



Análise de Metodologias de Desenvolvimento de Software aplicadas ao Desenvolvimento de Jogos Eletrônicos

Trabalho de Graduação

Aluno: Raphael Lima Belém de Barros (rlbb@cin.ufpe.br)

Orientador: Prof. Alexandre Marcos Lins de Vasconcelos (amlv@cin.ufpe.br)

Co-orientador: Prof. Geber Lisboa Ramalho (glr@cin.ufpe.br)

Recife, 22 de agosto de 2007

Assinaturas

Este Trabalho de Graduação é resultado dos esforços do aluno Raphael Lima Belém de Barros, sob a orientação dos professores Alexandre Marcos Lins de Vasconcelos e Geber Lisboa Ramalho, sob o título de “Análise de Metodologias de Desenvolvimento de Software aplicadas ao Desenvolvimento de Jogos Eletrônicos”. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

Alexandre Marcos Lins de Vasconcelos (Orientador)

Geber Lisboa Ramalho (Co-orientador)

Raphael Lima Belém de Barros (Autor)

Dedico este trabalho a toda a minha família. Em especial aos meus pais, Volney e Elmira por todo o amor, atenção e carinho incondicionais. E a Luciana pelo amor, amizade e companheirismo. Agora e sempre.

Resumo

O desenvolvimento de jogos eletrônicos tornou-se uma atividade lucrativa e bem-sucedida. Enquanto os números alcançados provocam otimismo, muitas empresas, principalmente as de pequeno e médio porte ainda encontram dificuldades em produzir jogos através de projetos bem-sucedidos. Isto é, projetos que respeitem restrições de escopo, cronograma e orçamento. Este Trabalho de Graduação propõe uma Análise de Metodologias de desenvolvimento de jogos existentes na Literatura e utilizadas por Empresas Locais. Como resultado desta Análise, será proposto um “Manual de Boas Práticas para Desenvolvimento de Jogos”, que poderá ser utilizado por empresas de desenvolvimento de jogos, principalmente as iniciantes.

Palavras-chave: Engenharia de Software, Metodologias de Desenvolvimento de Software, Entretenimento Digital e Jogos eletrônicos.

Abstract

The development of video games became a lucrative and successful activity. While the reached numbers provoke optimism, a lot of companies, mainly small and medium companies still have difficulties in producing games through successful projects. That is, projects on scope, schedule and budget constraints. This Work of Graduation proposes an Analysis of Development Models for games, existents in the Literature and used by Local Companies. A “Good Practices Manual for Development” will be proposed as a result of this Analysis. This Manual can be used by Development Game Companies, Specially the Rookies.

Key words: Software Engineering, Software Development Models, digital entertainment, video games

Agradecimentos

Primeiramente, agradeço a Deus por tudo que me foi dado. Sem Ele nada somos.

Agradeço a meus pais, Volney e Elmira, por todo amor e apoio oferecidos durante toda a minha vida, em especial durante esse tempo de minha graduação. Agradeço também a minha irmã, Juliana, por todo o carinho e companheirismo e a toda a minha família, avós, tios e primos.

Agradeço a minha namorada Luciana, por todo respeito, amor e compreensão. Obrigado por me animar nos momentos de fraqueza.

Agradeço a Alexandre Vasconcelos e Geber, meus orientadores, pelos ensinamentos passados durante a graduação. Estendo esses agradecimentos a todos os professores do Centro de Informática.

Agradeço a João Paulo, Domingos, Ramonzinho, Roberto, Mozart, Cesar, Diego, Shirley, Grasi, Chico, Antonio, Andrea, Ernani e muitos outros verdadeiros amigos que formei durante a Graduação. Além desses, agradeço a todos os outros companheiros de turma, pelas alegrias e dificuldades compartilhadas.

Peço perdão a todos os outros amigos que não foram citados aqui, por esquecimento de minha parte.

A todos vocês, um muito obrigado!

Sumário

1. Introdução	10
1.1. Objetivos	10
1.2. Estrutura do Trabalho	11
2. Jogos Eletrônicos	12
2.1. Evolução	12
2.2. Mercado de Jogos.....	17
3. Projetos de TI.....	20
3.1. Cenário	20
4. Características Gerais do Desenvolvimento de Jogos.....	23
4.1. Ciclo de Vida de Projeto de Jogos	23
4.1.1. Concepção.....	24
4.1.2. Pré-Produção.....	24
4.1.3. Produção	25
4.1.4. Pós-Produção	25
4.1.5. Pós-Lançamento.....	26
4.2. Papéis envolvidos	27
4.2.1. Produtor	27
4.2.2. Artista.....	27
4.2.3. Game Designer.....	28
4.2.4. Programador	28
4.2.5. Engenheiro de Áudio	29
4.2.6. Engenheiro de Qualidade	29
5. Metodologias de Desenvolvimento de Jogos	30
5.1. Game Waterfall Process	30
5.2. Extreme Game Development.....	31
5.3. Game Unified Process	34
5.4. Scrum.....	35
5.5. Análise Comparativa	38
6. Análise de Uso na Prática	41
6.1. Introdução	41
6.2. As Questões	41
7. Consolidação da Análise.....	50
7.1. Análise de Presença	50
7.2. Resultado da Análise	51
7.2.1. Artefatos Sugeridos	51
7.2.2. Práticas sugeridas	52
7.2.3. Papéis Sugeridos.....	55
7.3. Pontos em divergência.....	56
8. Conclusão	58
8.1. Conclusões	58
8.2. Trabalhos Futuros.....	60
Referências	62
Apêndice A – Glossário	65
Apêndice B – <i>Rational Unified Process</i>	66

Índice de Figuras

Figura 1. Tennis for Two	13
Figura 2. Space War!	13
Figura 3. Odyssey	14
Figura 4. Super Mario Bros.	15
Figura 5. <i>Playstation</i>	16
Figura 6. Sega <i>Saturn</i>	16
Figura 7. Nintendo 64.....	16
Figura 8. Nintendo DS.....	17
Figura 9. PSP.....	17
Figura 10. Tipos de Jogos mais vendidos para consoles.....	18
Figura 11. Tipos de Jogos mais vendidos para computadores.	18
Figura 12. Crescimento do Montante de Vendas (em US\$) nos USA.....	19
Figura 13. Chaos Report 94, projetos quanto a sua Resolução.....	21
Figura 14. Extreme Chaos 2001, projetos quanto a sua Resolução.	22
Figura 15. Fases do Ciclo de Vida e Artefatos gerados.	23
Figura 16. Modelo 3D criado no 3D Studio Max™	28
Figura 17. Engine Torque Game Builder, da Garage Games.	29
Figura 18. Fases do Modelo Cascata Tradicional e Adaptado para Jogos	30
Figura 19. Práticas do Extreme Programming.....	33
Figura 20. Visão Geral do Scrum.	38
Figura 21. Gráfico de utilização de artefatos nas empresas	42
Figura 22. Gráfico da presença de papéis nas empresas	42
Figura 23. Versões de jogos.....	43
Figura 24. Uso de Publicadoras	43
Figura 25. Presença dos Patrocinadores	44
Figura 26. Tamanho das Equipes	44
Figura 27. Número de Projetos Iterativos.....	45
Figura 28. Duração dos ciclos de projetos iterativos	45
Figura 29. Ocorrência de reuniões de retrospectivas de iterações	46
Figura 30. Ocorrência de Stand-up Meetings.....	46
Figura 31. Utilização de profissionais com funções especializadas.	47
Figura 32. Presença de equipes auto-organizadas	47
Figura 33. Momento de realização de testes no projeto.....	48
Figura 34. Utilização de UML na modelagem de projetos	48
Figura 35. Uso de software de controle de versão	49
Figura 36. Ferramentas de auxílio à comunicação.....	49
Figura 37. Arquitetura do RUP	67

Índice de Tabelas

Tabela 1. Visão Geral do Documento	11
Tabela 2. Visão Geral do Documento (Continuação)	12
Tabela 3. Princípios do Extreme Game Development.....	32
Tabela 4. Práticas do <i>Extreme Game Development</i>	33
Tabela 5. Práticas do <i>Extreme Game Development</i> (continuação).....	34
Tabela 6. Artefatos do Scrum	36
Tabela 7. Quadro Comparativo – Metodologias de desenvolvimento	39
Tabela 8. Presença de práticas nas metodologias estudadas	50
Tabela 9. Presença de práticas nas empresas pesquisadas	51
Tabela 10. Quadro-resumo das boas práticas selecionadas.....	54
Tabela 11. Quadro-resumo dos papéis identificados.	56
Tabela 12. Quadro-resumo dos pontos divergentes.	58
Tabela 13. Disciplinas do RUP	68
Tabela 14. Fases do RUP.	69

1. Introdução

Na década de 70, o desenvolvimento de software não era capaz de acompanhar o crescimento da demanda e a complexidade dos problemas a serem resolvidos por este meio [WIKI – III]. Surgia então, como reação a este acontecimento, a chamada Engenharia de Software, que tinha por objetivo estabelecer métodos, técnicas e métricas para auxiliar o desenvolvimento e validação de sistemas.

Hoje, a Engenharia de Software se encontra estabelecida em diversas áreas. Sejam elas mais tradicionais ou recentes, cujo domínio é pouco conhecido ou altamente mutável. É neste último contexto que se insere o desenvolvimento de jogos eletrônicos.

Apesar de possuir diversas características inerentes ao desenvolvimento de softwares e sistemas habituais, o desenvolvimento de jogos possui características singulares, a começar pela equipe multidisciplinar que desenvolve projetos, composta, em sua maioria, por artistas e não apenas projetistas de software e programadores.

Para atender as necessidades particulares do desenvolvimento dessa categoria de software, algumas metodologias criadas especialmente para o domínio, ou adaptada de outras já estabelecidas [FLOOD], são utilizadas por empresas especializadas em jogos ao redor do mundo.

1.1. *Objetivos*

O trabalho de graduação a ser desenvolvido tem por finalidade realizar uma análise das metodologias de desenvolvimento de jogos eletrônicos utilizadas por empresas pernambucanas e estrangeiras.

Em um primeiro momento, serão realizadas pesquisas bibliográficas sobre as metodologias mais utilizadas pelas empresas que desenvolvem jogos fora do País. Abordando suas diversas características, mas mantendo o foco nas principais práticas realizadas em cada metodologia.

Em seguida, após coletadas as principais características das metodologias utilizadas por empresas a nível mundial, será feita uma verificação da adesão a essas metodologias por parte de algumas empresas presentes no Porto Digital.

Para isso, será aplicado um questionário baseado nos dados coletados na pesquisa bibliográfica às empresas de jogos situadas no Porto Digital.

Por fim, as informações coletadas nas pesquisas bibliográficas e os resultados dos questionários aplicados serão confrontados, gerando assim um “Manual de Boas Práticas” para o desenvolvimento de jogos eletrônicos.

1.2. Estrutura do Trabalho

As tabela 1 e 2 abaixo apresentam uma rápida descrição dos capítulos deste documento:

Capítulos	Nomes	Descrições
1	Introdução	Introduz ao problema e ao documento proposto, apresentando seus objetivos e uma síntese de suas seções.
2	Jogos Eletrônicos	Apresenta uma introdução histórica aos jogos eletrônicos, do início ainda em meados do século 20 até os dias atuais. Ainda neste mesmo capítulo são apresentados números que comprovam a importância e relevância econômica dos jogos eletrônicos.
3	Projetos de TI	Este capítulo ilustra a importância da adoção de metodologias no desenvolvimento de softwares de uma maneira geral, comprovando inclusive, através de números da indústria de software.
4	Características Gerais do Desenvolvimento de Jogos	Como o próprio nome sugere, neste capítulo são apresentadas características gerais de projeto de desenvolvimento de jogos. Nele são apresentados papéis, fases e artefatos citados em literatura.

Tabela 1. Visão Geral do Documento

Capítulos	Nomes	Descrições
5	Metodologias de Desenvolvimento de Jogos	Aqui serão realizadas análises sobre algumas metodologias de desenvolvimento de jogos encontradas em bibliografia.
6	Estudo de Caso	Após terem sido mapeadas, no capítulo 5, é hora de verificar se os papéis, práticas e artefatos sugeridos na literatura correspondem ao praticado nas empresas do cenário local de Recife.
7	Consolidação da Análise	Capítulo que sintetiza o estudo realizado nos 2 capítulos anteriores. Neste capítulo são sugeridos os papéis, práticas e artefatos que devem compor o projeto de desenvolvimento de um jogo.
8	Conclusão	Conclusões do trabalho e possibilidades de trabalhos futuros.

Tabela 2. Visão Geral do Documento (Continuação)

2. Jogos Eletrônicos

Este capítulo apresenta, dentro de um contexto histórico, o surgimento e a evolução dos jogos eletrônicos até os dias atuais. Além disso, serão apresentadas algumas análises mercadológicas sobre esta forma de entretenimento. Todas essas informações deverão servir para uma contextualização da área na qual este trabalho se concentra. Mais ainda para, auxiliar na ratificação da relevância deste trabalho.

