

Tabela 37 – Fase de Pré-Produção


Fase de Pré-Produção	
 Fase de Pré-Produção	Objetivo
	O objetivo desta fase é definir, reunir e especificar os elementos de arte, áudio e código do jogo necessários ao início efetivo da produção.
	Milestones
	<i>Game Design Document (GDD)</i> <i>Technical Design Document (TDD)</i>

5.3.3 Produção

A fase de Produção é a terceira das quatro fases do MDJ-PC. Ela começa a partir do momento que se tem uma versão estável e devidamente detalhada do *Game Design Document* e do *Technical Design Document*.

Quando os elementos de arte, áudio e código estão especificados com detalhes suficientes, inicia-se a produção efetiva do jogo. Veja na Tabela 38 o objetivo principal e *milestone* desta fase.


Tabela 38 – Fase de Produção

Fase de Produção	
 Fase de Produção	Objetivo
	O objetivo desta fase é produzir, testar e integrar os componentes de arte, áudio e código com o objetivo de criar a mecânica e fluxo do jogo.
	Milestone
	<i>Release Candidate</i>

5.3.4 Pós-Produção

A fase de Pós-Produção é a última das quatro fases do MDJ-PC. Ela começa quando se tem um *Release Candidate* que será submetido às atividades finais. É quando se concentram os esforços para finalizar o jogo. Veja na Tabela 39 o objetivo principal e *milestone* desta fase.

Tabela 39 – Fase de Pós-Produção

Fase de Pós-Produção	
 Fase de Pós-Produção	Objetivo
	O objetivo desta fase é produzir um pacote com o jogo finalizado e outros materiais (manuais, encartes, etc.) que serão entregues à publicadora para iniciar o processo de lançamento.
	Milestone
	<i>Gold Version</i>

5.4 Estrutura Estática – Disciplinas

As 12 disciplinas do MDJ-PC estão reunidas em dois grandes grupos: criação e apoio. As 9 disciplinas de criação estão associadas com as atividades desempenhadas com o objetivo principal de produzir os artefatos que comporão o jogo.

Já as 3 demais disciplinas de apoio estão associadas com as atividades realizadas com a função de prover apoio e suporte à criação do jogo. As próximas subseções trarão mais detalhes sobre cada uma das disciplinas do MDJ-PC.

5.4.1 Modelagem de Mercado

A disciplina Modelagem de Mercado engloba o conjunto de atividades comprometidas com a elaboração do conceito do jogo (elementos centrais) baseado nas necessidades de entretenimento do público-alvo do mercado em foco. Esta disciplina congrega 2 papéis, 6 atividades e 1 artefato. Veja na Figura 80 uma visão geral da disciplina.

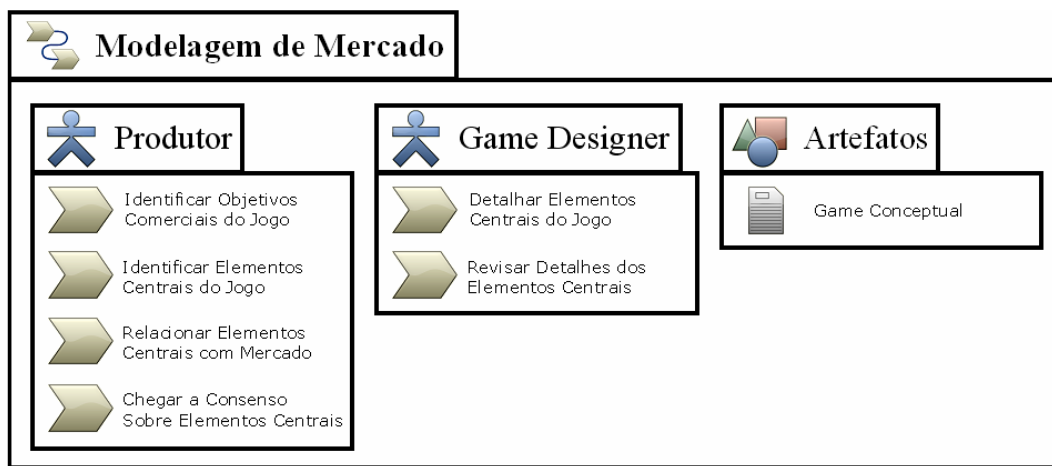


Figura 80 – Visão Geral da Disciplina Modelagem de Mercado

Os papéis presentes nesta disciplina são:

1. Produtor

Possui um papel parecido com o de um gerente de projetos; no entanto, combina também as responsabilidades e competências de estar atento às necessidades e tendências do mercado de jogos.

2. Game Designer

É um dos principais responsáveis pelo detalhamento dos elementos centrais do jogo. É um papel preocupado em fazer com que o jogo tenha os elementos certos dosados de forma coerente.

O principal artefato produzido nesta disciplina é:

1. *Game Conceptual Document*

Descreve uma visão comum do conceito (elementos centrais) do jogo.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 81.

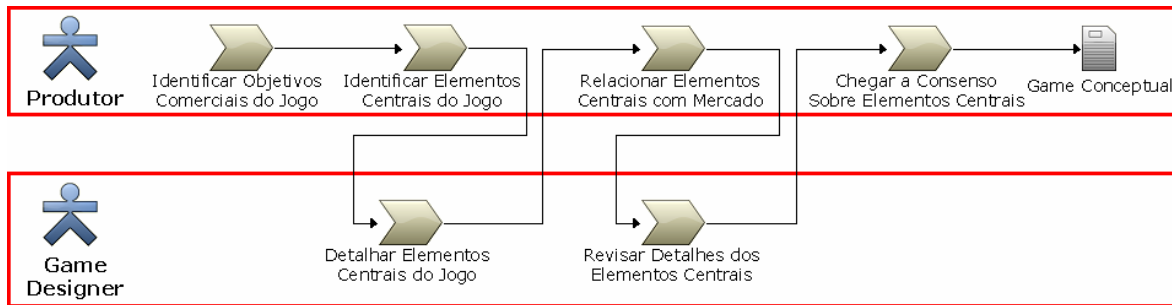


Figura 81 – Atividades da Disciplina Modelagem de Mercado

Identificados os objetivos comerciais do jogo, o produtor identifica os elementos centrais que o jogo deve ter. A partir de então, o *game designer* detalha estes elementos.

Em seguida, o produtor relaciona os elementos centrais detalhados com as necessidades de entretenimento do mercado para analisar o apelo comercial do jogo. Após atividades de revisão, chega-se a um consenso sobre os elementos centrais. Os elementos centrais consensuais são, então, registrados no *Game Conceptual Document*.

5.4.2 Game Design

A disciplina *Game Design* engloba o conjunto de atividades comprometidas com a definição e especificação principalmente dos elementos de arte e áudio do jogo, em um nível de detalhes que permita o início da fase de produção. Esta disciplina congrega 7 papéis, 16 atividades e 4 artefatos. Veja na Figura 82 uma visão geral da disciplina.

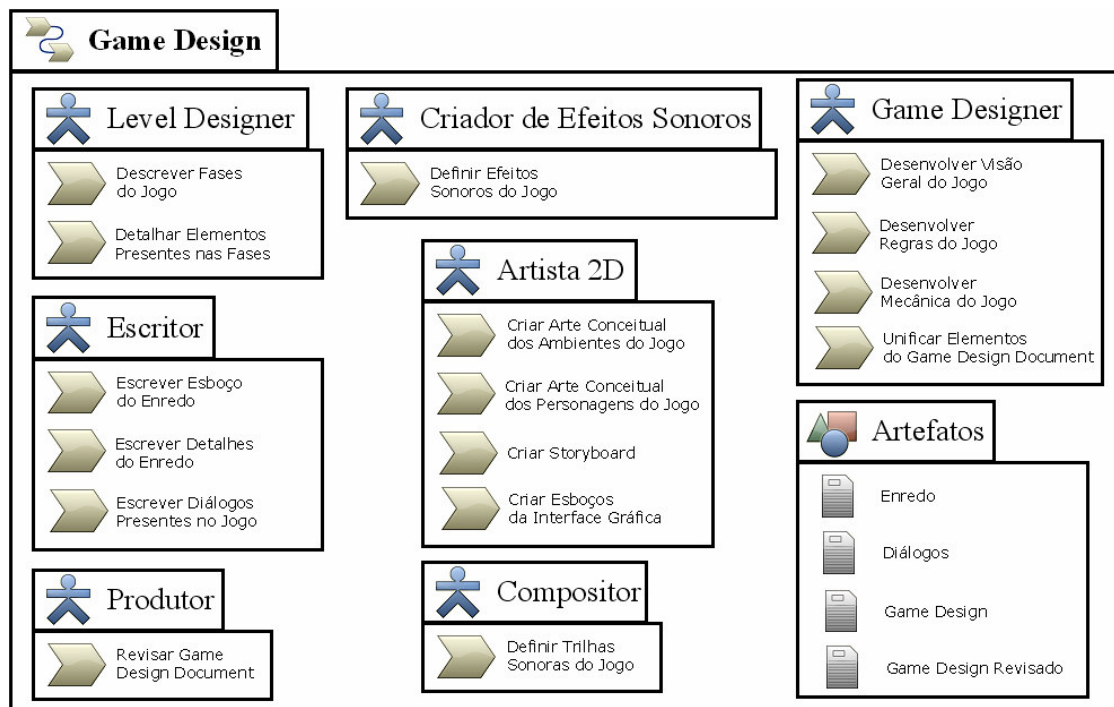


Figura 82 – Visão Geral da Disciplina Game Design

Os papéis presentes nesta disciplina são:

1. Level Designer

Responsável por criar os níveis (fases) do jogo. Geralmente é atribuição deste papel a integração dos componentes de arte, áudio e código com o objetivo de criar o fluxo do jogo.

2. Escritor

Responsável por escrever o enredo do jogo. Ou seja, cria desde diálogos até elementos narrativos.

3. Produtor

Este papel já foi descrito anteriormente.

4. Criador de Efeitos Sonoros

Responsável por criar os efeitos sonoros presentes no jogo.

5. Artista 2D

Cria os elementos artísticos bidimensionais do jogo. Por exemplo: texturas, interfaces gráficas (caso estas sejam bidimensionais), desenhos, *storyboards*, etc.

6. Compositor

Compõe as trilhas musicais que aparecerão ao longo do fluxo do jogo.

7. Game Designer

Este papel já foi descrito anteriormente.

Os principais artefatos produzidos nesta disciplina são:

1. Enredo

Contém o texto referente ao enredo do jogo.

2. Diálogos

Contêm os trechos de diálogos dos personagens do jogo.

3. Game Design (Revisado)

Uma das principais referências para todo o ciclo de desenvolvimento do jogo. Reúne informações detalhadas dos elementos do jogo. No entanto, sem uma perspectiva mais técnica em termos de codificação.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 83.