2.1. Evolução

Em meados do século passado, a iniciativa militar era a principal, e talvez única responsável pela evolução dos computadores, como conhecemos hoje. Foi nesse contexto em que surgiu o primeiro jogo eletrônico de que se tem conhecimento. Tratava-se do *Tennis for Two*, criado no laboratório de

pesquisas militares *Brookhaven National Laboratory*, nos Estados Unidos. O jogo era uma simplificação do tênis e era apresentado em uma tela de Osciloscópio. [WIKI – I]

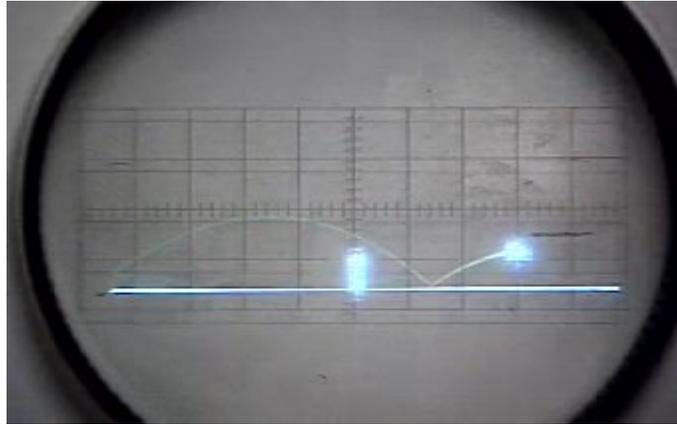


Figura 1. Tennis for Two

Entre 1961 e 1962, um grupo de estudantes do MIT¹, liderados por Steve Russel, desenvolveu um jogo chamado de *Space War!*. Tratava-se de um jogo que simulava uma batalha de naves, no espaço. O jogo foi desenvolvido no DEC PDP-1, um computador que custava milhões de dólares na época. Embora o esforço despendido e o custo tenham sido grandes, seus criadores ainda não viam o jogo como uma oportunidade de negócios.



Figura 2. Space War!

¹ Massachusetts Institute of Technology.

Com os avanços tecnológicos evoluindo em grande velocidade, o Engenheiro Ralph Baer da *Sanders Associates* começa a criar um jogo que pudesse sair dos laboratórios e ser jogado em lares comuns. Trabalhando na idéia desde a década de 50, apenas em 1971, Ralph consegue firmar um acordo com a Magnavox, que começa então a desenvolver o primeiro vídeo game a ser conectado em um televisor, o *Odyssey*.



Figura 3. Odyssey

Enquanto isso, Nolan Bushnell, percebendo que *Main Frames* eram muito caros e grandes, começa a adaptar o jogo *Space War!* de Russel, criando um computador dedicado chamado *Computer Space*. Com o sucesso da idéia, Bushnell começa a comercializar o que se considera hoje como o primeiro arcade da história. Mais tarde Bushnell seria o fundador da Atari. [WIKI – I]

Já na Atari, Bushnell lança o *Pong* e posteriormente o *Tank*, títulos de bastante sucesso na época. E ainda na década de 70, a Atari viria a ser uma das maiores empresas de jogos eletrônicos de todos os tempos, principalmente após a segunda metade, quando os consoles domésticos começaram a se tornar populares, com a chegada do Atari2600 ao mercado.

Entre final da década de 70 e os primeiros anos da década seguinte, outras empresas passaram a também produzir jogos para Arcades e para o Atari2600. Foi nessa época que surgiram clássicos como o *Space Invaders*, *Pac-man* e *Donkey-Kong*, apresentando ao mundo empresas como a Midway, Namco, Sega, SNK e a Nintendo, que viriam ter grande sucesso nas décadas posteriores.

Do outro lado do mundo, no Japão, começava a surgir o Nes², console que transformaria a Nintendo na gigante mundial de jogos. O vídeo-game era de 8 bits, o que dava mais poderes aos desenvolvedores de jogos. Nesse período, outros clássicos surgiram. Dentre eles “Mario Bros.”, que viria a ser o mascote oficial da Nintendo. A empresa continuou a crescer, lançando ótimos títulos, o que fez com que a concorrência fosse encurralada. Outro console o *Master System* da Sega, também obteve sucesso nessa época, porém suas vendas ainda estavam longe dos números alcançados pelo Nes.



Figura 4. Super Mario Bros.

No início da década de 90, o mercado estava segmentado em jogos para console, computadores e portáteis (destaque para o Game Boy da Nintendo). Enquanto isso, as duas principais concorrentes do mercado de consoles naquele momento, a Nintendo e a Sega lançam seus consoles de 16 bits: o SNES e o Mega Drive, respectivamente. A Nintendo mantém a liderança, mesmo com a criação do mascote da Sega, o “*Sonic*”, que foi um sucesso de vendas. Foi quando, em 1993, o primeiro console de 32 bits foi criado. Tratava-se do 3DO, da Panasonic. O console não foi bem sucedido, porém outra gigante do mercado de eletrônicos planejava entrar no mercado.

Eis que a Sony entra no mercado de games no final de 94 com o seu Playstation, vídeo game de 32 bits que fazia frente ao *Saturn*, também de 32 bits, lançado no mesmo ano. Foi nesse momento que foi inaugurada a era dos jogos 3D. Neste momento, a concorrência restringia-se a 3 consoles: o *Playstation* da Sony, o Sega *Saturn* e o Nintendo 64. Nessa guerra a Sony

² Nintendo Entertainment System.

levou a melhor, principalmente pelo apoio de grandes *Softhouses*³ e pelo fato da Nintendo ter apostado no antigo formato de cartuchos, enquanto seus concorrentes utilizavam os cds como mídia.



Figura 5. Playstation



Figura 6. Sega Saturn



Figura 7. Nintendo 64

Já no início do século XXI, chegava ao mercado a 6ª geração de consoles, os de 128 bits. O primeiro a ser lançado foi o Sega *Dreamcast*, em 2000. Ele obteve bons resultados até que a Sony anunciou o lançamento do seu novo console: o *Playstation 2*.

No ano seguinte, a Sega tomou uma importante decisão: abandonou o mercado de consoles para se dedicar apenas ao desenvolvimento de jogos para as plataformas existentes. A Sony havia lançado o seu Playstation 2 no final do ano anterior, continuando a saga de sucesso do seu antecessor. A Nintendo lança o seu Game Cube, que por medo de pirataria, foi lançado o Mini-DVD, diferente do *playstation 2* que utilizava o DVD comum. Ainda nesse ano, a maior empresa de softwares do mundo entra no mercado de games. A Microsoft apresenta o seu XBox, também compatível com os DVD e possuindo também um Disco Rígido (HD), que permitia que jogos e músicas fossem salvos. [WIKI – I]

Hoje, o mercado de consoles está em sua 7ª geração. Lançado em 2005, o XBox 360 da Microsoft inaugura a era. No ano seguinte, o Wii da Nintendo inaugura uma nova forma de interação em jogos, fazendo com que os movimentos dos usuários possam ser detectados pelo jogo. A Sony também lança seu novo console, o Playstation 3, que amarga o último lugar em vendas, principalmente pelo alto preço.

Com o crescimento exponencial da Internet, as fabricantes de jogos apostam também na interatividade entre os jogadores, permitindo que jogadores joguem simultaneamente e de forma cooperativa. São os chamados

³ Empresas que desenvolvem jogos.

Jogos Massivos para Múltiplos jogadores (*Mutiplayer Massive Online Role-Playing Games*). O jogo de maior destaque desse segmento é o “World of Warcraft”.

O terceiro segmento do mercado de jogos, o de consoles portáteis, também segue em crescimento. Aqui a disputa se dá entre o Nintendo DS e PSP da Sony. Até o momento, o console da Nintendo vem vencendo facilmente essa disputa, dentre outras coisas, devido ao baixo preço e a função *Touch Screen*, que permite novas formas de interação entre jogos e jogadores.



Figura 8. Nintendo DS



Figura 9. PSP

Como é possível observar, o Mercado de jogos encontra expansão e cada vez mais, jogadores das mais variadas faixas etárias tem aderido ao seguimento. A próxima seção apresentará alguns números que apontam a relevância deste lucrativo negócio

2.2. Mercado de Jogos

Como dito anteriormente, o mercado de jogos encontra-se segmentado basicamente e 3 categorias: consoles, computadores, portáteis. E, apesar das altas taxas de crescimento nas duas últimas categorias, o mercado de consoles ainda é o mais lucrativo.

Anualmente, é publicada uma pesquisa feita pela *Entertainment Software Association* (ESA), associação composta pela maioria das gigantes do setor, que busca identificar os mais relevantes aspectos com relação ao desenvolvimento de software voltado para o entretenimento. Na edição de

“The video game industry is entering a new era, an era where technology and creativity will fuse to produce some of the most stunning entertainment of the 21st Century. Decades from now, cultural historians will look back at this time and say it is when the definition of entertainment changed forever.”
– Douglas Lowenstein, President, Entertainment Software Association

2006 do ESA Facts foram apresentadas algumas características acerca do Mercado de Jogos Eletrônicos. [ESA06]

Nessa pesquisa, dentre outros, foram identificados os estilos de jogos mais vendidos para consoles e computadores.

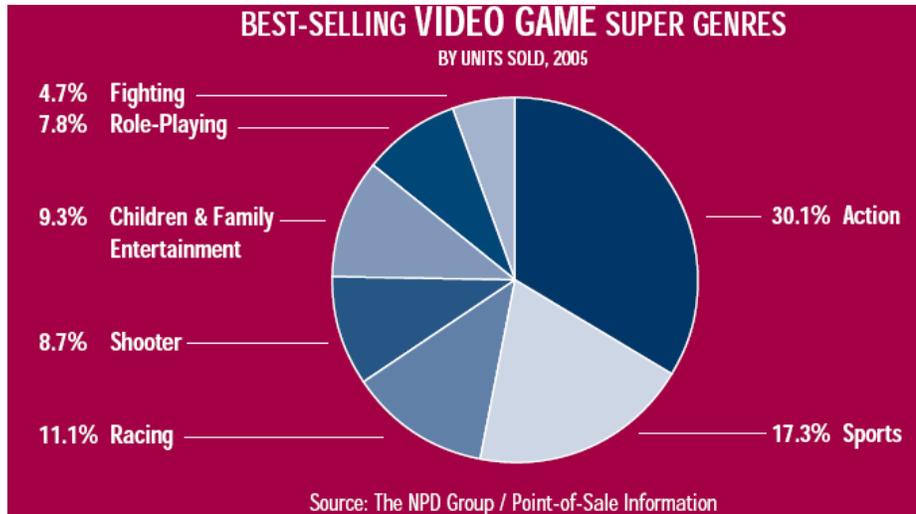


Figura 10. Tipos de Jogos mais vendidos para consoles

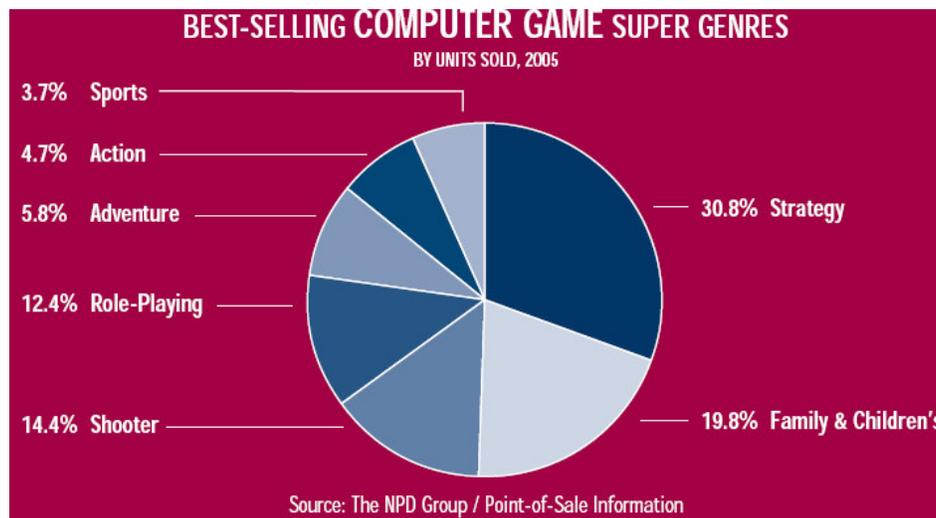


Figura 11. Tipos de Jogos mais vendidos para computadores.

Ainda na mesma pesquisa, foram apresentados resultados do montante de vendas de jogos nos últimos anos, para consoles e computadores, apenas nos Estados Unidos.