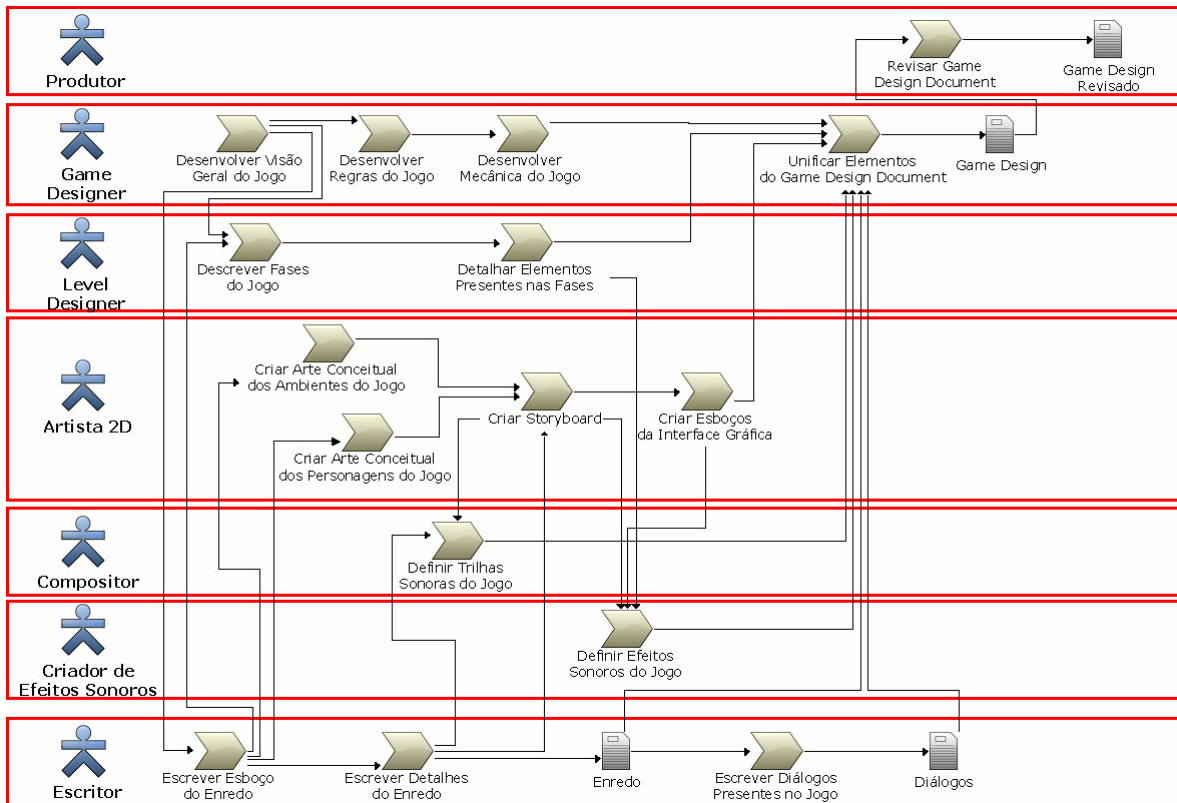


Figura 83 – Atividades da Disciplina *Game Design*

O *game designer* descreve uma visão geral do jogo que servirá como base para as atividades dos demais papéis. Em seguida, o *game designer* detalha estes elementos e desenvolve a mecânica do jogo. Entende-se por mecânica do jogo um conjunto de regras que permite criar um fluxo de jogo. Por fim, este papel é responsável por unificar os elementos produzidos pelos demais papéis em uma única visão: o *Game Design Document* que é então revisado pelo produtor.

A partir da visão geral do jogo e de um esboço do enredo, o *level designer* preocupa-se em descrever e detalhar como serão as fases do jogo.

O esboço do enredo do jogo é criado por um escritor que posteriormente evolui este para um enredo detalhado que é registrado em um documento específico. Geralmente, o escritor também cria um documento com os diálogos que existirão entre personagens durante o jogo.

O artista 2D, a partir principalmente do esboço do enredo, desenvolve a arte conceitual dos ambientes e personagens presentes na história do jogo. Quando o enredo evolui para uma versão mais detalhada, o artista 2D, utilizando os elementos de arte conceitual criados, cria *storyboards* ilustrando a sequência de eventos do enredo. Por fim, este papel também é responsável por criar esboços da interface gráfica do jogo.

A partir dos resultados das atividades do artista 2D, do *level designer* e do escritor, o compositor e criador de efeitos sonoros criam, respectivamente, as trilhas musicais e os efeitos sonoros do jogo.

5.4.3 Technical Design

A disciplina *Technical Design* engloba o conjunto de atividades comprometidas com a definição e especificação dos elementos técnicos do código do jogo, em um nível

de detalhes que permita o início da fase de produção. Esta disciplina congrega 3 papéis, 11 atividades e 1 artefato. Veja na Figura 84 uma visão geral da disciplina.

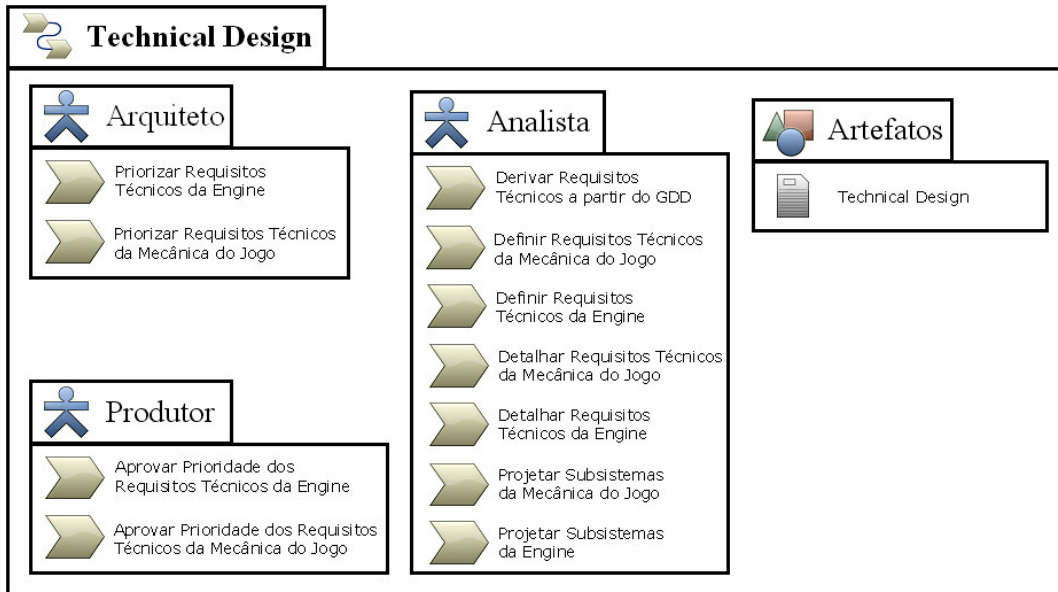


Figura 84 – Visão Geral da Disciplina *Technical Design*

Os papéis presentes nesta disciplina são:

1. Arquiteto

Responsável por identificar que funcionalidades do jogo são mais críticas em termos de esforço de codificação e por criar uma arquitetura que comporte a implementação destas funcionalidades.

2. Analista

Responsável por derivar requisitos a partir dos elementos descritos no *Game Design Document*. Requisitos estes que serão descritos em modelos que representem funcionalidades.

3. Produtor

Este papel já foi descrito anteriormente.

O principal artefato produzido nesta disciplina é:

1. *Technical Design*

É uma descrição técnica do que terá que ser implementado para que os elementos descritos no *Game Design Document* possam no futuro ser integrados em torno de um elemento só: o jogo.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 85.

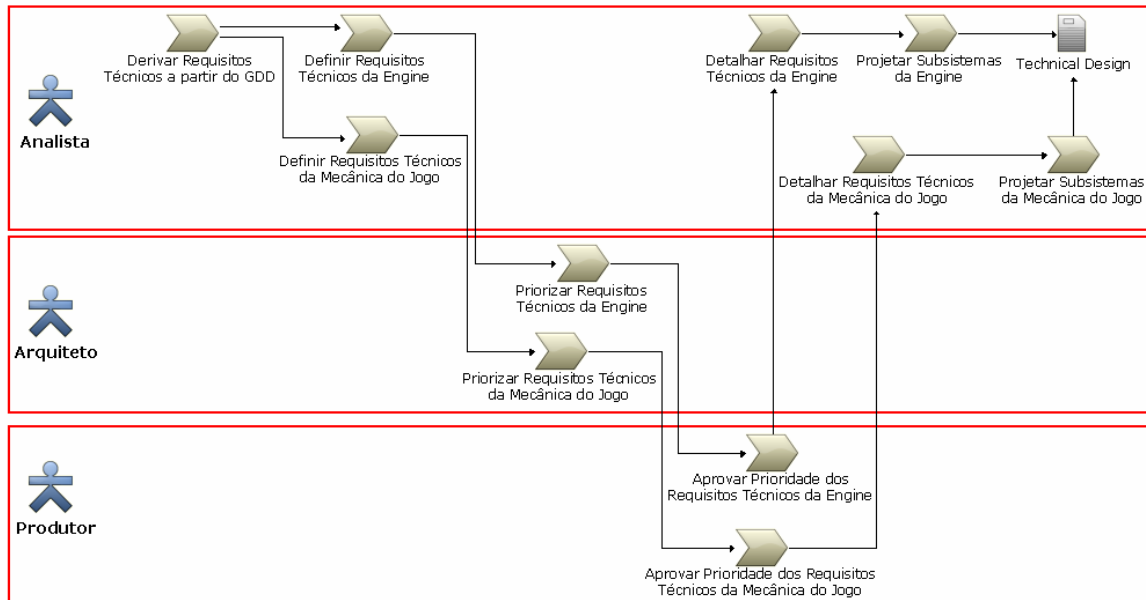


Figura 85 – Atividades da Disciplina *Technical Design*

O analista, a partir do *Game Design Document*, deriva requisitos técnicos que serão desdobrados em requisitos técnicos da *engine* e da mecânica do jogo. O arquiteto, então, prioriza estes requisitos tendo em vista a criticidade destes em relação à arquitetura do jogo como um todo.

O produtor tem um papel rápido nesta disciplina ao aprovar esta priorização feita pelo arquiteto. Em seguida, o analista irá detalhar e projetar os requisitos priorizados da *engine* e da mecânica do jogo.

5.4.4 Implementação

A disciplina Implementação engloba o conjunto de atividades comprometidas com a criação dos componentes de código e posterior integração destes com o conteúdo artístico e sonoro. Esta disciplina congrega 3 papéis, 9 atividades e 1 artefato. Veja na Figura 86 uma visão geral da disciplina.

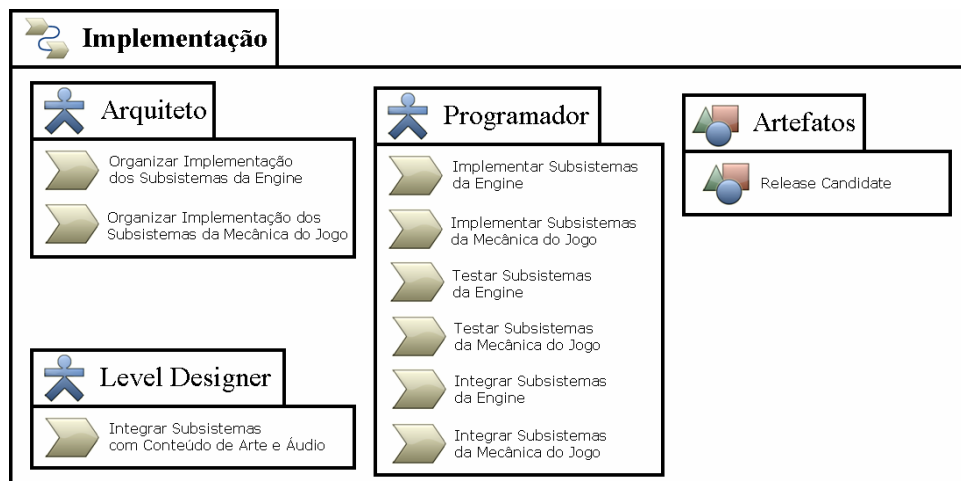


Figura 86 – Visão Geral da Disciplina Implementação

Os papéis presentes nesta disciplina são:

1. Arquiteto

Este papel já foi descrito anteriormente.

2. Level Designer

Este papel já foi descrito anteriormente.

3. Programador

Responsável por implementar os componentes de código do jogo.

O principal artefato produzido nesta disciplina é:

1. Release Candidate

O *Release Candidate* é uma versão estável do jogo com os componentes de arte, áudio e código integrados.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 87.

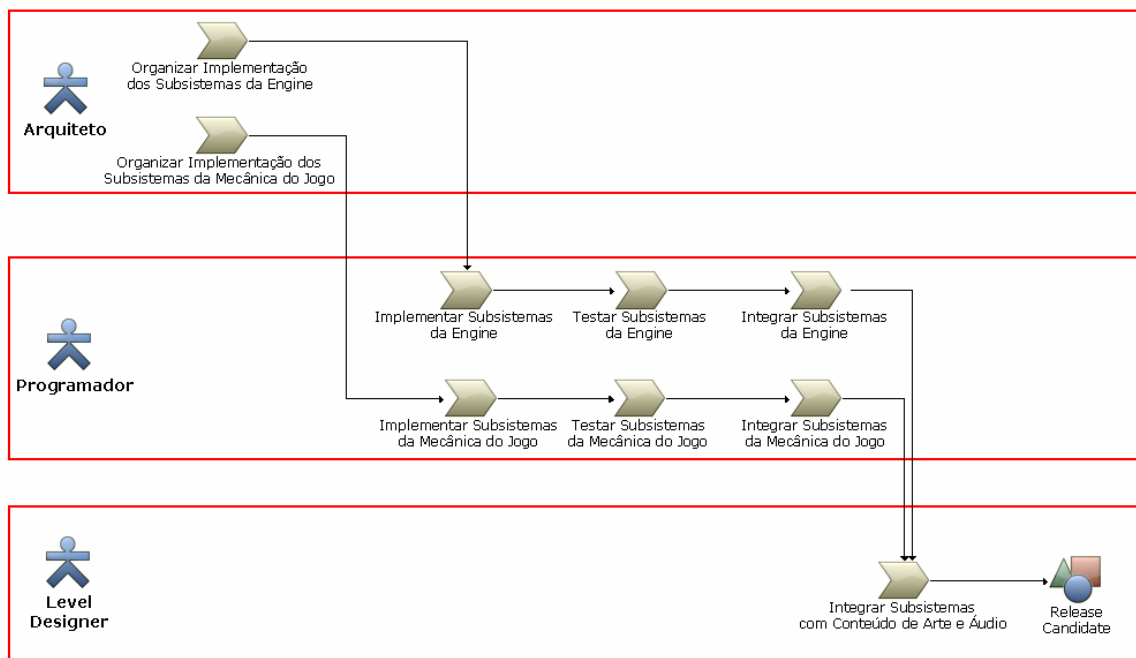


Figura 87 – Atividades da Disciplina Implementação

O arquiteto organiza como será a arquitetura dos subsistemas que serão implementados em relação à *engine* e à mecânica do jogo. Em seguida, o programador implementa estes subsistemas, realiza testes unitários e, por fim, integra-os.

Após se ter os subsistemas da *engine* e da mecânica do jogo integrados, parte-se então para a integração com os demais componentes (arte e áudio). Atividade esta realizada pelo *level designer*.

5.4.5 Criação de Arte 2D

A disciplina Criação de Arte 2D engloba o conjunto de atividades comprometidas com a criação dos componentes artísticos bidimensionais do jogo. Exemplos de componentes bidimensionais são: arte conceitual, texturas, etc. Esta disciplina congrega 2 papéis, 10 atividades e 5 artefatos. Veja na Figura 88 uma visão geral da disciplina.

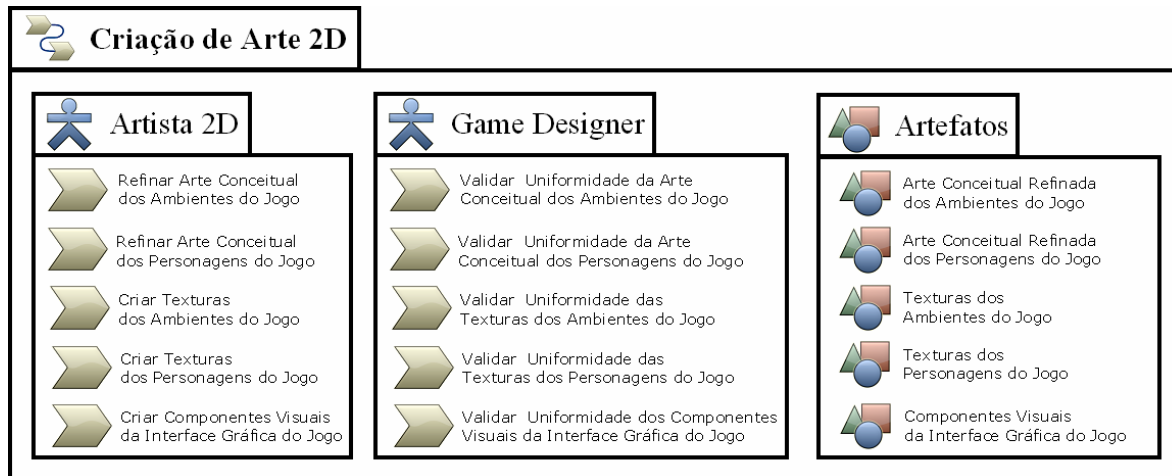


Figura 88 – Visão Geral da Disciplina Criação de Arte 2D

Os papéis presentes nesta disciplina são:

- 1. Artista 2D**
Este papel já foi descrito anteriormente.
- 2. Game Designer**
Este papel já foi descrito anteriormente.

Os principais artefatos produzidos nesta disciplina são:

- 1. Arte Conceitual Refinada dos Ambientes do Jogo**
A arte conceitual dos ambientes são desenhos ilustrando como serão os ambientes do jogo. Este artefato guia o trabalho de modelagem 3D realizado em outro momento.
- 2. Arte Conceitual Refinada dos Personagens do Jogo**
A arte conceitual dos personagens são desenhos ilustrando como serão os personagens do jogo. Este artefato guia o trabalho de modelagem 3D realizado em outro momento.
- 3. Texturas dos Ambientes do Jogo**
São imagens que são mapeadas nas superfícies dos ambientes com o objetivo de representar as superfícies destes.
- 4. Texturas dos Personagens do Jogo**
São imagens que são mapeadas nas superfícies dos personagens com o objetivo de representar as superfícies destes.

5. Componentes Visuais da Interface Gráfica do Jogo

São componentes gráficos bidimensionais que serão incluídos no jogo e associados uma funcionalidade para criar a interface gráfica com o usuário.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 89.

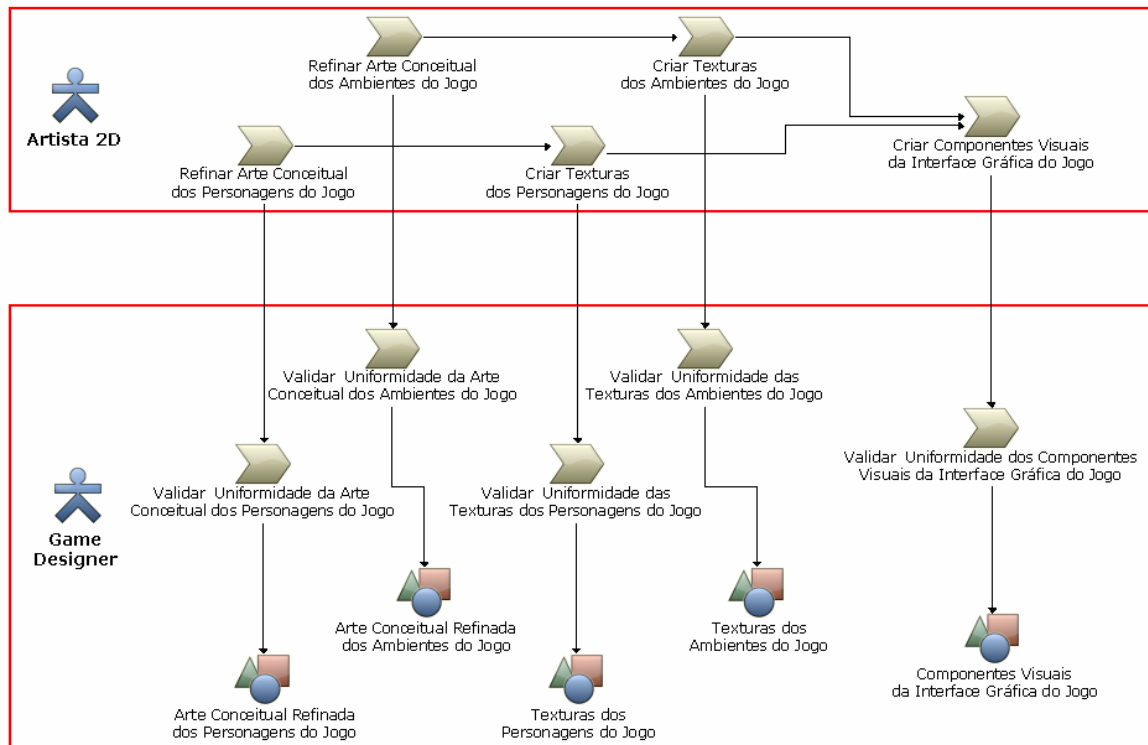


Figura 89 – Atividades da Disciplina Criação de Arte 2D

O artista 2D refina a arte conceitual dos ambientes e personagens criada em atividades anteriores de outra disciplinas (*Game Design*). Este conteúdo artístico é então enviado ao *game designer* que verifica a uniformidade destes com o conceito e elementos do jogo.

Quando os modelos 3D dos ambientes e personagens ficam prontos, o artista 2D cria as texturas destes. Texturas estas que também são enviadas para o *game designer* validar sua uniformidade.

Por fim, o artista 2D também cria os componentes visuais da interface gráfica do jogo. Componentes que são também validados em relação a sua uniformidade com o jogo pelo *game designer*.

5.4.6 Criação de Arte 3D

A disciplina Criação de Arte 3D engloba o conjunto de atividades comprometidas com a criação dos componentes artísticos tridimensionais do jogo e suas eventuais animações. Esta disciplina congrega 3 papéis, 12 atividades e 4 artefatos. Veja na Figura 90 uma visão geral da disciplina.