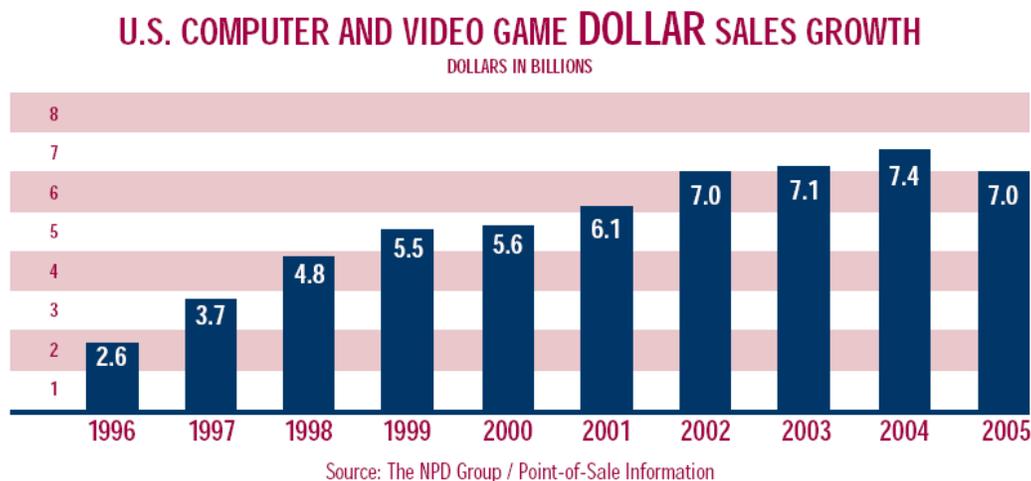


Figura 12. Crescimento do Montante de Vendas (em US\$) nos USA.

Esses e outros números demonstram a relevância de um mercado que faturado mais que o de cinema há vários anos seguidos, nos Estados Unidos. No Brasil, o mercado segue a mesma tendência de crescimento.

Em 1995, foi criada a primeira empresa brasileira dedicada exclusivamente a jogos: a *Perceptum*. 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 [SOFTEX].

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.

Os dados apresentados neste capítulo só reforçam a importância no que diz respeito à geração de negócios, do segmento de jogos. Apesar dos números serem impressionantes, as empresas de jogos, principalmente as pequenas e médias ainda cometem diversos de execução em seus projetos. A maioria destes erros é conhecida, inerente do desenvolvimento de *softwares*..

3. Projetos de TI

Um projeto de software pode ser comparado ao projeto de uma ponte. Partindo-se da premissa que o projeto de uma ponte é sempre entregue dentro do cronograma e orçamento previstos, portanto é bem sucedido. [CHAOS04] Por que o mesmo não acontece aos projetos de Software?

Uma das principais razões para isso é o tempo gasto na fase de planejamento do projeto. Mas a maior de todas são os 3000 anos de experiência na construção civil, contra cerca de 50 anos do desenvolvimento de softwares. Além disso, quando uma ponte cai, as causas da falha ocorrida são investigadas. Enquanto que, em software, quando um erro ocorre, nem sempre suas causas são investigadas. Fazendo com que os mesmos erros sejam cometidos outrora. [CHAOS04]

3.1. Cenário

Segundo Friedrich Ludwig Bauer, "Engenharia de software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais⁴".

A Engenharia de Software pode ser vista como a aplicação de tecnologias e práticas multidisciplinares voltadas para a especificação, desenvolvimento e manutenção de sistemas e softwares. [WIKI –II]

Essas tecnologias e práticas vêm se desenvolvendo pela dificuldade de atingir o sucesso em projetos de software. As causas para tamanho estorvo vão desde a própria imaturidade da Engenharia de Software como ciência até a própria natureza dinâmica do domínio.

Observando essas dificuldades encontradas no desenvolvimento de projetos de software bem-sucedidos, o *Standish Group International*⁵ desenvolve, desde 1994, um relatório sobre projetos de software. Apontando, dentre outras coisas, a taxa de projetos bem-sucedidos e as principais causas para o sucesso destes projetos.

⁴ Retirado de: http://pt.wikipedia.org/wiki/Engenharia_Software

⁵ <http://www.standishgroup.com>

Um dos dados mais relevantes deste relatório foi a determinação do *status* dos projetos de software, em empresas Norte-Americanas. Esses projetos analisados foram divididos em 3 categorias:

- I. **Bem-sucedidos:** Projetos desenvolvidos com as funcionalidades planejadas e entregues dentro do orçamento e prazos previstos;
- II. **Comprometidos:** Projetos que desrespeitaram o escopo, cronograma ou orçamento do projeto;
- III. **Fracassados:** Projetos que foram cancelados em algum ponto de seu ciclo de desenvolvimento.

No primeiro relatório, publicado em 1994, foram analisados cerca de 175 mil projetos e os resultados encontrados foram os seguintes:

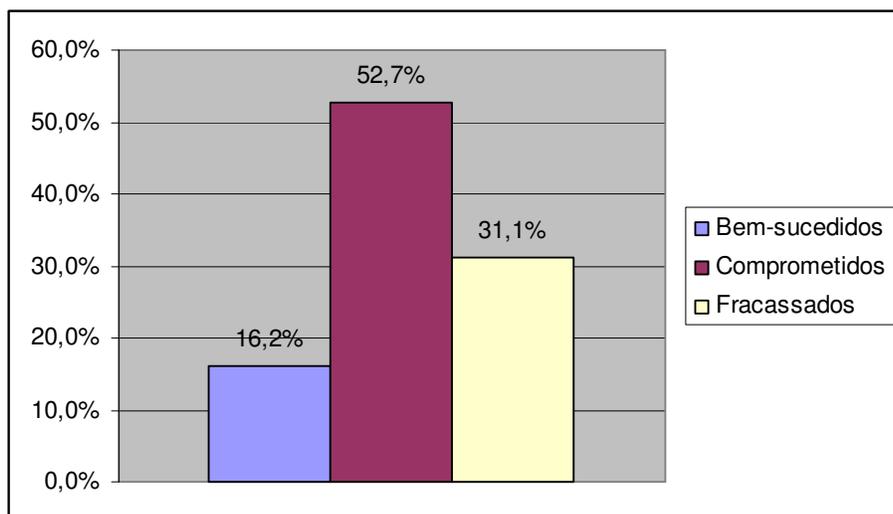


Figura 13. Chaos Report 94, projetos quanto a sua Resolução.

Em 2001, em seu novo relatório, o *Extreme Chaos*, o *Standish Group* apresentou novos valores para os cerca de 280 mil projetos analisados:

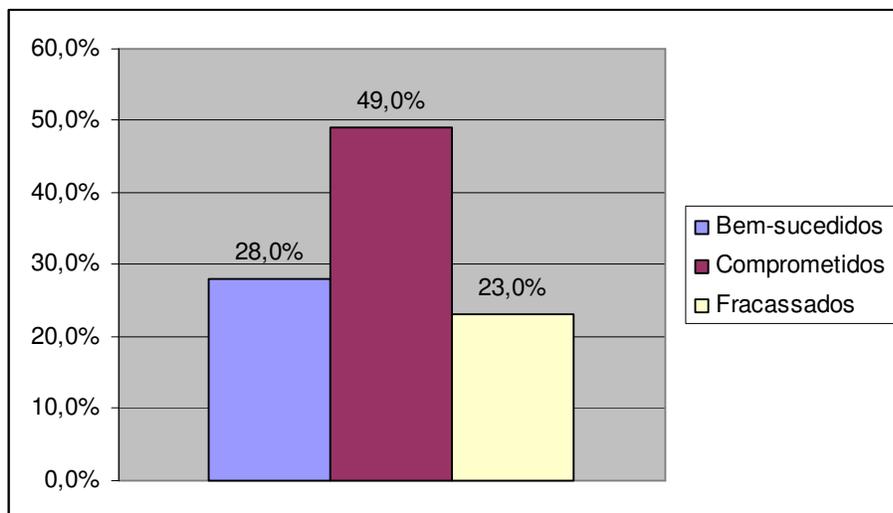


Figura 14. Extreme Chaos 2001, projetos quanto a sua Resolução.

Analisando os dados apresentados é possível observar que houve uma melhora no percentual de projetos bem-sucedidos, mesmo que o percentual de projetos que, de alguma forma, falham em alguma das dimensões planejadas ainda seja alta (72%).

Esse comportamento se deu pela evolução de metodologias e ferramentas de auxílio ao desenvolvimento, pela diminuição da complexidade dos projetos, dentre outros. É importante observar também que, alguns desses projetos podem “mascarar” os reais resultados apresentados. Na pesquisa realizada para o relatório de 2001, alguns gerentes de TI entrevistados afirmaram que, após realizar sua melhor estimativa para o projeto, multiplicavam o resultado por 2 e somavam mais $\frac{1}{2}$ da estimativa. [EXTREME]

A realidade apresenta dados alarmantes em relação ao desenvolvimento de *softwares* e o desperdício de recursos em projetos fracassados. Desde que foram observados, ainda no século passado, diversos problemas que podem ocorrer no desenvolvimento de um software, tem se investido, dentre outras coisas, em metodologias e/ou processos específicos de desenvolvimento.

As empresas de jogos também caminham para o amadurecimento, neste sentido. Investindo em criação e/ou adaptação de metodologias específicas para projetos de desenvolvimento de jogos eletrônicos. Antes de fazer uma análise de algumas metodologias para jogo, é preciso entender melhor quais as características gerais de projetos desta categoria de *software*.

4. Características Gerais do Desenvolvimento de Jogos

Os dados apresentados no capítulo anterior demonstram a necessidade de definição e evolução de metodologias e ferramentas que auxiliem de alguma forma o correto desenvolvimento de projetos de software. Tratando especificamente do desenvolvimento de jogos, os problemas são potencializados devido à extrema dinamicidade dos fatores de influência e a multi-disciplinaridade do domínio.

Neste capítulo será apresentado o ciclo de vida geral de um projeto de desenvolvimento de jogos e as principais abordagens utilizadas no desenvolvimento de jogos ao redor do mundo. Analisando, inclusive, os pontos fracos e fortes de cada uma.

4.1. Ciclo de Vida de Projeto de Jogos

Durante o processo de produção de um jogo, existem 5 fases principais que compõem o seu ciclo de vida: Concepção, Pré-produção, Produção, Pós-Produção, Pós-Lançamento. [SLOPER]

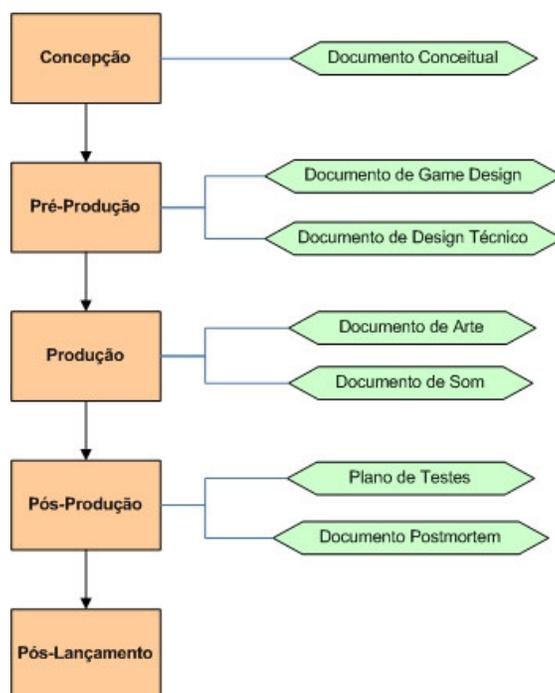


Figura 15. Fases do Ciclo de Vida e Artefatos gerados.

4.1.1. Concepção

Na concepção, após decidir pelo desenvolvimento de um jogo, a empresa que planeja desenvolvê-lo realiza um estudo para atestar a viabilidade operacional, técnica, econômica e de cronograma para o projeto. O principal objetivo desta fase é, baseado no conhecimento prévio da empresa desenvolvedora e características mercadológicas, definir o conceito de um jogo que seja Mercadologicamente bem-sucedido. [SLOPER]

As decisões tomadas nesta fase são apresentadas no chamado Documento Conceitual (*Conceptual Document*). Após o Documento Conceitual ter sido aceito por todas as partes envolvidas é o momento de iniciar a fase de produção propriamente dita. Esta fase costuma ser dividida em 3 outras: Pré-Produção, Produção e Pós-Produção.

4.1.2. Pré-Produção

Nesta fase são definidos os elementos necessários para o desenvolvimento do jogo. É aqui que a equipe de desenvolvimento é alocada, que termos de confidencialidade e não-divulgação são assinados e são desenvolvidos 2 dos principais artefatos necessários para o desenvolvimento do jogo: o Documento de Game Design (Game Design Document) e o Documento de Design Técnico (Technical Design Document).

No documento de Game Design são apresentadas as características principais do jogo, como por exemplo, detalhes da dinâmica e características áudio-visuais do jogo. Além disso, é no Game Design que é definido um plano de desenvolvimento para o jogo. No documento de Design Técnico são especificados detalhes técnicos das idéias apresentadas do documento de Game Design. Essa especificação técnica deve seguir como guia para a atividade de programação do jogo, identificar os desafios e dificuldades que podem ser encontradas e as ferramentas de desenvolvimento a serem utilizadas. Por fim, estas informações presentes nos 2 documentos citados devem servir como base para a definição das atividades do processo de produção do jogo.