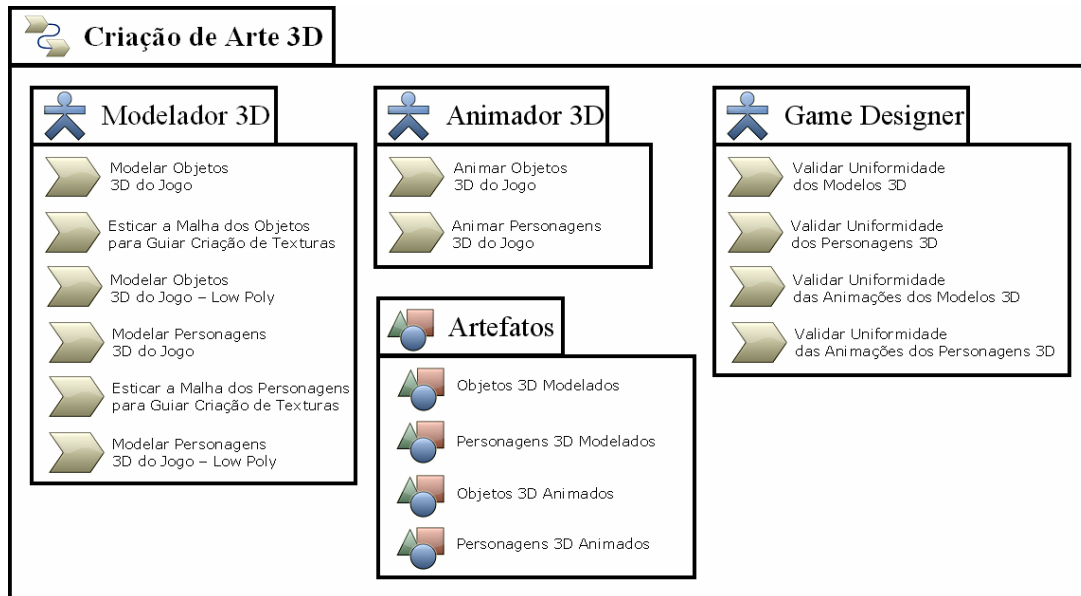


Figura 90 – Visão Geral da Disciplina Criação de Arte 3D

Os papéis presentes nesta disciplina são:

- 1. Modelador 3D**
Responsável por modelar o conteúdo tridimensional do jogo.
- 2. Animador 3D**
Responsável por animar o conteúdo tridimensional do jogo.
- 3. Game Designer**
Este papel já foi descrito anteriormente.

Os principais artefatos produzidos nesta disciplina são:

- 1. Objetos 3D Modelados**
São os objetos 3D do jogo modelados
- 2. Personagens 3D Modelados**
São os personagens 3D do jogo modelados
- 3. Objetos 3D Animados**
São os objetos 3D do jogo animados.
- 4. Personagens 3D Animados**
São os personagens 3D do jogo animados.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 91.

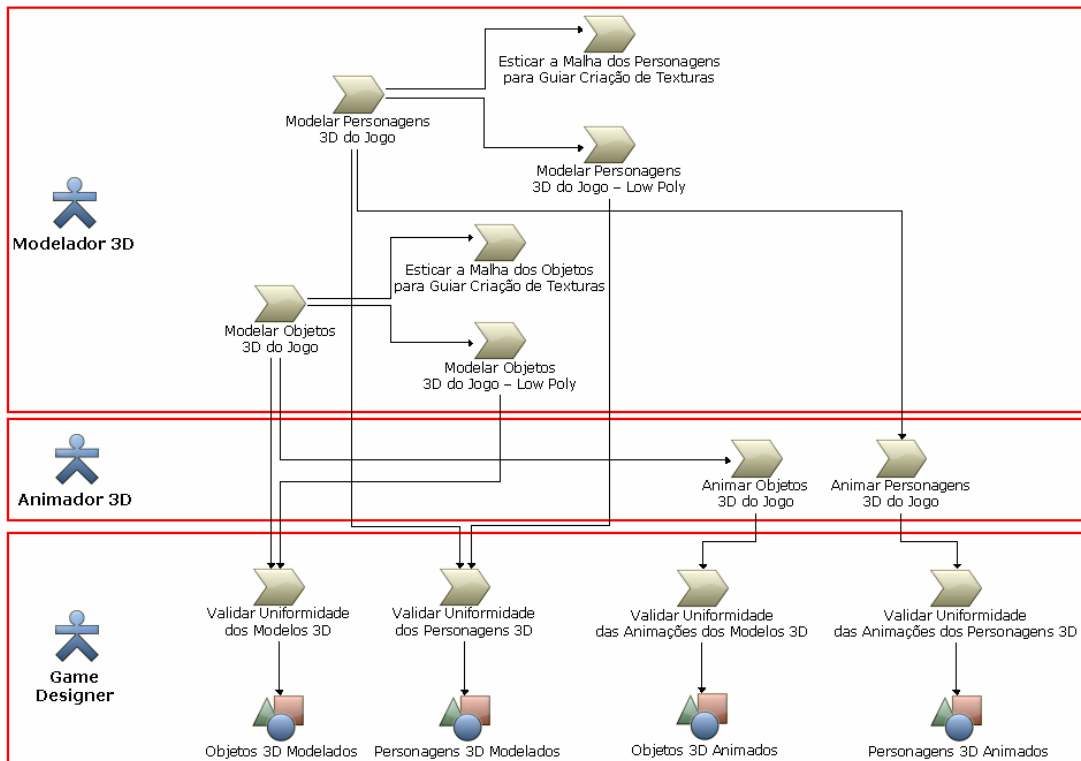


Figura 91 – Atividades da Disciplina Criação de Arte 3D

O modelador 3D cria os objetos e personagens tridimensionais do jogo. Em seguida, tanto para os objetos como para os personagens, o modelador 3D realiza uma atividade conhecida como esticar a malha dos modelos (objetos e personagens). O objetivo desta atividade é guiar a criação das texturas pelo artista 2D.

O modelador 3D também cria versões dos objetos e personagens tridimensionais conhecidas como *Low Poly* (poucos polígonos), utilizadas em circunstâncias específicas com o objetivo de diminuir o nível de processamento do jogo.

Os modelos 3D criados pelo modelador são então utilizados pelo animador 3D para criar as animações necessárias. Todo este conteúdo, modelos e animações 3D são, por fim, encaminhados ao *game designer* que verifica a uniformidade destes em relação ao conceito e elementos do jogo.

5.4.7 Criação de Áudio

A disciplina Criação de Áudio engloba o conjunto de atividades comprometidas com a criação das trilhas musicais e dos efeitos sonoros do jogo. Esta disciplina congrega 4 papéis, 8 atividades e 2 artefatos. Veja na Figura 92 uma visão geral da disciplina.

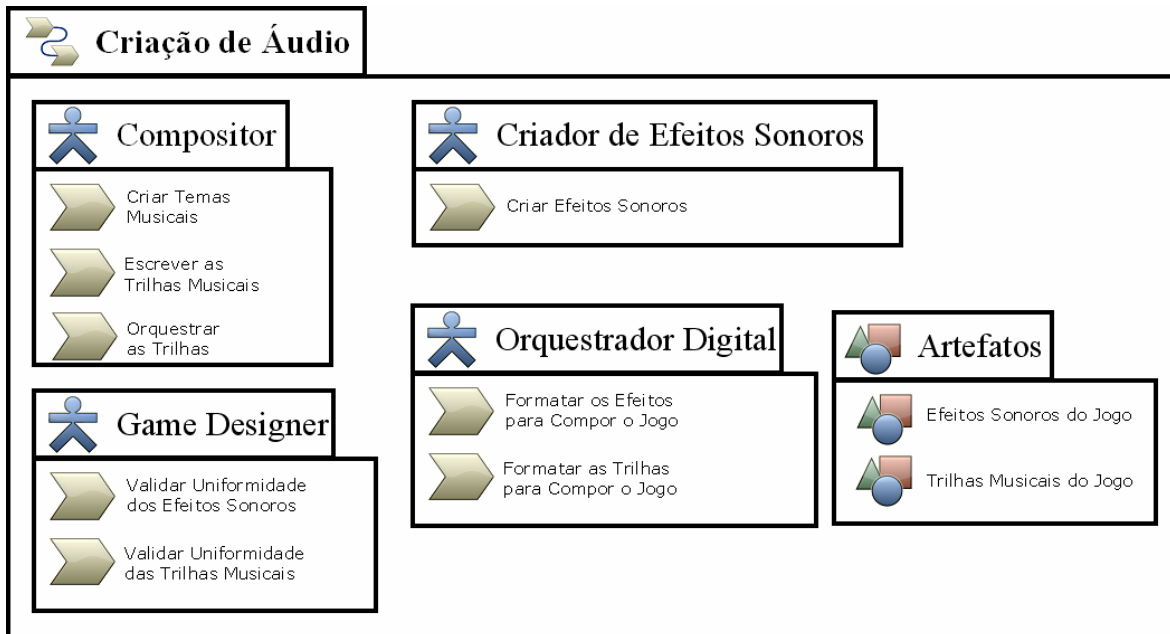


Figura 92 – Visão Geral da Disciplina Criação de Áudio

Os papéis presentes nesta disciplina são:

1. Compositor

Este papel já foi descrito anteriormente.

2. Criador de Efeitos Sonoros

Este papel já foi descrito anteriormente.

3. Orquestrador Digital

Responsável por editar e converter o conteúdo de áudio produzido pelo compositor e criador de efeitos sonoros em formatos que possam ser carregados e executados no jogo.

4. Game Designer

Este papel já foi descrito anteriormente.

Os principais artefatos produzidos nesta disciplina são:

1. Efeitos Sonoros do Jogo

São os efeitos sonoros que compõem o jogo.

2. Trilhas Musicais do Jogo

São as trilhas musicais que compõem o jogo.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 93.

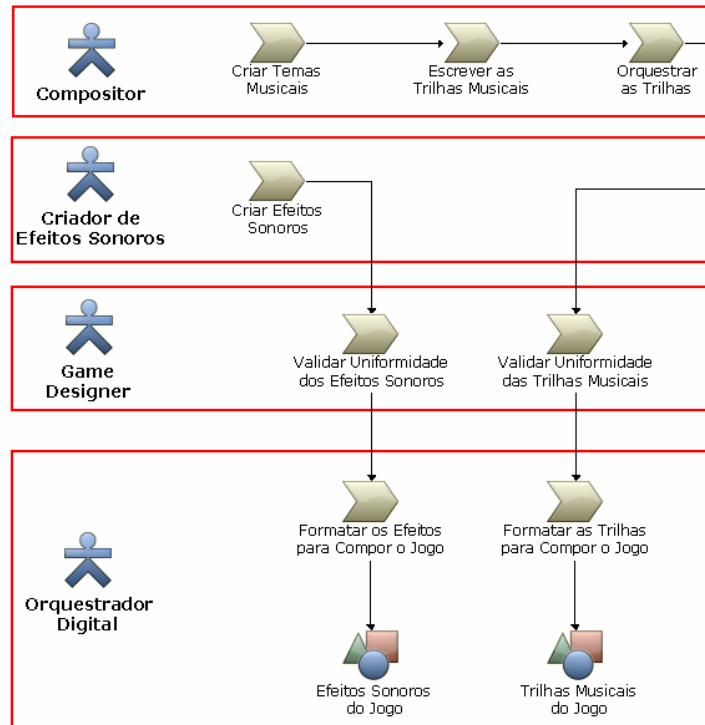


Figura 93 – Atividades da Disciplina Criação de Áudio

O compositor é o responsável por criar os temas musicais do jogo coerentes com momentos da história. Em seguida, estes temas inspiram a escrita de trilhas musicais que são orquestradas diretamente ou indiretamente pelo compositor.

Já o criador de efeitos sonoros é responsável por criar os efeitos sonoros presentes no jogo. Em seguida, este conteúdo (trilhas e efeitos) é encaminhado para o *game designer* que, mais uma vez, verifica a uniformidade destes com o conceito e elementos do jogo.

Por fim, o orquestrador digital converte as trilhas e efeitos para formatos que poderão ser carregados e reproduzidos no jogo.

5.4.8 Teste

A disciplina Teste engloba o conjunto de atividades comprometidas com testes. Estes testes podem ser divididos em dois grandes grupos: testes *alpha* e *beta*. Os primeiros envolvem essencialmente membros do desenvolvedor e publicadora. Já os testes *beta* costumam envolver amostras do público-alvo. Esta disciplina congrega 7 papéis, 29 atividades e 8 artefatos. Veja na Figura 94 uma visão geral da disciplina.

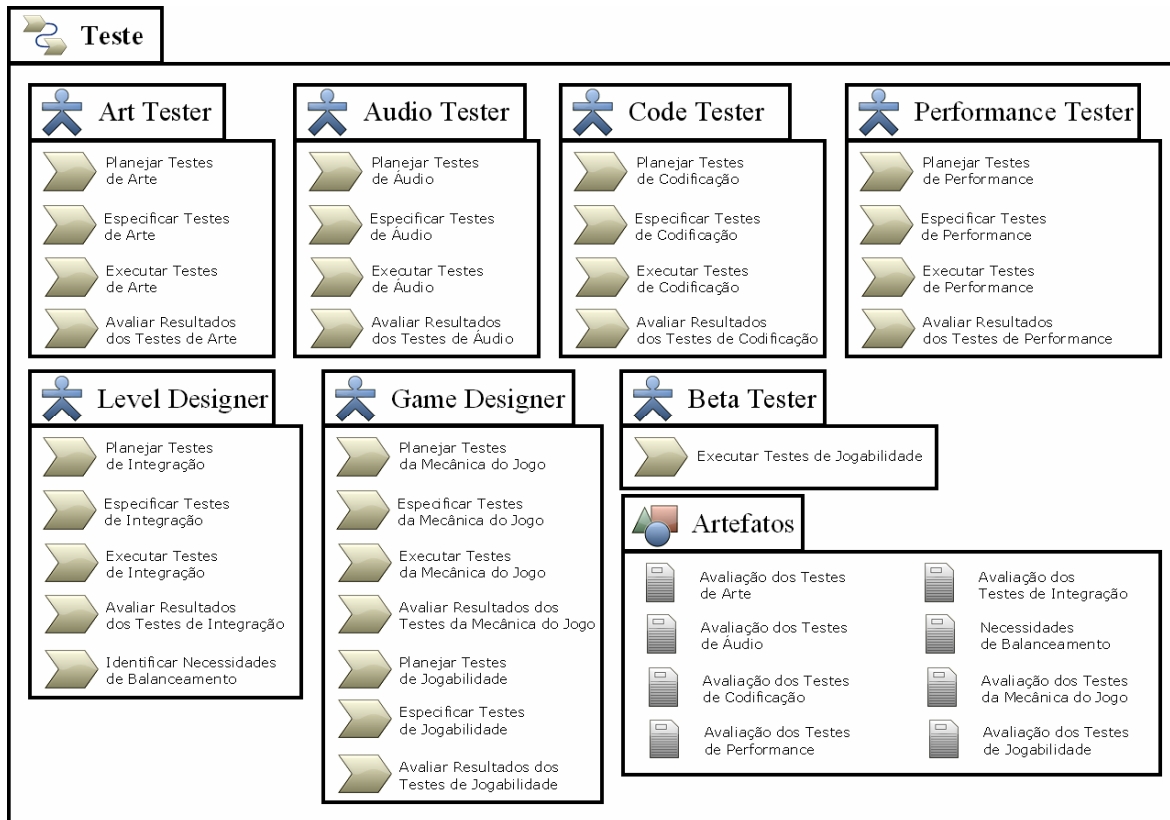


Figura 94 – Visão Geral da Disciplina Teste

Os papéis presentes nesta disciplina são:

- 1. Art Tester**
Responsável pelas atividades de testes do conteúdo artístico produzido.
- 2. Audio Tester**
Responsável pelas atividades de testes do conteúdo de áudio produzido.
- 3. Code Tester**
Responsável pelas atividades de testes do conteúdo de código produzido.
- 4. Performance Tester**
Responsável pelas atividades de testes em relação ao desempenho do jogo.
- 5. Level Designer**
Este papel já foi descrito anteriormente.
- 6. Game Designer**
Este papel já foi descrito anteriormente.
- 7. Beta Tester**
Amostra do público-alvo do jogo responsável por testar a jogabilidade (nível de dificuldade, grau de entretenimento, etc.) do jogo.

Os principais artefatos produzidos nesta disciplina são:

1. Avaliação dos Testes de Arte

Resultados obtidos através dos testes do conteúdo de arte.

2. Avaliação dos Testes de Áudio

Resultados obtidos através dos testes do conteúdo sonoro.

3. Avaliação dos Testes de Codificação

Resultados obtidos através dos testes do conteúdo de código.

4. Avaliação dos Testes de *Performance*

Resultados obtidos através dos testes em relação ao desempenho do jogo.

5. Avaliação dos Testes de Integração

Resultados obtidos através dos testes de integração do conteúdo de arte, código e áudio.

6. Necessidades de Balanceamento

Necessidades de balanceamento detectadas a partir dos testes de jogabilidade.

7. Avaliação dos Testes da Mecânica do Jogo

Resultados obtidos através dos testes em relação à mecânica do jogo.

8. Avaliação dos Testes de Jogabilidade

Resultados obtidos através dos testes de jogabilidade.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 95.

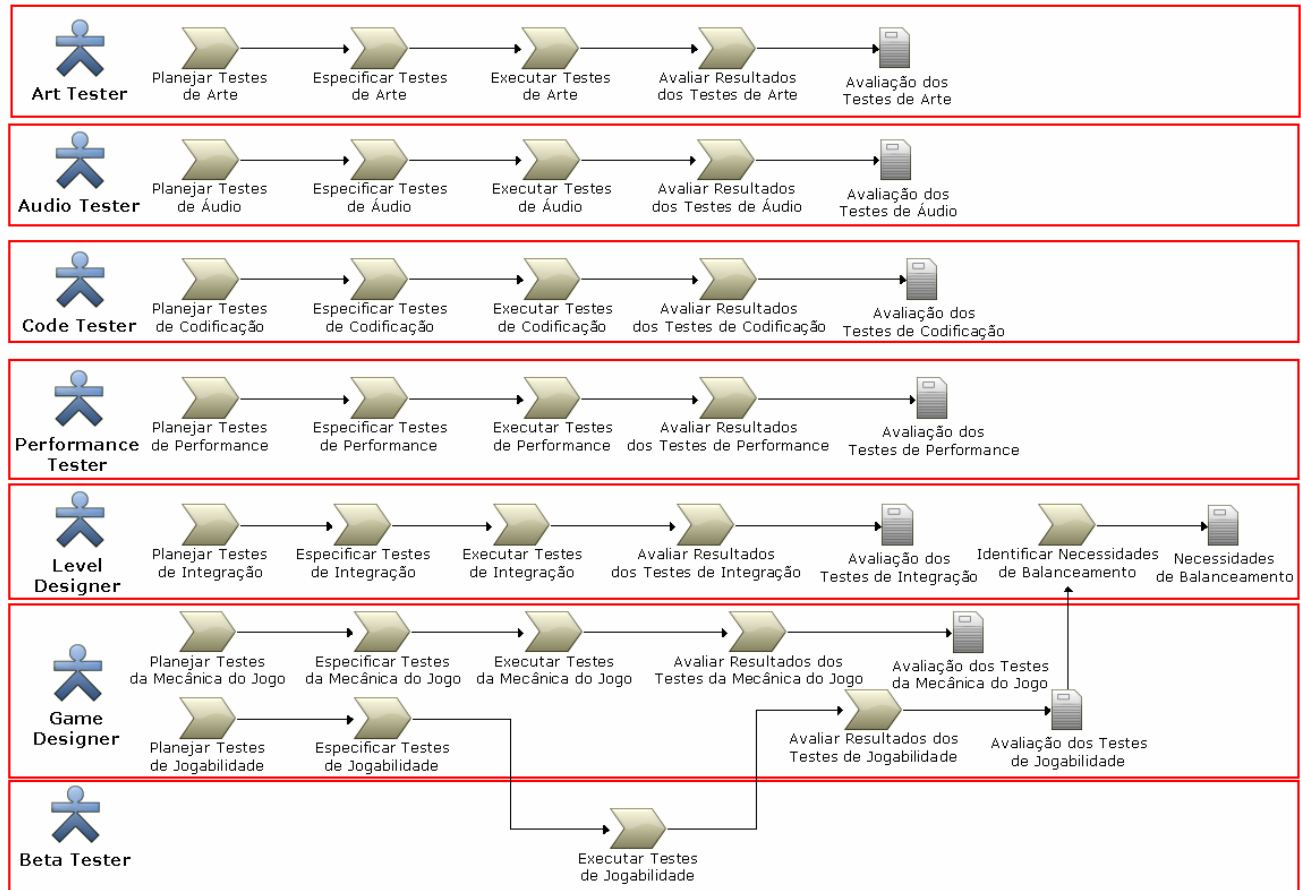


Figura 95 – Atividades da Disciplina Teste

De uma maneira geral, os testes são constituídos de quatro atividades sequenciais: planejar, especificar, executar e avaliar os testes. O *art tester*, *audio tester*, *code tester*, *performance tester*, *level designer* e *game designer* fazem estas atividades para a parte de arte, áudio, código, performance, integração e mecânica do jogo, respectivamente. Estas atividades fazem parte dos testes *alpha*.

Os testes *beta* são conduzidos inicialmente pelo *game designer*, responsável por planejar e especificar os testes de jogabilidade. No entanto, quem executa os testes de jogabilidade é o *beta tester*. Posteriormente, o *game designer* avalia os resultados destes testes e, em seguida, repassa estes resultados para o *level designer* identificar necessidades de balanceamento no jogo.

5.4.9 Distribuição

A disciplina Distribuição engloba o conjunto de atividades comprometidas com a finalização do jogo e eventuais encartes que serão encaminhados para a publicadora distribuir através dos seus canais. Esta disciplina congrega 6 papéis, 8 atividades e 4 artefatos. Veja na Figura 96 uma visão geral da disciplina.