4.1.3. Produção

É nesta fase em que ocorre o desenvolvimento do código-fonte, e da arte áudio-visual do jogo. É nesta fase também que são desenvolvidos e constantemente atualizados, dois artefatos do projeto do jogo: o Documento de Arte (Art List Document) e o Documento de Som (Sound List Document). [SLOPER]

O Documento de Arte é o responsável por apresentar à equipe de arte, detalhes sobre cenários, personagens e objetos que eles devem criar. O Documento de Som é o equivalente para equipe de som do jogo. Ele apresenta detalhes sobre músicas e efeitos sonoros do jogo. É importante observar que ambos os documentos devem ser criados em total concordância com o Documento de Game Design do projeto.

4.1.4. Pós-Produção

Após finalizados os elementos do jogo são verificados nessa fase possíveis falhas existentes. A equipe de testes verifica a existência de bugs, repassando-os para a equipe de desenvolvimento corrigi-los. Ao mesmo tempo o Departamento de Marketing da empresa e a Empresa Publicadora (*Publisher*) trabalham na campanha de promoção para o lançamento do jogo.

É nesta fase que são criados: o Plano de Testes e um documento para *feedback* do projeto, chamado de *Post mortem*. O primeiro especifica, baseado no documento de Game Design e de Design Técnico, o que deve ser testado e que abordagem será dada aos testes. É nesta fase que as versões de teste do jogo são divididas em 2 categorias: Alpha e Beta. No Alpha, os testes envolvem basicamente a equipe de desenvolvimento, enquanto o Beta envolve também os consumidores finais.

O Documento de *Post mortem* apresenta, ao final do projeto, os pontos positivos, negativos e lições aprendidas durante o projeto. Esse documento deve servir para que, em projetos posteriores, o número erros seja minimizado e o de acertos maximizado.

4.1.5. Pós-Lançamento

Nesta fase são monitoradas as reações do público ao jogo, já presente no mercado. As ações realizadas nesta fase podem ser divididas em 5 grupos [CARVALHO]:

- 1. Suporte às Vendas:** O varejista pode ter questionamentos a cerca do jogo oriundas dos consumidores. Alguns destes questionamentos podem voltar para a Empresa 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 Empresa 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.

4.2. Papéis envolvidos

Na seção anterior foram apresentadas as principais etapas que compõem o ciclo de vida do desenvolvimento de um jogo. Após uma rápida descrição das fases que compõem esse ciclo, ficou clara a heterogeneidade de uma equipe de desenvolvimento desta categoria. Assim, aqui serão apresentados os principais papéis, no total de 6, assumidos por colaboradores durante o desenvolvimento de um jogo eletrônico.[OLSEN]

4.2.1. Produtor

O produtor tem um papel semelhante a um gerente de projetos. Ele é o responsável por planejar e acompanhar o desenvolvimento do projeto, envolvendo-se em todas as áreas e fases do projeto. Ou seja, o Produtor é o responsável direto pelo sucesso do projeto. Neste caso, entende-se por um projeto bem-sucedido aquele que termina dentro do escopo previsto, respeitando restrições de custo e de cronograma, satisfazendo ao fim, seu cliente. [PMBOK]

Por ter responsabilidades semelhantes às de um gerente de projetos, o produtor costuma ser o responsável por alocar recursos, estabelecer o cronograma do projeto, garantir a comunicação da equipe de desenvolvimento, etc. O Produtor participa ativamente de todas as fases do projeto, principalmente na fase de Concepção, onde atua ativamente na produção do Documento Conceitual (*Conceptual Document*).

4.2.2. Artista

O artista é o responsável por produzir o conteúdo de arte 2D e 3D do jogo. Logo, dependendo da especialidade do artista dentro da equipe de desenvolvimento, ele pode desempenhar atividades como a concepção de personagens, modelagem 3D, desenho de texturas, etc. Representando a maioria de uma equipe de desenvolvimento de jogos, os Artistas são os responsáveis por produzir material artístico em adequação ao Documento de Game Design, proposto na fase de Pré-Produção. [CARVALHO]

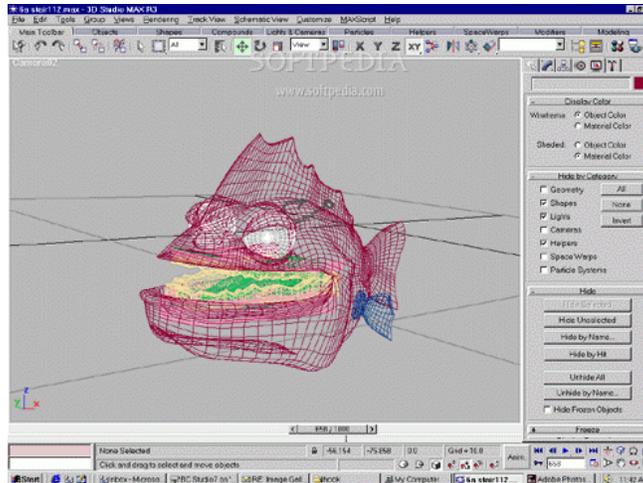


Figura 16. Modelo 3D criado no 3D Studio Max™

4.2.3. Game Designer

O Game Designer é o responsável por descrever a história, os elementos, características e fluxos do jogo a ser desenvolvido. Essa atividade requer criatividade e um grande conhecimento dos jogos existentes. É também responsabilidade do Game Designer fazer com que todos os elementos do jogo sejam dispostos de forma coerente. Todos esses elementos criados pelo Game Designer são apresentados do Documento de Game Design.

4.2.4. Programador

O programador é o responsável por codificar os elementos responsáveis pela execução do jogo. Diversas atividades de subáreas distintas podem ser atribuídas aos programadores dentro de um jogo: codificação da parte gráfica do jogo, codificação da dinâmica/física do jogo, codificação do sistema de som do jogo, etc. Atualmente, as atividades dos programadores tem sido facilitada pelo uso de *Engines* e *Frameworks*.

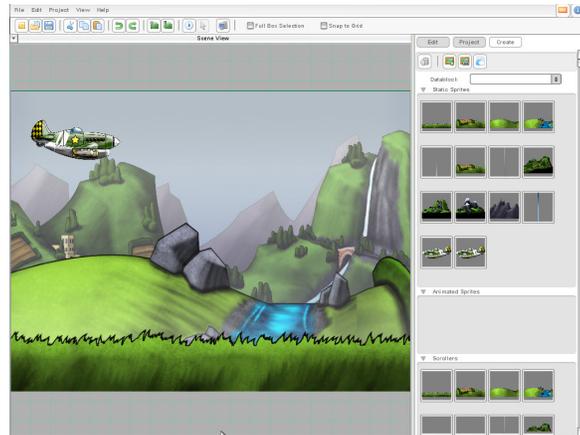


Figura 17. Engine Torque Game Builder, da Garage Games.

4.2.5. Engenheiro de Áudio

O Engenheiro de Áudio é o responsável por criar as trilha sonora e os efeitos sonoros do jogo. Atuando juntamente com a equipe de artistas, o Engenheiro de Som é responsável por dar identidade áudio-visual ao jogo, baseando-se no conteúdo descrito no documento de Game Design.

4.2.6. Engenheiro de Qualidade

O Engenheiro de Qualidade é o responsável por garantir que o jogo produzido ao final de um projeto possua o nível de qualidade exigido pelo Mercado. Ele realiza atividades ligadas a testes de codificação, de usabilidade, de jogabilidade, etc. Este papel tem ganhado destaque tanto em desenvolvimento de jogos, quanto no desenvolvimento de outros tipos de software. Isso se deve, principalmente, pela necessidade de diminuir o número de falhas e conseqüentemente, custos de produção.

Após apresentadas informações gerais sobre o ciclo de vida, papéis e artefatos de um projeto de desenvolvimento de jogos eletrônicos, será feito, no próximo capítulo, um estudo mais detalhado das metodologias existentes na literatura. Serão apresentadas características peculiares de algumas metodologias, já que, no geral, todas elas são aderentes às características apresentadas neste capítulo.

5. Metodologias de Desenvolvimento de Jogos

Neste capítulo serão apresentadas algumas metodologias de desenvolvimento de jogos, encontradas em literatura. A maioria destas metodologias surgiu de adaptações de outras metodologias pré-existentes, como o *Extreme Programming* e o *Rational Unified Process (RUP)*. Esta necessidade de adaptação se deve principalmente a 2 aspectos que concernem projetos de jogos eletrônicos:

- I. As equipes de projetos de jogos costumam ser multidisciplinares, onde programadores e artistas, por exemplo, então inseridos;
- II. Os projetos de desenvolvimento de jogos costumam ser bastante dinâmicos, e altamente susceptíveis a mudanças do ambiente e do negócio;

5.1. Game Waterfall Process

O Processo Cascata para Jogos é uma versão adaptada do tradicional processo de desenvolvimento de sistemas denominado Cascata. Ele foi batizado com este nome em alusão a uma cascata, onde o fluxo de água tem um único sentido. Assim, analogamente, no modelo Cascata todas as atividades envolvidas com a produção de um sistema ocorrem de forma seqüencial.

No Processo Cascata adaptado ao desenvolvimento de jogos é possível observar uma relação direta entre as fases de desenvolvimento previstas no modelo original e as suas. A figura a seguir apresenta a relação:

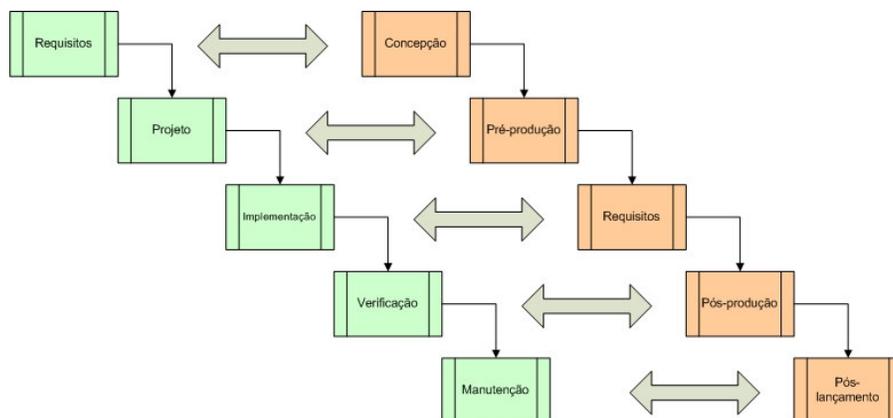


Figura 18. Fases do Modelo Cascata Tradicional e Adaptado para Jogos

Apesar de ainda ser bastante utilizado, tanto o modelo tradicional quanto o modelo adaptado apresenta sérios problemas relacionados à produtividade, manutenibilidade e reposta a mudanças. Isso porque, devido a sua estrutura seqüencial, problemas que poderiam ser resolvidos em fases iniciais do projeto acabam crescendo e trazendo mais transtornos por serem detectados tardiamente. Esses problemas costumam levar projetos ao fracasso, seja pela não adequação ao escopo acordado ou pelo “estouro” de prazo e custos.

5.2. Extreme Game Development

O Extreme Game Development (XGD) é uma metodologia ágil de desenvolvimento de jogos, baseado no Extreme Programming (XP). Basicamente, todos os princípios e a maioria das práticas do XP se aplicam também ao XGD. Mas então, por que não usar simplesmente o XP? A resposta é simples: como o Extreme Programming foi criado por programadores, algumas práticas precisam ser adaptadas para os outros papéis que compõem uma equipe de desenvolvimento de jogos. (artistas, engenheiros de som, etc.)

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. Isto faz com que a equipe de desenvolvimento trabalhe sobre grande pressão e aumenta as chances da entrega de *milestones* instáveis. [CARVALHO]

Assim como o Extreme Programming, o Extreme Game Development se baseia em 5 princípios ou valores:

Valor	Descrição
 Comunicação	A comunicação entre os integrantes da equipe é de suma importância. Comunicação auxilia na solução de problemas e aumenta o sentimento de cooperação entre a equipe. Além disso, comunicação direta deve ser feita em detrimento a documentos formais.
1 + 1 = 2 Simplicidade	Simplicidade é talvez o principal valor do XGD. A lei do XP e do <i>Extreme Game Development</i> é: “Faça o item que funcione, da forma mais simples possível.”
 Feedback	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. [BECK]
 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> [BECK].
 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. Assim como no <i>Extreme Programming</i> , no <i>Extreme Game Development</i> todos são responsáveis pelo projeto. [BECK]

Tabela 3. Princípios do Extreme Game Development

Para executar o trabalho visando à utilização desses valores, o Extreme Programming possui um conjunto de práticas. Na prática, nem todas as práticas são utilizadas em um mesmo projeto, apenas um subconjunto delas.