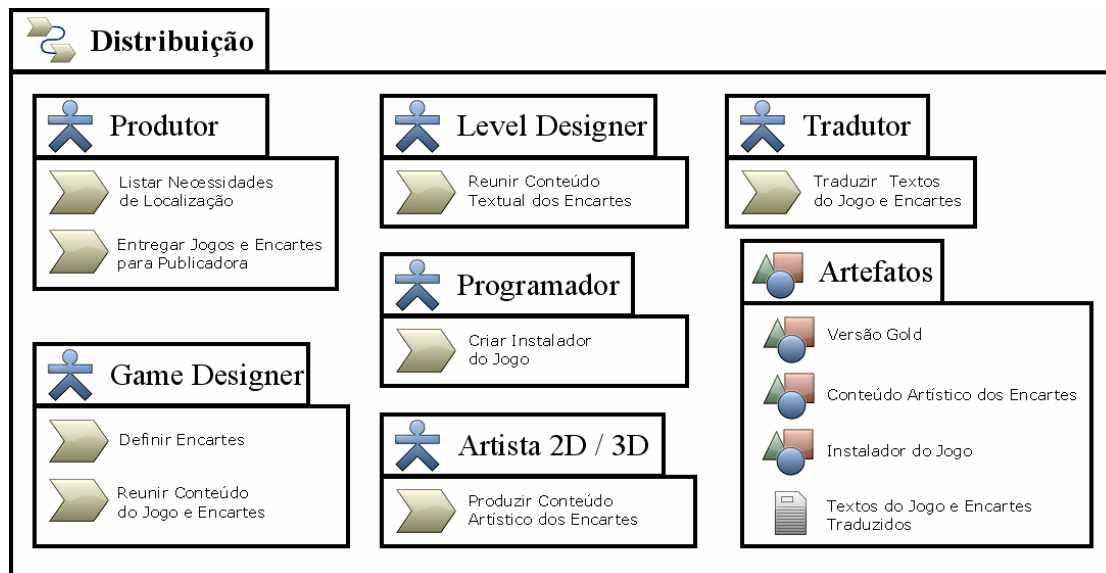


Figura 96 – Visão Geral da Disciplina Distribuição

Os papéis presentes nesta disciplina são:

- 1. Produtor**
Este papel já foi descrito anteriormente.
- 2. Game Designer**
Este papel já foi descrito anteriormente.
- 3. Level Designer**
Este papel já foi descrito anteriormente.
- 4. Programador**
Este papel já foi descrito anteriormente.
- 5. Artista 2D / 3D**
Este papel já foi descrito anteriormente.
- 6. Tradutor**
Responsável por traduzir os textos do jogo e encartes para as línguas dos países nos quais o jogo será publicado.

Os principais artefatos produzidos nesta disciplina são:

- 1. Versão Gold**
A versão *gold* é a versão final de todo o conteúdo que compõe o jogo que será gravada em alguma mídia específica (geralmente CD) e entregue aos canais de distribuição pela publicadora.
- 2. Conteúdo Artístico dos Encartes**
É o conteúdo de arte presente em eventuais encartes (manuais, *pôster*, etc.) do jogo.

3. Instalador do Jogo

É um programa executável responsável por instalar a versão *gold* do jogo.

4. Textos do Jogo e Encartes Traduzidos

São todos os textos do jogo e eventuais encartes traduzidos para as línguas dos países nos quais o jogo será publicado.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 97.

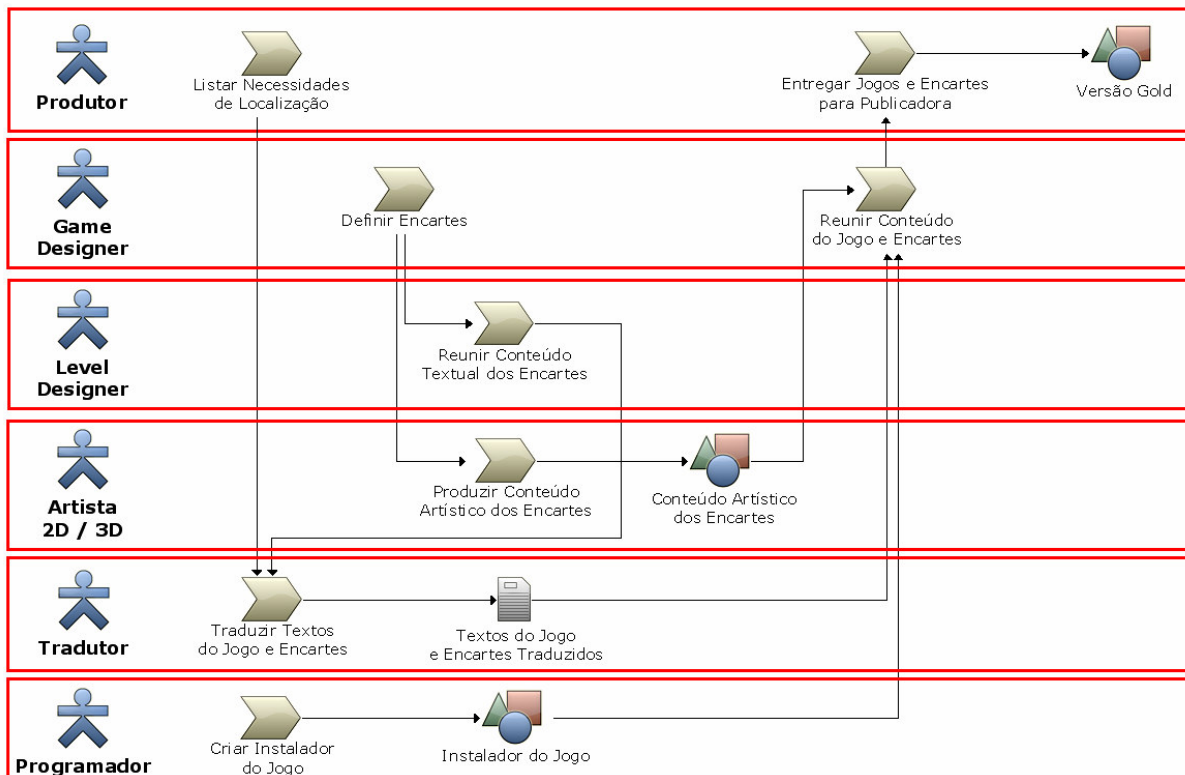


Figura 97 – Atividades da Disciplina Distribuição

O *game designer* define que encartes serão inclusos no jogo. Em seguida, o *level designer* começa a reunir os textos que estarão presentes nestes encartes. No mesmo momento, o artista 2D e o artista 3D produzem o conteúdo artístico destes encartes.

Quando o *level designer* reúne todos os textos dos encartes, o tradutor traduz estes para as línguas dos países nos quais o jogo será publicado. Já o programador, cria um instalador para a versão final do jogo.

Todo este conteúdo (instalador, textos e conteúdo artístico dos encartes) é reunido pelo *game designer* que repassa para o produtor entregar a chamada versão *gold* do jogo: todo o conteúdo que compõe este.

5.4.10 Gerência de Configuração e Mudanças

A disciplina Gerência de Configuração e Mudanças engloba o conjunto de atividades comprometidas com o monitoramento das mudanças solicitadas e ocorridas no conteúdo utilizado e produzido no ciclo de desenvolvimento do jogo. Esta disciplina

congrega 3 papéis, 8 atividades e 1 artefato. Veja na Figura 98 uma visão geral da disciplina.

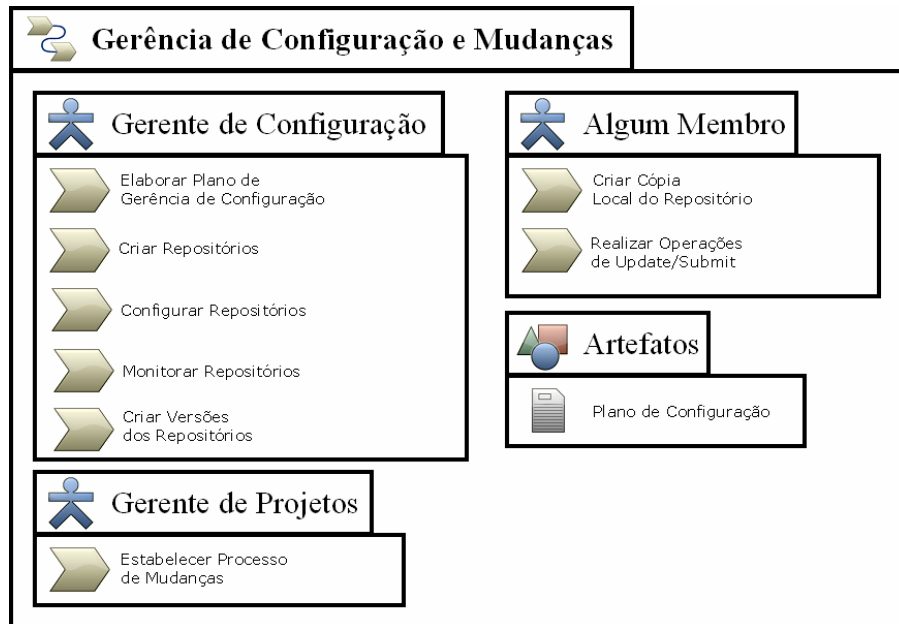


Figura 98 – Visão Geral da Disciplina Gerência de Configuração e Mudanças

Os papéis presentes nesta disciplina são:

1. Gerente de Configuração

Responsável pelo monitoramento das modificações solicitadas e ocorridas durante o ciclo de desenvolvimento do jogo.

2. Gerente de Projetos

Responsável direto pela gerência das atividades do projeto.

3. Algun Membro

Pode ser potencialmente qualquer membro da equipe de criação do jogo

O principal artefato produzido nesta disciplina é:

1. Plano de Configuração

O plano de configuração descreve os procedimentos de gerenciamento de configuração e controle de mudanças.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 99.

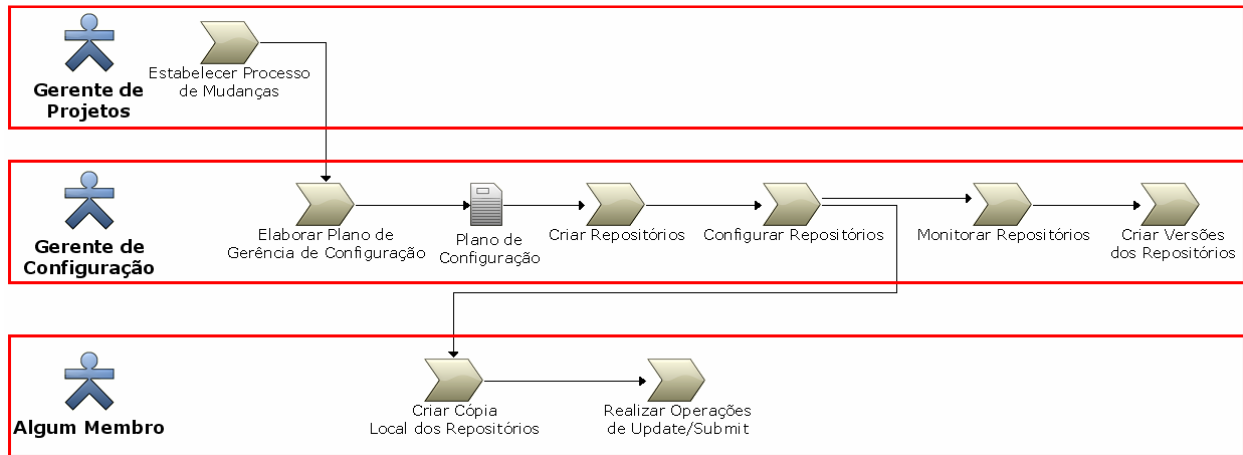


Figura 99 – Atividades da Disciplina Gerência de Configuração e Mudanças

O gerente de projetos estabelece qual será o processo para gerenciar as mudanças. Então, o gerente de configuração elabora um plano de gerência de configuração.

O gerente de configuração também cria e configura os repositórios que serão utilizados durante todo o projeto. Logo, estes repositórios são constantemente monitorados para, quando necessário, criar versões deste.

Por fim, todos os membros da equipe do projeto devem sempre ter uma cópia local dos repositórios e realizar operações de *update* e *submit* de modificações estáveis.