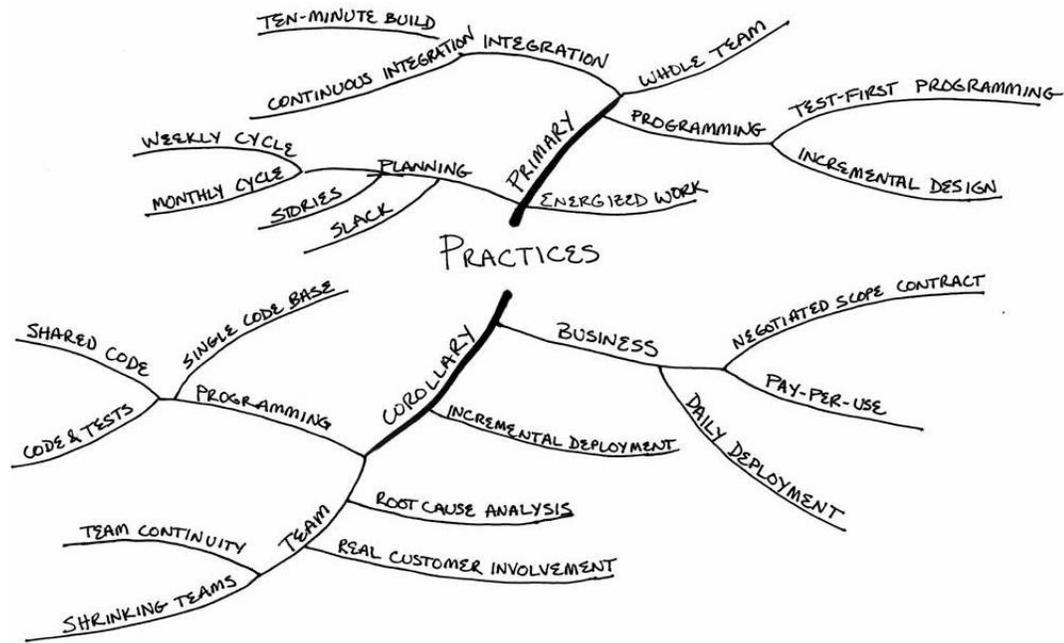


Figura 19. Práticas do Extreme Programming

Assim, o *Extreme Game Development* manteve algumas dessas práticas, com poucas ou nenhuma alteração em relação a sua base, no *Extreme Programming*. A tabela abaixo apresenta algumas dessas práticas que foram mantidas:

Prática	Descrição
<i>Whole Team</i>	A equipe deve permanecer coesa em comunicação. A equipe é um todo, e não a soma de forças individuais.
<i>Incremental Design</i>	As tarefas (<i>assets</i>) de um jogo devem ser feitas da forma mais simples possível, que funcionem corretamente.
<i>User Stories</i>	São pequenas descrições das funcionalidades do jogo, descritas normalmente pelo cliente. Muitas vezes, que tem esse papel de cliente são as Publicadoras (<i>Publishers</i>).
<i>Weekly Cycle</i>	O projeto é organizado e planejado para ser executado em ciclos de curta duração.

Tabela 4. Práticas do *Extreme Game Development*

Prática	Descrição
<i>Continuous Integration</i>	No projeto não existem, por exemplo, pedaços do código-fonte. O projeto está continuamente integrado e funcionando.
<i>Shared Code</i>	Todos da equipe compartilham todo código-fonte do projeto. Não existe a idéia do “Programador especialista em Sistema de Partículas”. [DEMACHY]
<i>Stand-up Meetings</i>	Reuniões rápidas de projeto, com objetivo de que toda a equipe esteja ciente do trabalho que está sendo feito num dado momento.

Tabela 5. Práticas do *Extreme Game Development* (continuação)

Com essas e outras práticas do XP e outras do RUP, como por exemplo, a utilização de UML, o *Extreme Game Development*, criada na Titus®, empresa Francesa de Jogos, possui muitos pontos positivos e que se adéquam bem a realidade de empresas de desenvolvimento de jogos.

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. [CARVALHO]

5.3. Game Unified Process

Criado por Kevin Flood, um gerente de projetos da área de jogos, o *Game Unified Process* ou simplesmente GUP, foi concebido com a proposta de unir características presentes em duas outras metodologias de desenvolvimento: o *Extreme Programming* e o RUP⁶. Apesar desta proposta promissora, não se pôde encontrar na literatura, relatos precisos da utilização do GUP.

A motivação para a criação desta metodologia por parte de Kevin e sua equipe foram: o grande número de problemas encontrados na metodologia Cascata para Jogos (*Game Waterfall Process*), citado na seção 5.1 e o fato de que, ambas as metodologias consideram ser, o desenvolvimento de software um processo iterativo, e não linear.

⁶ Mais informações sobre o *Rational Unified Process* podem ser encontradas no Apêndice B deste mesmo documento.

No projeto que serviu para a gênese do processo, os diversos grupos existentes na produção do jogo (desenvolvedores, artistas, etc.) utilizaram diferentes práticas, tanto do XP quanto do RUP. Na prática, a principal proposta do GUP é unir o foco em iterações curtas do XP ao foco em iterações longas do RUP. Adaptando cada um desses focos às equipes multidisciplinares envolvidas no desenvolvimento de um jogo.

5.4. Scrum

O *Scrum* é um processo ágil, iterativo e incremental que foca na aceitação das mudanças que podem ocorrer, principalmente em contextos pouco definidos. Ele foi primeiramente descrito por Takeuchi e Nonaka em seu trabalho chamado "*The New Product Development Game*" e embora não seja exatamente uma metodologia de desenvolvimento específica para jogos, vem sendo usada por algumas empresas ao redor do mundo.

Essa premissa de aceitação das mudanças que podem ocorrer e adaptação às mesmas vai de encontro a metodologias preditivas, como o RUP. Estas metodologias, ditas preditivas consideram que é possível prever todas as necessidades de um processo, antes do mesmo ser executado. [SCHWABER]

Na execução do *Scrum*, há um facilitador chamado de *Scrum Master*. Ele é responsável basicamente por monitorar a execução da metodologia e servir como mediador entre a equipe de desenvolvimento e o *Product Owner*. Este último é o responsável, dentro da equipe do projeto, por representar o patrocinador do projeto. Assim, ele é o responsável por priorizar e acompanhar as funcionalidades a serem desenvolvidas no projeto.

Além de definir papéis, o *Scrum* prevê também a confecção de alguns artefatos que auxiliam o gerenciamento do projeto. São eles:

Artefato	Descrição
<i>Product Backlog</i>	É uma lista das funcionalidades que o produto deve ter. Estas funcionalidades são priorizadas pelo <i>Product Owner</i> .
<i>Sprint Backlog</i>	Lista de tarefas a serem realizadas em cada iteração (<i>sprint</i>) do projeto. Essa lista é normalmente definida em negociação pelo <i>Product Owner</i> e a equipe de desenvolvimento
<i>Impediment List</i>	Esta lista contém todos os empecilhos encontrados pela equipe de desenvolvimento, que a impede de alcançar seus objetivos. Estes problemas devem ser resolvidos pelo <i>Scrum Master</i> .
<i>Product Backlog Burn Down</i>	Relatório contendo as funcionalidades que já foram completadas. Permite a visualização do <i>status</i> do projeto como um todo.
<i>Sprint Backlog Burn Down</i>	Relatório contendo as funcionalidades que já foram completadas em um <i>Sprint</i> específico.

Tabela 6. Artefatos do Scrum

Por último, no *Scrum*, o projeto é dividido nas seguintes fases:

- **Preparação:** Nesta fase são feitas as definições iniciais do projeto como, por exemplo, o *Product Backlog*;
- **Sprint:**
 - **Encontro de Planejamento do *Sprint* (*Sprint Planning Meeting*):** Este é o primeiro encontro de um *Sprint*. Aqui o *Product Owner* prioriza as funcionalidades presentes no *Product Backlog* e define o que deve ser entregue no *Sprint* (Incremento do Produto);
 - **Encontro Diário do *Scrum* (*Daily Scrum Meeting*):** Estes encontros diários servem para que todos da equipe possam apresentar aos outros integrantes em que trabalharam, o que

ainda falta ser feito e quais os problemas encontrados enquanto a atividade estava sendo realizada;

- **Incremento do Produto (*Product Increment*):** O conjunto de funcionalidades que devem ser entregues (acordadas) em um *Sprint*;
 - **Revisão do *Sprint* (*Sprint Review*):** É neste momento que o *Product Owner* valida o *Product Increment* entregue pela equipe. Caso não esteja satisfeito, o *Product Owner* pode solicitar algumas mudanças;
 - **Retrospectiva do *Sprint* (*Sprint Retrospective*):** Este encontro busca levantar, entre todos da equipe, os pontos positivos e negativos observados durante o último *Sprint* realizado;
 - **Atualização do *Backlog* do Produto (*Update Product Backlog*):** Aqui o *Product Owner* é responsável por dar uma nova classificação quanto a prioridade, para as funcionalidades a serem entregues no próximo *Sprint* do projeto.
- **Encerramento:** Realizada após todos os *Sprints* do projeto. É nesta fase que é feita a entrega do produto do projeto realizado;

A figura abaixo apresenta uma visão geral do *Scrum*:

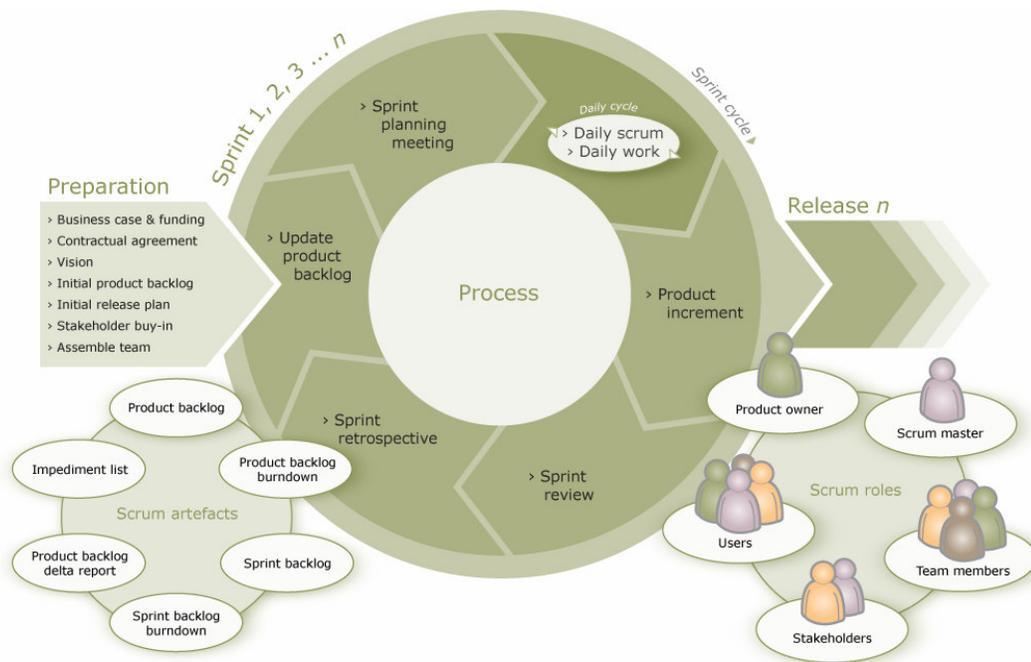


Figura 20. Visão Geral do Scrum.

5.5. Análise Comparativa

Segundo Araújo, em seu trabalho sobre o desenvolvimento de uma Metodologia Ágil para *Advergames*, algumas métricas podem ser utilizadas para avaliar metodologias de desenvolvimento de jogos. Em seu trabalho, essas métricas foram selecionadas pela adequação e necessidade ao desenvolvimento desta categoria de jogos, ditos jogos para propaganda. As métricas utilizadas foram as seguintes: [ARAÚJO]

- I. **Flexibilidade:** Qualidade de se adaptar a ambientes dinâmicos, complexos e sujeitos a mudanças do ambiente. Essas características representam bem projetos de jogos;
- II. **Comunicação:** Capacidade de manter todos os *stakeholders* (equipe de desenvolvimento, patrocinadores, etc.) do projeto em constante comunicação e focados no objetivo, utilizando para isso mecanismos eficazes;
- III. **Suporte à Multidisciplinaridade:** Capacidade de dar suporte a equipes compostas por pessoas de formações e especialidades diferentes. Como equipes de desenvolvimento de jogos;

- IV. Gerenciamento Descentralizado:** Trata-se da capacidade de distribuir, algumas vezes, o gerenciamento do projeto entre os colaboradores com maior conhecimento em determinada atividade ou área do projeto;
- V. Tratamento de Riscos:** Capacidade de antever riscos e de possuir um planejamento para os casos em que este risco se concretiza, podendo responder aos mesmos;
- VI. Valor:** Qualidade do projeto de apresentar um resultado de valor aos interessados no projeto, antes de sua finalização;
- VII. Suporte ao Desenvolvimento:** Trata-se da capacidade de prover boas práticas de desenvolvimento para as diversas áreas envolvidas em projetos de jogos.