5.4.11 Gerência de Projeto

A disciplina Gerência de Projeto engloba o conjunto de atividades comprometidas com o gerenciamento do projeto. Esta disciplina congrega 1 papel, 7 atividades e 4 artefatos. Veja na Figura 100 uma visão geral da disciplina.

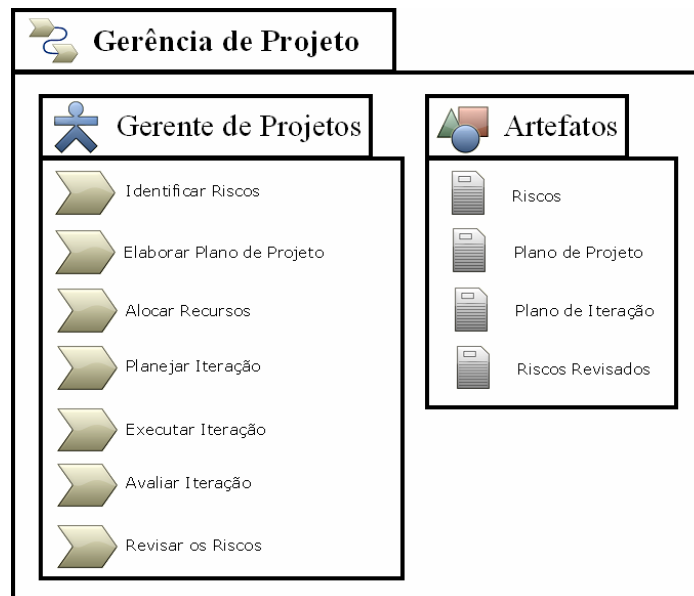


Figura 100 – Visão Geral da Disciplina Gerência de Projeto

O papel presente nesta disciplina é:

1. Gerente de Projetos

Este papel já foi descrito anteriormente.

Os principais artefatos produzidos nesta disciplina são:

1. Riscos

Uma lista de riscos relacionados com o projeto listados a partir de uma ordem decrescente de importância. Para cada risco, é preciso associar um plano de mitigação e contingência.

2. Plano de Projeto

Reúne os elementos principais do projeto que precisam ser monitorados e controlados: escopo, tempo e custo. Este documento detalha como será realizada a gerência do projeto.

3. Plano de Iteração

É um plano detalhado com seqüências de atividades que devem ser realizadas na iteração consumindo certos recursos. É detalhado o suficiente para permitir uma gerência efetiva da iteração.

4. Riscos Revisados

É a lista de riscos levantada anteriormente revisada a partir da evolução do projeto ao longo da iteração.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 101.

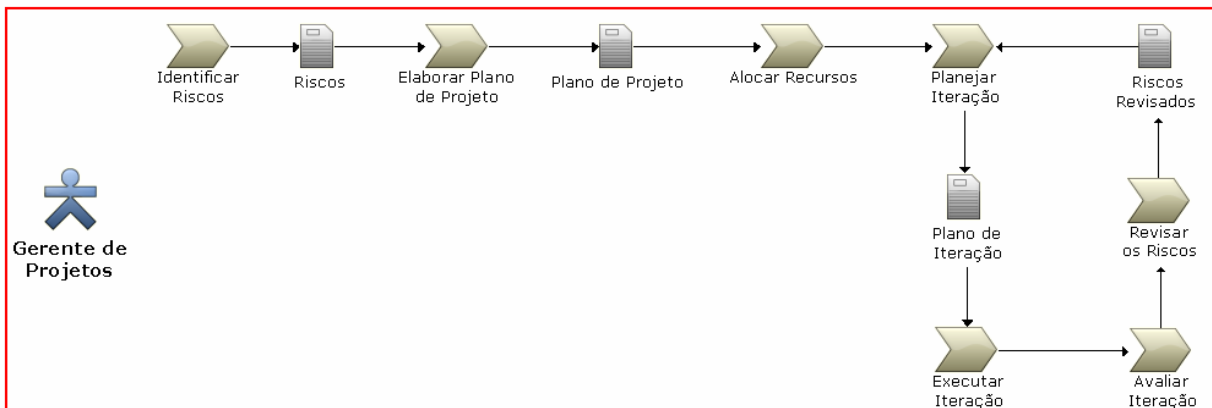


Figura 101 – Atividades da Disciplina Gerência de Projeto

O gerente de projetos identifica os riscos, elabora um plano de projeto e aloca recursos. Após estas atividades iniciais, cada iteração é planejada, executada e avaliada. Esta avaliação permite uma atualização dos riscos que irá influenciar no planejamento da próxima iteração. Este ciclo segue até o final do projeto.

5.4.12 Ambiente

A disciplina Ambiente engloba o conjunto de atividades comprometidas com o suporte de processos e ferramentas que auxiliem o ciclo de produção do jogo. Esta disciplina congrega 2 papéis, 7 atividades e 1 artefato. Veja na Figura 102 uma visão geral da disciplina.

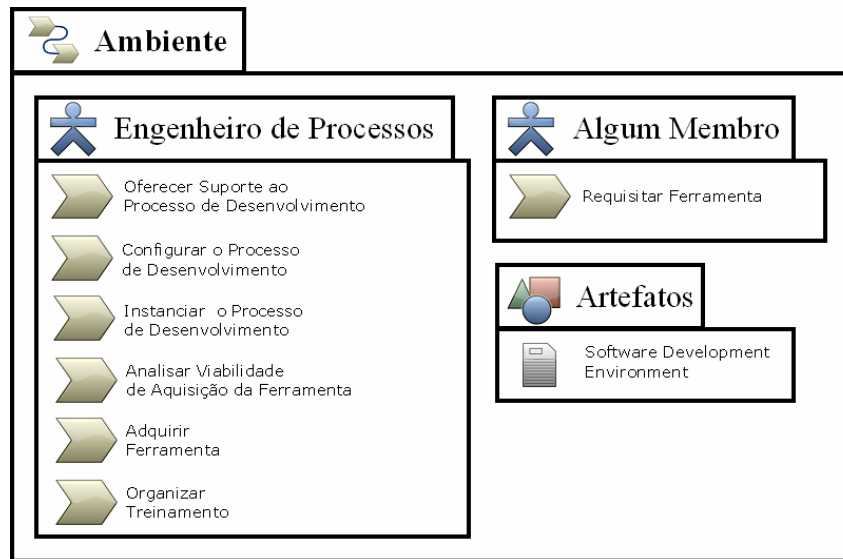


Figura 102 – Visão Geral da Disciplina Ambiente

Os papéis presentes nesta disciplina são:

1. Engenheiro de Processos

Responsável por adaptar processos que auxiliem o ciclo de produção. Também é responsável por adquirir ferramentas que auxiliem a equipe de projeto.

2. Algum Membro

Este papel já foi descrito anteriormente.

O principal artefato produzido nesta disciplina é:

1. Software Development Environment

Descreve com detalhes o ambiente de desenvolvimento do *software*. Ou seja, quais processos, ferramentas e padrões serão utilizados no desenvolvimento.

As atividades, papéis e artefatos acima apresentados relacionam-se de acordo com o fluxo de atividades presente na Figura 103.

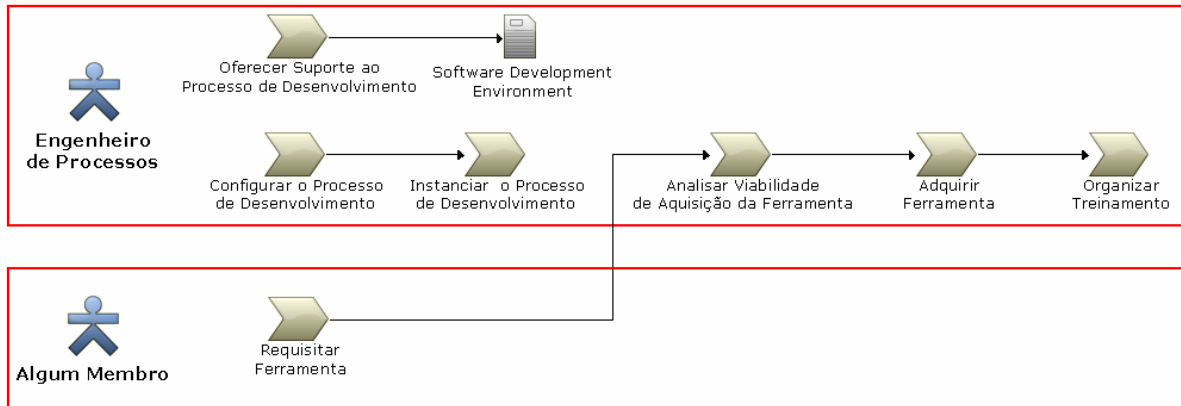


Figura 103 – Atividades da Disciplina Ambiente

O engenheiro de suporte detalha os elementos de suporte ao processo de desenvolvimento do jogo. Também faz parte das atividades do engenheiro de processos configurar este processo e instanciá-lo para a realidade do projeto e da organização em questão.

Eventualmente, algum membro pode vir a requisitar uma ferramenta de apoio ao desenvolvimento. O engenheiro de processos fica encarregado de convocar especialistas para analisar a viabilidade da aquisição desta ferramenta.

Caso chegue-se a um parecer favorável, o engenheiro de processos dá início ao processo de aquisição da ferramenta e começa a organizar, em conjunto com os especialistas, um treinamento da equipe de projeto na ferramenta adquirida.

6. Conclusões

Ao longo deste trabalho conheceu-se pouco a pouco a indústria de entretenimento digital, em particular, o universo dos jogos digitais.

Este trabalho começou provendo um panorama geral deste mercado, no qual se analisou a crescente busca pelo entretenimento digital e conheceram-se os muitos gêneros e mídias de jogos digitais.

A partir dos dados quantitativos e qualitativos apresentados ao longo do Capítulo 2, observou-se que apesar da recente história da indústria de jogos digitais, esta evoluiu bastante desde as suas origens e encontra-se em um claro estado de expansão crescente.

Esta expansão exige mudanças nos modos de conceber, produzir e publicar jogos digitais. Logo, pode-se observar uma preocupação crescente do mercado em estruturar, formalizar e automatizar o ciclo de desenvolvimento de jogos.

Investigando com mais detalhes o desenvolvimento de jogos para plataforma *PC*, com demandas de modelos melhor concebidos e estruturados, no Capítulo 3 conheceu-se a cadeia de valor deste segmento, as etapas do ciclo de desenvolvimento, assim como os atores e as principais atividades que fazem parte deste ciclo.

Levantaram-se, então, os principais desafios e características pertinentes ao processo de desenvolvimento de jogos de computador. Características estas que foram utilizadas no fim do Capítulo 4 para identificar que metodologia de desenvolvimento de *software* melhor atende à realidade de criação de jogos para *PC*.

O Capítulo 4 investigou as duas grandes correntes de modelos de desenvolvimento *software*: as ágeis e as preditivas. A análise realizada na parte final deste capítulo levou a conclusão que um modelo de desenvolvimento mais preditivo e baseado nas essências do RUP melhor se enquadra às características da criação de jogos para *PC*.

Construiu-se, portanto, um modelo preditivo de desenvolvimento de jogos de computador inspirado na essência do RUP: o MDJ-PC (modelo preditivo de desenvolvimento de jogos para *PC*).

As práticas, características, fases e disciplinas deste modelo foram apresentadas e detalhadas no Capítulo 5. Veja nas próximas subseções as principais contribuições e trabalhos futuros inerentes ao MDJ-PC.

6.1 Contribuições

Veja a seguir quais são as principais contribuições do estudo realizado neste trabalho de graduação para academias, instituições e pessoas envolvidas com o mercado de jogos digitais.

1. Descrição do mercado de jogos digitais

O Capítulo 2 fornece uma descrição atualizada do cenário passado e atual do mercado de jogos digitais. Portanto, este capítulo pode ser utilizado para conhecer um pouco da evolução, saber os principais acontecimentos de 2006 na área e quais são as perspectivas futuras deste mercado.

2. Descrição do ciclo de produção de jogos para *PC*

O Capítulo 3 contém uma descrição das etapas, atores, atividades e artefatos presentes no ciclo de produção de jogos de acordo com Tom Sloper. Este capítulo pode, logo, ser utilizado como referência para entender os principais aspectos existentes no desenvolvimento de um jogo para plataforma *PC*.

3. Levantamento de características da produção de jogos para PC

Ainda no Capítulo 3, levantam-se as principais características presentes na produção de jogos para PC. Esta enumeração permite reflexões a cerca de que modelos de desenvolvimento melhor se enquadram à realidade de cada projeto, de cada organização.

4. Análise de modelos de desenvolvimento de software aplicados a jogos

A análise comparativa, realizada no final do Capítulo 4, de como modelos de desenvolvimento de software atendem ou não às necessidades da produção de jogos, permite entender melhor como os modelos atuais podem ser adaptados ao mercado de jogos.

5. Criação de um modelo preditivo de desenvolvimento de jogos para PC

Por fim, a principal contribuição deste trabalho de graduação é a definição de um modelo preditivo, iterativo e incremental, inspirado na essência do RUP, para o desenvolvimento de jogos de PC.

Este modelo permite uma melhor estruturação do processo de criação de jogos, o que permite gerenciar melhor o projeto e, conseqüentemente, melhorar o aproveitamento dos recursos consumidos.

6.2 Trabalhos Futuros

Podem ser propostos os seguintes trabalhos futuros desenvolvidos em cima do MDJ-PC, como também em um escopo mais abrangente (modelos de desenvolvimento de jogos).

1. Estudos de casos

Os princípios e características do MDJ-PC precisam ser analisados e validados em estudos de casos reais que preencham as características às quais o modelo se aplica: projetos de longa duração e com equipes heterogêneas. Em função do tempo que concerne a produção de um trabalho de graduação, um semestre, estes estudos não foram inclusos no escopo deste trabalho.

2. Refinamento do MDJ-PC

O MDJ-PC pode ser refinado a partir do *feedback* de especialistas. Na elaboração do mesmo considerou-se uma visão superficial e abrangente das atividades dos principais papéis envolvidos com a produção de jogos para PC. Especialistas de cada um dos papéis podem, logo, ajudar a refinar o modelo.

3. Detalhamento do MDJ-PC

O nível de detalhamento do MDJ-PC foi até o nível de atividades. No entanto, pode-se aprofundar este detalhamento descrevendo, por exemplo, as ações contidas nas atividades e as entradas e saídas destas.

4. Ferramentas de suporte

No futuro pode-se vir a desenvolver, assim como o RUP, ferramentas específicas para apoiar a implantação do MDJ-PC em uma dada realidade de projeto.

5. Modelos para outros segmentos

Por fim, também se pode pensar em estudar modelos de desenvolvimento de jogos para segmentos que não sejam necessariamente o de PC. Por exemplo: modelos de desenvolvimento de jogos para dispositivos móveis.

Há inclusive, no presente momento, um estudo em andamento com o objetivo de propor o uso de uma metodologia para o desenvolvimento ágil de *advergames* [48].

6.3 Considerações Finais

Conclui-se aqui este trabalho de graduação com a realização dos quatro objetivos traçados no início do mesmo: analisar o mercado de jogos eletrônicos, identificar as características do desenvolvimento de jogos para *PC*, reconhecer as necessidades emergentes do desenvolvimento de jogos para *PC* e, finalmente, propor um modelo preditivo para este cenário.

No entanto, é importante frisar que este modelo é um esforço inicial e precisa ainda ser estudado, evoluído e instanciado para melhor se adequar à realidade do mercado.

Pode-se dizer, portanto, que a mensagem principal deste trabalho é: deve-se buscar a utilização de modelos de desenvolvimento de *software* adaptados à realidade do mercado de jogos com o objetivo de poder melhor estruturar, gerenciar e controlar o processo de criação.

Referências

- [1] PricewaterhouseCoopers' Global Entertainment and Media Outlook: 2006-2010.
- [2] Relatório SOFTEX. Tecnologias de Visualização na Indústria de Jogos Digitais. Disponível em: <http://observatorio.softex.br>. 2005.
- [3] ASSIS, D., MATIAS, A.. Game supera cinema como opção de entretenimento em 2003. Artigo da Folha de São Paulo. Disponível em: www1.folha.uol.com.br/folha/ilustrada/ult90u40114.shtml. 2003.
- [4] IGDA White Paper. IGDA Online Games White Paper. Disponível em: <http://www.igda.org/online>. 2003.
- [5] U.S. Census Bureau, Population Division. Disponível em: <http://www.census.gov/population/www/>. 2006.
- [6] REIMER, J.. Cross-platform Game Development and the Next Generation of Consoles. Disponível em: <http://arstechnica.com/articles/paedia/hardware/crossplatform.ars>. 2005.
- [7] Game Developer Magazine. Postmortem for Quantic Dream's fascinating Indigo Prophecy/Fahrenheit. Disponível em: http://gamasutra.com/features/20060620/cage_01.shtml. 2006.
- [8] DAGLOW, D.. Beyond Technology: Process Issues in Next Gen Development. Game Developers Conference Lecture. Disponível em: <https://www.cmpevents.com/sessions/GD/S1531i1.ppt>. 2006.
- [9] Wikipedia. List of software engineering topics. Disponível em: http://en.wikipedia.org/wiki/List_of_software_engineering_topics. 2006
- [10] Relatório Abragames. A indústria de Jogos Eletrônicos no Brasil. Disponível em: www.abragames.org/downloads.html. 2005.
- [11] DEMACHY, T.. Extreme Game Development: Right on Time, Every Time. Gamasutra Article. Disponível em: www.gamasutra.com/resource_guide/20030714/demachy_pfv.htm. 2003.
- [12] FLOOD, K.. Game Unified Process (GUP). Gamedev Article. Disponível em: <http://www.gamedev.net/reference/articles/article1940.asp>. 2003.
- [13] BOEHM, B., TURNER, R.. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley. 2004.
- [14] WALTON, G.. Bringing Engineering Discipline to Game Development. Gamasutra Article. Disponível em: http://www.gamasutra.com/features/19981218/walton_01.htm. 1998.
- [15] Game Developers Conference 2006 – All Sessions. Disponível em: <https://www.cmpevents.com/GD06/a.asp?option=C&V=1>. 2006.
- [16] Definição de Entretenimento. Dicionário Eletrônico Houaiss.
- [17] Declaração de MINSKY, M. na Game Developers Conference. 2004.
- [18] Definição de Game do Wikipedia. Disponível em <http://en.wikipedia.org/wiki/Game>. 2006
- [19] MALONE, F.. Toward a theory of intrinsically motivation instruction. Cognitive Science. 1981.
- [20] FALSTEIN, N.. Natural Funativity. Gamasutra Article. 2004.
- [21] BAKIE, R.T.. A Brief History of Video Games. Introduction to Game Development. 2005.
- [22] Definição de Adventure Game do Wikipedia. Disponível em http://en.wikipedia.org/wiki/Adventure_game. 2006.

- [23] Definição de Advergaming do Wikipedia. Disponível em: <http://en.wikipedia.org/wiki/Advergame>. 2006.
- [24] Playstation 3 vs. Xbox 360. Tech Head-to-head. Gamespot Article. Disponível em: <http://www.gamespot.com/features/6125087/index.html?type=tech>. 2005.
- [25] Interactive Digital Software Association – IDSA. Economic impacts of the demand for playing interactive entertainment software. 2001.
- [26] World of Warcraft continua em sua marcha de dominação global e alcança 1 milhão na Europa. Finalboss news. Disponível em: http://finalboss.uol.com.br/fb3/ctu.asp?cid=29561&add_backpack=1. 2006.
- [27] Global Wireless Gaming Report. Datamonitor. 2001.
- [28] SLOPER, T.. Game Production and Project Management. Introduction to Game Development. 2005.
- [29] SCHOBACK, K.. Game Industry Roles and Economics. Introduction to Game Development. 2005.
- [30] RYAN, T.. The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal. Gamasutra Article. 1999.
- [31] SLOPER, T.. Following Up After the Game is Released: It's not Over when it's Over. Game Design Perspectives. 2002.
- [32] OLSEN, J.. 2003 Game Development Salary Survey. Gamasutra Article. Disponível em: http://www.gamasutra.com/features/20040211/olsen_pfv.htm. 2004.
- [33] IRISH, D.. The Game Producer's Handbook. 2005.
- [34] Mapeamento de Competências da Cadeia Produtiva de Games: aplicação da metodologia DACUM pelo Unibratrec com as principais empresas de desenvolvimento de games do estado de Pernambuco. 2006.
- [35] McCONNELL, S.. Quality of Life Summit: The Business Case for Improved Production Practices. 2005.
- [36] Standish Group. Chaos Report. 1995.
- [37] SOARES, M.. Metodologias Ágeis: Extreme Programming e Scrum para o Desenvolvimento de Software. Disponível em: <http://www.inf.ufsc.br/resi/edicao04/artigo06.pdf>. 2004.
- [38] Definição da Wikipedia de Software Development Process. Disponível em: http://en.wikipedia.org/wiki/List_of_software_engineering_topics. 2006.
- [39] LUIZ, R.. Obtendo Qualidade de Software com o RUP. Disponível em: <http://www.javafree.org/content/view.jf?idContent=7>. 2005.
- [40] Definição de Waterfall do Wikipedia. Disponível em: http://en.wikipedia.org/wiki/Waterfall_model. 2006.
- [41] Definição de RUP do Wikipedia. Disponível em: http://en.wikipedia.org/wiki/Rational_Unified_Process. 2006.
- [42] Definição de Agile Software Development do Wikipedia. Disponível em: http://en.wikipedia.org/wiki/Agile_software_development. 2006.
- [43] Definição de Extreme Programming do Wikipedia. Disponível em: http://en.wikipedia.org/wiki/Extreme_Programming. 2006.
- [44] BECK, K., ANDRES, C.. Extreme Programming Explained: Embrace Change. Second Edition. 2004.
- [45] CHARNE, J.. Time is of the Essence. Famous Last Words – IGDA Column. Disponível em: http://www.igda.org/columns/lastwords/lastwords_Dec01.php. 2001.
- [46] KRUCHTEN, P.. The Rational Unified Process – An Introduction. Addison-Wesley. 1998.

- [47] Software Process Engineering Metamodel v1.1. Formal Specification. Disponível em: <http://www.omg.org/technology/documents/formal/spem.htm>. 2006.
- [48] ARAUJO, A.. Metodologia para Desenvolvimento Ágil de Advergates. Trabalho de Graduação 2006-1 do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE). 2006.