	<i>Game Waterfall Process</i>	<i>Extreme Game Development</i>	<i>Scrum</i>	<i>Game Unified Process</i>
Flexibilidade	*	****	****	**
Comunicação	*	****	****	**
Suporte à Multidisciplinaridade	**	***	***	***
Gerenciamento Descentralizado	*	***	****	**
Tratamento de Risco	*	**	**	***
Valor	*	****	***	**
Suporte ao Desenvolvimento	*	*	*	**
* Fraco ** Regular *** Bom **** Ótimo				

Tabela 7. Quadro Comparativo – Metodologias de desenvolvimento

Resumo:

- I. **Game Waterfall Process:** apesar de trazer a inovação de prever alguns artefatos característicos do desenvolvimento de jogos, ainda sofre do “engessamento” característico do Modelo Cascata original;
- II. **Extreme Game Development:** traz consigo práticas e valores do *Extreme Programming* original. Com isso, apesar de muitas qualidades, o que a credencia para projetos de desenvolvimento de jogos, algumas deficiências ainda permanecem. Algumas delas: pouco ou nenhum planejamento de riscos, práticas não adaptadas aos “não-programadores” do projeto, etc.
- III. **Scrum:** outra metodologia ágil, esta teve a avaliação bastante semelhante ao *Extreme Game Development*, divergindo apenas no quesito valor, já que, na descrição oficial, as iterações do *Scrum* são mais longas, o que pode representar, em alguns casos, demora na entrega ao cliente.
- IV. **Game Unified Process:** Apesar do criador desta metodologia afirmar que esse processo de desenvolvimento une boas práticas do *Extreme Programming* e do *Rational Unified Process* (RUP), não fica claro quais práticas se aplicam em cada caso, dentro do projeto. Assim sendo, seu julgamento levou em consideração uma média das qualidades e defeitos de cada metodologia.

Após terem sido analisadas algumas metodologias presentes na literatura é hora de verificar se essas informações coletadas se aplicam às empresas brasileiras, mais especificamente empresas do cenário Recife. O próximo capítulo apresenta uma análise da aderência dessas empresas às práticas observadas nas metodologias analisadas neste capítulo.

6. Análise de Uso na Prática

Seguindo a análise de práticas e metodologias utilizadas em empresas de desenvolvimento de jogos, neste capítulo serão apresentados os resultados obtidos junto às empresas locais. Para obtenção de informações sobre essas empresas de jogos do cenário recifense, foi aplicado um questionário a alguns funcionários de 3 empresas. A análise dos resultados pode ser vista a seguir.

6.1. Introdução

O questionário aplicado teve como objetivo capturar algumas das definições e práticas utilizadas nas empresas questionadas. O questionário foi composto por questões com múltiplas escolhas e foi endereçado a gerentes de projetos e desenvolvedores com total conhecimento do processo de desenvolvimento de duas empresas.

Após terem sido respondidos, os questionários foram compilados e analisados. A categorização das questões, as repostas apresentadas e uma análise dos resultados serão apresentadas em outras subseções deste mesmo capítulo.

6.2. As Questões

Como dito anteriormente, as questões aplicadas objetivavam obter informações sobre práticas utilizadas nas empresas de jogos recifenses. As questões são as seguintes:

- a) Quais dos seguintes artefatos costumam ser desenvolvidos nos projetos de sua empresa?

Análise: O gráfico abaixo (figura 21) apresenta o percentual referente às respostas fornecidas para esta questão. Assim, é possível observar que, todos os artefatos observados na consulta à literatura são utilizados por pelo menos uma empresa. Porém, para fins de inserção do artefato no “Manual de Boas Práticas” do capítulo seguinte, foram considerados apenas os artefatos que se mostraram presentes em pelo menos 50% das empresas.

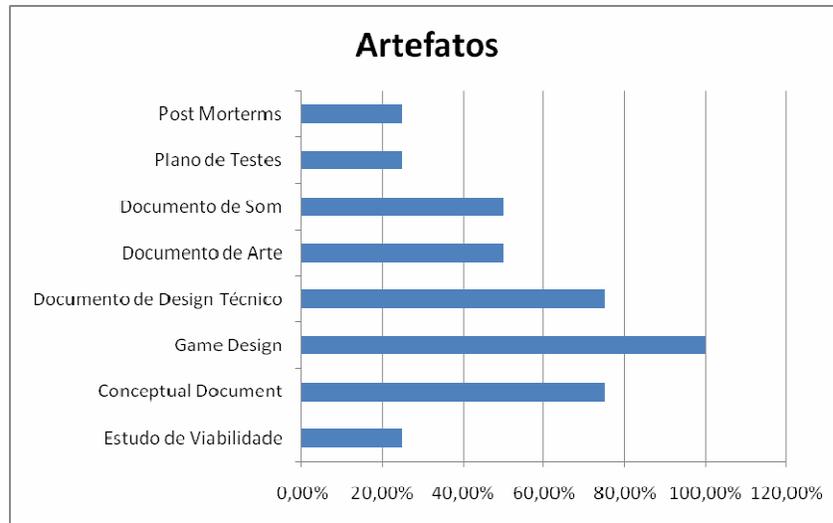


Figura 21. Gráfico de utilização de artefatos nas empresas

b) Quais os seguintes papéis costumam fazer parte da equipe de desenvolvimento dos jogos de sua empresa?

- ✓ **Análise:** Nesta questão, as empresas entrevistadas foram quase unânimes no que diz respeito aos papéis presentes em seus projetos. A observação fica por parte do papel do Produtor, que para muitos tem a mesma função de um Gerente, em projetos de jogos. Assim, é possível considerar que Gerente de Projeto e Produtor são os mesmos papéis, o que garante 100% de presença nas empresas analisadas.

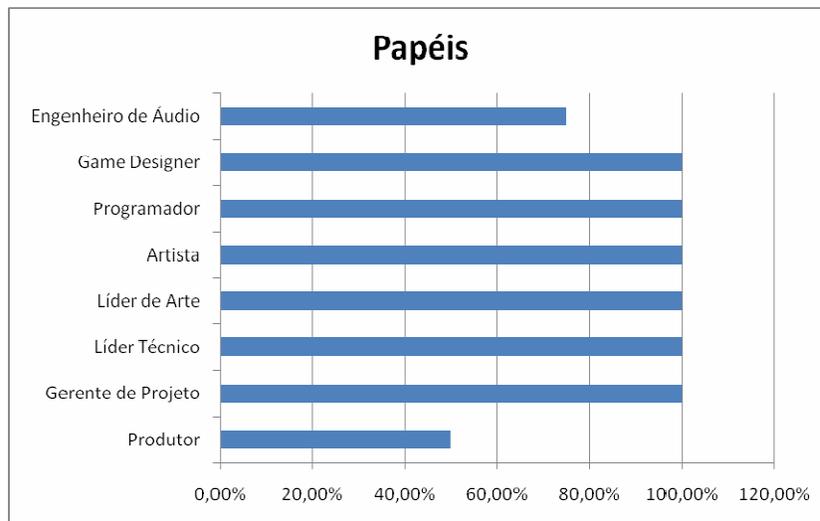


Figura 22. Gráfico da presença de papéis nas empresas

c) Os jogos desenvolvidos por sua empresa costumam sair em que versões, antes da versão final?

- ✓ **Análise:** Versões Beta e Alpha de jogos diferenciam-se basicamente pelo fato de que a primeira é testada por alguns jogadores externos, e não apenas pela equipe de desenvolvimento. A figura 23 abaixo apresenta o resultado.

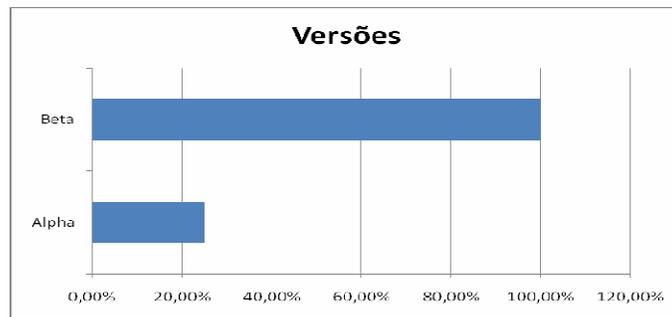


Figura 23. Versões de jogos

d) O lançamento dos jogos de sua empresa costuma ser feito por Publicadoras (*Publishers*)?

- ✓ **Análise:** Publicadoras costumam ser “canais de comunicação” nos casos em que empresas de jogos locais desejam lançar seus jogos no Mercado Internacional. Como a própria área fim de umas das empresas entrevistadas é o desenvolvimento de jogos publicitários, é coerente que uso de Empresa Publicadoras não sejam uma unanimidade entre os entrevistados.

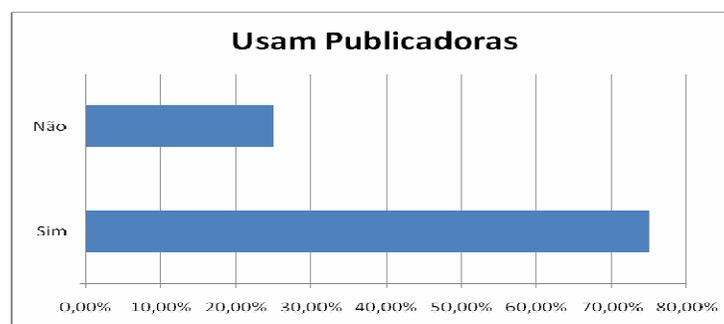


Figura 24. Uso de Publicadoras

e) Qual destas frases descreve melhor a participação dos patrocinadores, durante o processo de desenvolvimento dos projetos realizados em sua empresa?

✓ **Análise:** Ponto crítico no desenvolvimento de softwares em geral, a participação ativa dos patrocinadores (*stakeholders*) pode ser determinante para o sucesso do projeto. A figura 25 apresenta os resultados das respostas coletadas. Esses resultados demonstram certa dificuldade em inferir o grau de interação ideal com os patrocinadores do projeto. Via de regra, quanto maior o contato com o patrocinador/cliente, maiores são as chances de sucesso de um projeto.

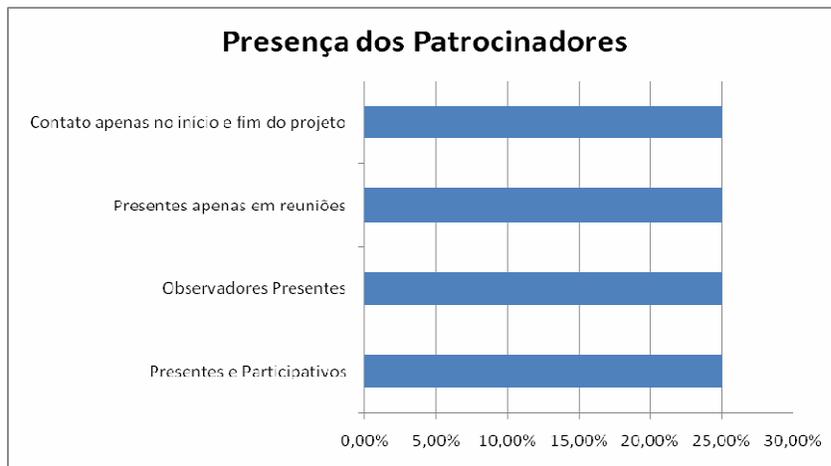


Figura 25. Presença dos Patrocinadores

f) As equipes de projeto de sua empresa costumam ser compostas por quantos integrantes?

✓ **Análise:** Os dados apresentados pela figura 26 demonstram que todas as empresas de jogos locais trabalham com equipes pequenas.

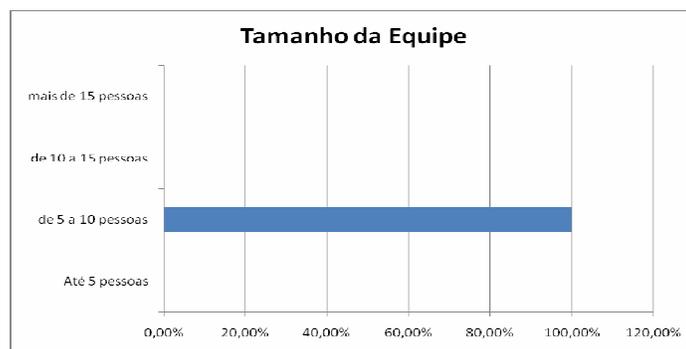


Figura 26. Tamanho das Equipes

- g) Os projetos realizados em sua empresa costumam ser feitos de forma de forma iterativa? (por exemplo, vários ciclos de desenvolvimento e testes)

✓ **Análise:** Como era previsto, a maioria das empresas realiza projetos de forma iterativa. Essa característica se mostra importante pela necessidade de rápidas respostas a mudanças que possam ocorrer durante o desenvolvimento.

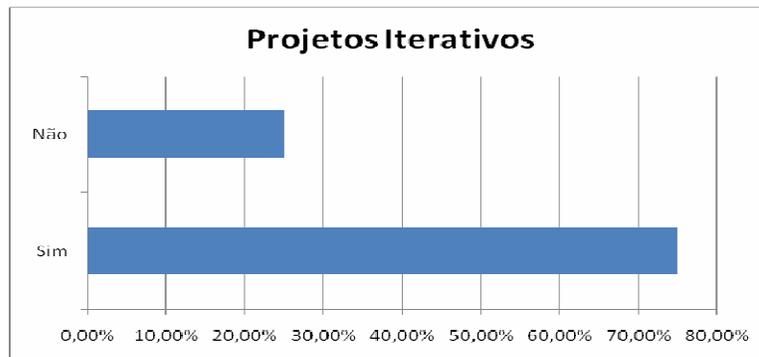


Figura 27. Número de Projetos Iterativos

- h) Se você respondeu “Sim” para a questão anterior, responda essa: os ciclos dos projetos da sua empresa costumam ter que duração?

✓ **Análise:** Aqui a análise foi realizada de forma qualitativa. Isso leva a crer que a percepção da duração dos ciclos de execução do projeto pode variar de empresa para empresa. Assim, de uma forma genérica, os ciclos devem ser de curta a média duração.

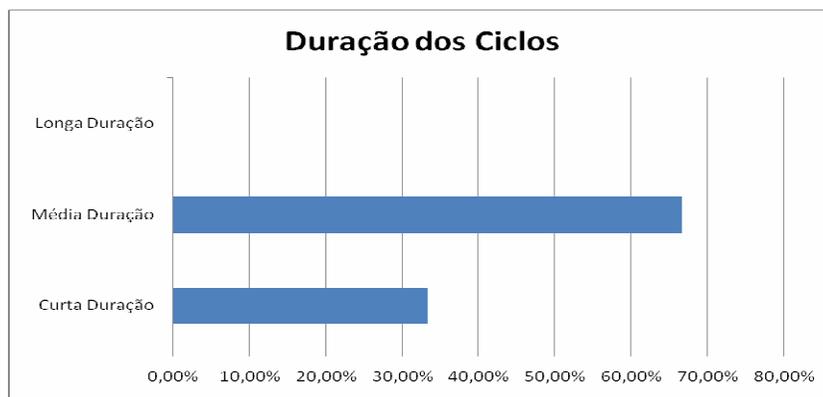


Figura 28. Duração dos ciclos de projetos iterativos

- i) Se você respondeu “Sim” na questão “g”, responda essa: Após cada ciclo de desenvolvimento, costumam ocorrer reuniões retrospectivas? (onde os pontos positivos e de possibilidade de melhorias são discutidos)

✓ **Análise:** Reuniões de Retrospectivas são importantes para que sejam coletados os pontos positivos e negativos de cada iteração, para que os acertos não sejam repetidos e os erros corrigidos nas iterações seguintes. Todas as empresas entrevistadas realizam essas reuniões.

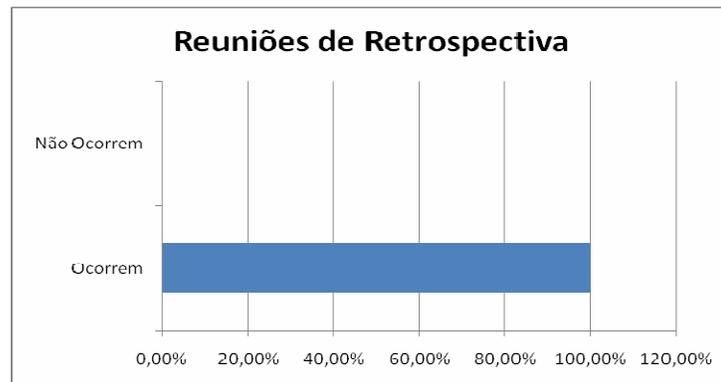


Figura 29. Ocorrência de reuniões de retrospectivas de iterações

- j) Em sua empresa, são comuns reuniões diárias? (*stand-up meetings*)

✓ **Análise:** Mais um tipo de reunião analisada. $\frac{3}{4}$ das empresas entrevistadas realizam estas reuniões, que servem para, rapidamente, manter toda equipe coesa e em comunicação.

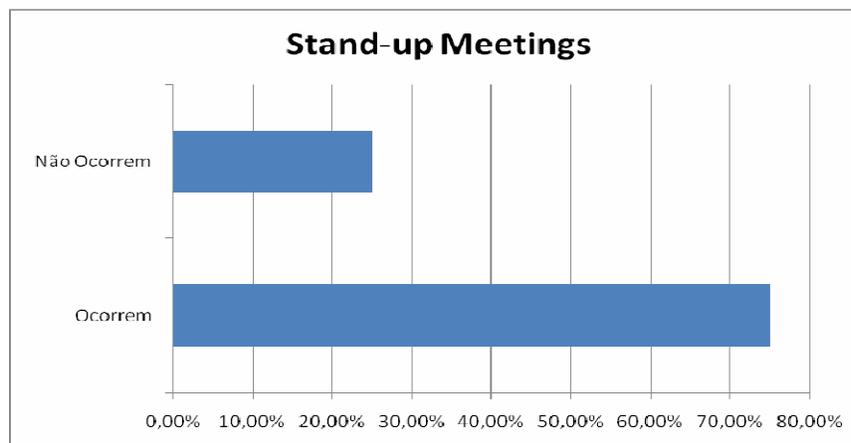


Figura 30. Ocorrência de Stand-up Meetings

- k) Em sua empresa, os colaboradores costumam ter especialidades. (Ex.: Programador especialista em física do jogo)

✓ **Análise:** Os entrevistados se mostraram divididos em relação a esta prática. Usar pessoas para tarefas especializadas pode trazer aumento de produtividade, porém pode ser tornar um grave problema, caso este colaborador venha a se retirar da empresa.

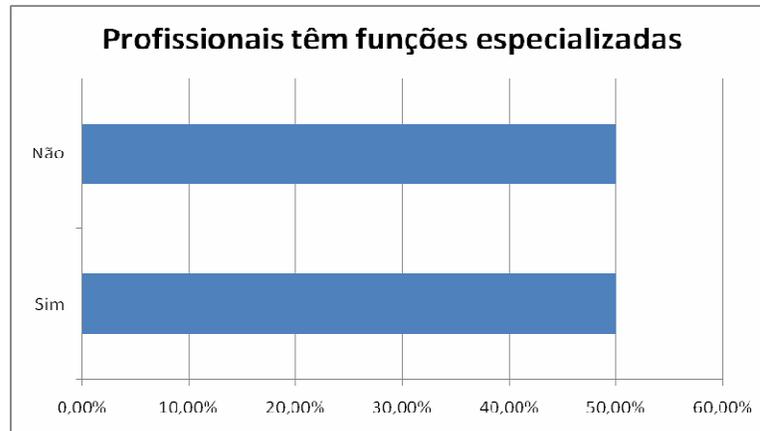


Figura 31. Utilização de profissionais com funções especializadas.

- l) As equipes de projeto de sua empresa costumam se auto-organizar? (definir papéis e tarefas entre si, sem intervenção de um líder e/ou gerente)?

✓ **Análise:** Prática polêmica, a auto-organização da equipe de um projeto costuma ser bem-sucedida apenas para aquelas experientes. Novamente $\frac{3}{4}$ das empresas não permitirem a auto-organização de suas equipes, ficando a cargo do gerente fazê-lo.

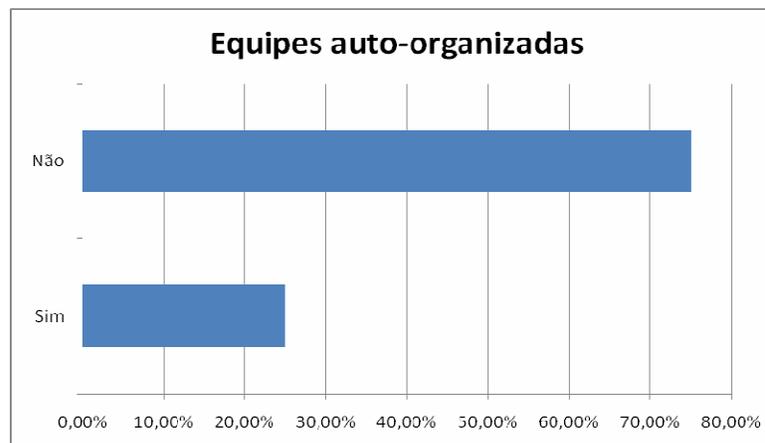


Figura 32. Presença de equipes auto-organizadas

m) Em que momento do projeto de sua empresa, os testes costumam ser feitos?

- ✓ **Análise:** É unânime a noção de que testes são necessários para todo e qualquer *software*. A figura 33 apresenta resultados que comprovam apenas que todas as empresas realizam testes não apenas no final do projeto, como no modelo Cascata.

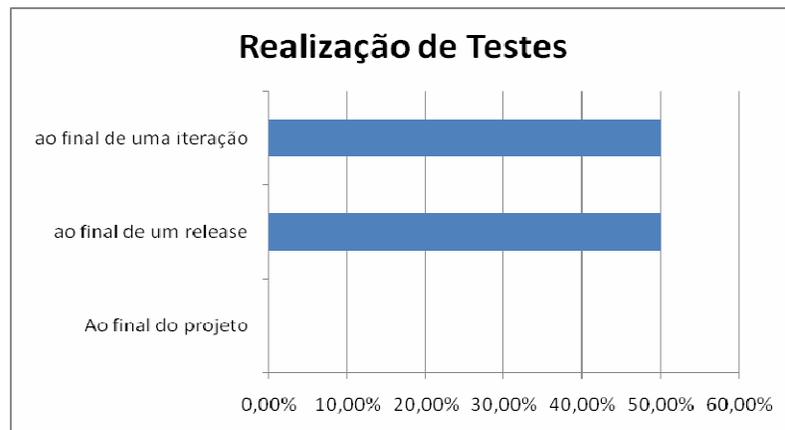


Figura 33. Momento de realização de testes no projeto

n) Sua empresa costuma utilizar o UML para criar modelos para o projeto?

- ✓ **Análise:** UML tem se mostrado ineficiente para modelar conceitos de jogos. Em alguns casos, diagramas de estados e de classes ainda são aplicados. O fato de que, 75% das empresas não utilizam o UML só comprovam o essa afirmação.

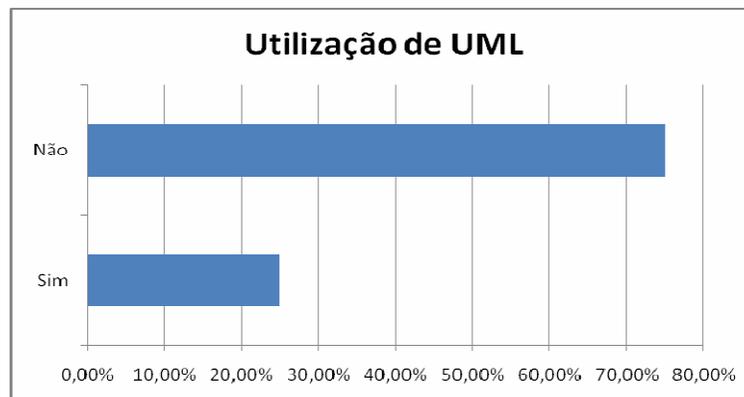


Figura 34. Utilização de UML na modelagem de projetos

- o) Sua empresa costuma utilizar algum *software* para controle de versão (ex.: SVN, CVS, etc.)?

✓ **Análise:** Unanimidade entre as empresas de desenvolvimento de *software*. Aqui, todas as empresas entrevistadas afirmaram também utilizarem *softwares* para controle de versão.

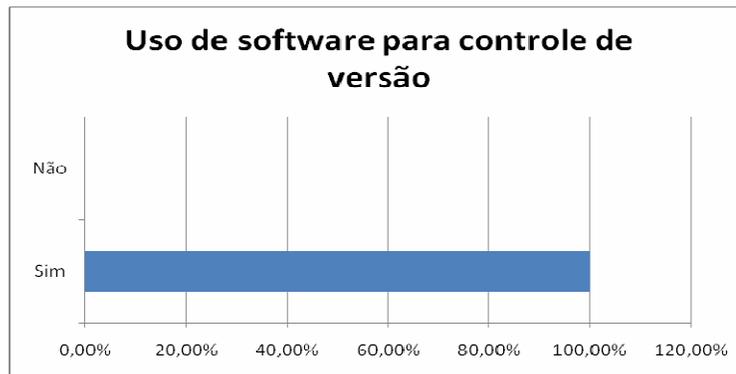


Figura 35. Uso de software de controle de versão

- p) Sua empresa costuma utilizar que ferramentas para facilitar a comunicação da equipe?

✓ **Análise:** Todas as empresas se valem de algum artifício para manter um canal ativo de comunicação entre seus colaboradores. O que varia aqui são esses tipos de artifícios (ferramentas).



Figura 36. Ferramentas de auxílio à comunicação

7. Consolidação da Análise

Neste capítulo serão apresentados os resultados dos estudos apresentados anteriormente. No capítulo 5 foram apresentadas análises de metodologias de desenvolvimento de jogos encontradas em bibliografia. Já no capítulo 6 foram apresentados os resultados de uma pesquisa realizada junto às empresas de jogos Pernambucanas, a fim de confrontar as práticas observadas nas metodologias estudadas junto às práticas utilizadas por essas empresas. Espera-se por tanto, que seja possível apresentar as práticas mais utilizadas por empresas de jogos de uma forma geral. Por outro lado, deve-se também apresentar práticas observadas de forma singular em algumas empresas, justificando, quando cabível, a não inclusão delas no conjunto de boas práticas.

7.1. Análise de Presença

Nesta seção será apresentada, de forma concisa, a presença de práticas nas metodologias estudadas e nas empresas entrevistadas. As tabelas abaixo apresentam os resultados.

Metodologias Práticas	GWP	XGD	Scrum	GUP
Desenvolvimento iterativo	Ausente	Presente	Presente	Presente
Design Incremental	Ausente	Presente	Presente	Presente
<i>Stand-up Meetings</i>	Ausente	Presente	Presente	Não relatado
Integração contínua	Ausente	Presente	Presente	Presente
Código compartilhado	Não relatado	Presente	Não relatado	Presente
Lista de Requisitos ⁷	Presente	Presente	Presente	Presente
Encontro de retrospectiva ⁸	Ausente	Ausente	Presente	Ausente
Testes contínuos	Quase inexistente	Presente	Parcialmente Presente	Parcialmente Presente
Envolvimento contínuo do cliente	Quase inexistente	Presente	Presente	Parcialmente Presente

Tabela 8. Presença de práticas nas metodologias estudadas

⁷ Em algumas metodologias pode ser conhecido como *User Stories* ou *Product Backlog* (normalmente presente no *Game Design Document*).

⁸ Encontrado normalmente em metodologias iterativas.

Práticas	Empresas	Empresa A	Empresa B	Empresa C
Desenvolvimento iterativo		Presente	Presente	Presente
Design Incremental		Parcialmente Presente	Presente	Presente
<i>Stand-up Meetings</i>		Presente	Parcialmente Presente	Presente
Integração contínua		Presente	Presente	Presente
Código compartilhado		Ausente	Ausente	Presente
Lista de Requisitos		Presente	Presente	Presente
Encontro de retrospectiva		Presente	Presente	Presente
Testes contínuos		Parcialmente Presente	Parcialmente Presente	Parcialmente Presente
Envolvimento contínuo do cliente		Presente	Quase Inexistente	Parcialmente Presente

Tabela 9. Presença de práticas nas empresas pesquisadas

7.2. Resultado da Análise

Nesta seção serão apresentados os resultados encontrados após terem sido realizados os estudos relativos às metodologias utilizadas no desenvolvimento de jogos de uma forma geral.

Após analisados os artefatos e práticas utilizadas nas Metodologias encontradas em Bibliografia e nas Empresas Locais, serão apresentadas a seguir, o que acredito ser um manual de boas práticas para desenvolvimento de jogos.

7.2.1. Artefatos Sugeridos

Seguem o subconjunto mínimo de artefatos que devem ser desenvolvidos por uma empresa que deseja desenvolver jogos e manter o controle e conhecimento dos projetos que desenvolve. São eles:

- I. Documento de Conceito do Jogo:** Documento que atesta a viabilidade operacional, técnica, econômica e de cronograma do projeto proposto. Ou seja, define o conceito de um jogo mercadologicamente bem-sucedido.

- II. **Documento de *Game Design*:** Este documento define um plano de desenvolvimento de um jogo e características áudio-visuais do mesmo. Esse é talvez o mais importante artefato desenvolvido para o projeto de desenvolvimento de um jogo.
- III. **Documento de *Design Técnico***⁹: Especifica detalhes técnicos de implementação das idéias descritas no documento de *Game Design*. Nele são apresentadas abordagens e ferramentas utilizadas, além de possíveis dificuldades durante o desenvolvimento do código-fonte do jogo.
- IV. **Documento de Arte:** Neste documento são apresentados à equipe de arte, detalhes sobre cenários, personagens e objetos que devem ser desenvolvidos para o jogo.
- V. **Documento de Som:** Como o anterior apresenta ao Engenheiro de som ou o responsável direto pelo som do jogo, detalhes da trilha sonora e efeitos sonoros que devem ser desenvolvidos para o jogo.

7.2.2. Práticas sugeridas

Será apresentado a seguir o conjunto de práticas sugeridas como as mais indicadas para o desenvolvimento de jogos. Vale salientar que, a indicação dessas práticas não exclui a inclusão de outras práticas julgadas individualmente como necessárias, dentro de uma empresa de jogos.

- I. **Desenvolvimento iterativo-incremental:** Este tipo de desenvolvimento se caracteriza pela existência de iterações de desenvolvimento do projeto. Ou seja, ao invés do projeto ser em fases seqüenciais ele é realizado em “rodadas” com planejamento, execução, testes e integração. A característica incremental fica por conta de que, ao fim de cada iteração do projeto, um incremento do produto final deve ser produzido. Em outras palavras, ao fim de cada iteração, novas funcionalidades foram acrescentadas ao produto.

⁹ Também conhecido como *Tech Doc*.

- II. **Stand-up Meetings:** Essas reuniões de caráter rápido ocorrem para que haja uma maior comunicação entre equipe de desenvolvimento do projeto. Nela cada integrante da equipe divulga para os demais, que tarefas foram completadas, que tarefas serão realizadas a seguir e que impedimentos podem ocorrer.

- III. **Integração Contínua:** Esta prática sugere que a integração das partes do projeto que foram desenvolvidas deve ser feita de forma contínua e não apenas ao final de fases de médio e longo prazo. Essa prática faz com que todos trabalhem na versão do projeto mais atual possível, fazendo que os integrantes tenham uma melhor visão do projeto como um todo. Além de diminuir os riscos de uma integração tardia, que costuma ser bastante problemática.

- IV. **Lista de Requisitos:** Conhecido de formas diferentes nas diversas metodologias, uma lista das funcionalidades/ características que o jogo deve possuir devem ser claras para a todos os *stakeholders* do projeto. Essas informações costumam estar contidas no Documento de *Game Design*.

- V. **Encontros de Retrospectiva:** Essas reuniões têm por objetivo a troca de experiências entre todos os integrantes da equipe de desenvolvimento. Nelas, cada integrante descreve os fatos que considera como pontos positivos ocorridos durante a execução de suas tarefas, referentes à iteração que acabara de terminar. Além disso, são apontados pontos passíveis de melhoria para a execução das iterações e projetos subsequentes.

- VI. **Testes a cada iteração:** Característica do desenvolvimento iterativo, a realização de testes a cada iteração deve ser tomada como premissa. Assim, antes de integrar ao projeto, cada indivíduo deve realizar, seguindo aos padrões estabelecidos pela empresa, testes para se certificar que a porção desenvolvida está correta.

- VII. Versão Beta:** A versão beta de jogos é um *release* anterior a entrega final do projeto. Nela são realizados tanto teste funcionais quanto testes de aceitação do jogo. Nesta fase os testes são realizados tanto pela equipe de desenvolvimento quanto pelos clientes e alguns potenciais consumidores.
- VIII. Uso de softwares para controle de versões:** Prática quase que unânime em empresas de desenvolvimento de software, o uso de softwares controle de versão facilitam o controle das diversas versões de artefatos desenvolvidos durante o projeto. Além de permitir que os diversos integrantes da equipe de desenvolvimento atuem em um mesmo artefato, trabalhando colaborativamente.
- IX. Uso de ferramentas de apoio a comunicação:** Diversas ferramentas na *Web* servem para diminuir fronteiras entre pessoas interessadas em interagir. Assim o uso dessas ferramentas a fim tornar a comunicação entre a equipe de um projeto se faz necessário. Alguns exemplos dessas ferramentas: *Instant Messengers*, ferramentas de vídeo-conferência, *Wiki*, *blogs*, etc.

A tabela 8 abaixo apresenta um resumo das práticas selecionadas.

Práticas
Desenvolvimento iterativo-incremental
<i>Stand-up Meetings</i>
Integração Contínua
Lista de Requisitos
Encontros de Retrospectivas
Testes a cada iteração
Versão Beta
Uso de softwares de controle de versão
Uso de ferramentas de apoio a comunicação

Tabela 10. Quadro-resumo das boas práticas selecionadas.

7.2.3. Papéis Sugeridos

Por último, após o estudo, foram identificados alguns papéis que devem assumidos por indivíduos de uma equipe de desenvolvimento de jogos. São os seguintes:

- I. **Gerente de Projeto (Produtor):** Ele é o responsável por planejar e acompanhar o desenvolvimento do projeto, envolvendo-se em todas as áreas e fases do projeto.

- II. **Game Designer:** O Game Designer é o responsável por descrever a história, os elementos, características e fluxos do jogo a ser desenvolvido. Essa atividade requer criatividade e um grande conhecimento dos jogos existentes. É também responsabilidade do Game Designer fazer com que todos os elementos do jogo sejam dispostos de forma coerente. Todos esses elementos criados pelo Game Designer são apresentados do Documento de Game Design.

- III. **Líder Técnico:** Líder da equipe de programadores. É normalmente responsável pela definição das abordagens a serem seguidas, para o desenvolvimento das funcionalidades do jogo.

- IV. **Programador:** é o responsável por codificar os elementos responsáveis pela execução do jogo. Esses elementos podem ser: parte gráfica do jogo, simulação de física, sistema de som, etc.

- V. **Líder de Arte:** Semelhante ao Líder Técnico é responsável pela definição das abordagens a serem seguidas, para o desenvolvimento do conteúdo de arte do jogo.

- VI. **Artista:** O artista é o responsável por produzir o conteúdo de arte 2D e 3D do jogo. Logo, dependendo da especialidade do artista dentro da equipe de desenvolvimento, ele pode desempenhar atividades como a concepção de personagens, modelagem 3D, desenho de texturas, etc.

VII. Engenheiro de Áudio: O Engenheiro de Áudio é o responsável por criar as trilha sonora e os efeitos sonoros do jogo. Atuando juntamente com a equipe de artistas, o Engenheiro de Som é responsável por dar identidade áudio-visual ao jogo, baseando-se no conteúdo descrito no documento de Game Design.

A tabela 9 abaixo apresenta um resumo dos papéis identificados.

Papéis
Gerente de Projeto (Produtor)
Game Designer
Líder Técnico
Programador
Líder de Arte
Artista
Engenheiro de Áudio

Tabela 11. Quadro-resumo dos papéis identificados.

7.3. Pontos em divergência

Após confrontar as informações coletadas na pesquisa bibliográfica e no questionário foram encontrados alguns pontos de divergência quanto às práticas, artefatos e papéis que devem fazer parte de um projeto de desenvolvimento de jogos. Essas divergências fizeram com que algumas práticas, ditas candidatas a serem inseridas no manual de boas práticas da seção 7.2, fossem excluídas. Em outros casos, apenas houve um equívoco em relação à definição e nomenclatura de alguns pontos. Seguem abaixo os chamados aqui como pontos de divergência:

- I. Presença de um Produtor:** Durante a resposta de alguns questionários foram obtidas respostas de que não havia o papel de Produtor na empresa em questão. Porém, segundo a definição, o produtor tem responsabilidades bastante semelhantes ao dos Gerentes de Projeto, podendo inclusive, serem considerados semelhantes. Assim, todas as empresas possuem Gerentes de

projeto, porém algumas responderão não possuírem. Isso apresenta um simples equívoco de nomenclatura.

- II. **Dimensão dos ciclos:** No questionário aplicado às empresas foi levantada a dimensão das iterações nas empresas de forma qualitativa. Na ocasião as possíveis respostas para a dimensão eram curta, média e longa duração. Porém essa percepção temporal é relativa a cada empresa. Na literatura, sugere-se que as iterações não sejam muito longas. Porém, o que para algumas empresas, uma iteração curta tem 1 mês, para outras pode ter a duração de apenas 1 semana.
- III. **Postmortem:** O Documento de *Postmortem* apresenta, ao final do projeto, os pontos positivos, negativos e lições aprendidas durante o projeto. Esse documento deve servir para que, em projetos posteriores, o número erros seja minimizado e o de acertos maximizado. A maioria das empresas não produz este artefato ao fim de um projeto, deixando a experiência apenas nas mentes dos seus participantes. Isso pode se tornar um problema quando um desses colaboradores resolve sair da empresa.
- IV. **Documento de Conceito, de Arte e de Som:** Durante o questionário, muitas empresas afirmaram não possuir alguns desses documentos. Contudo, o que ocorre algumas vezes é que essas informações encontram-se inseridas no documento de Game Design.
- V. **Plano de Testes:** Apesar de sugerido da literatura, as empresas locais não desenvolvem um plano de testes para o jogo a ser desenvolvido. Normalmente os testes são realizados apenas de forma exploratória, o que pode tornar bastante ineficaz esta prática.
- VI. **Versão Alpha:** Novamente apenas um problema de nomenclatura. Todas as empresas afirmaram realizar testes durante a fase de desenvolvimento. Os testes realizados apenas pela equipe de desenvolvimento é o que caracteriza as versões *Alpha* de jogos.

- VII. **Uso do UML para modelagem do jogo:** Mais uma prática sugerida na literatura e consagrada em algumas metodologias de desenvolvimento de software que não é comumente aplicada a empresas de jogos. O não uso do UML se dá principalmente pela característica ágil desses projetos e pela própria deficiência no modelo, quando utilizado para a modelagem de jogos.

A tabela 10 abaixo apresenta um resumo dos pontos de divergência identificados.

Pontos de Divergência
Presença de um Produtor
Dimensão dos ciclos
<i>Post Morterms</i>
Documento de Conceito, de Arte e de Som
Plano de Testes
<i>Versão Apha</i>
Uso do UML para modelagem do jogo

Tabela 12. Quadro-resumo dos pontos divergentes.

8. Conclusão

Este capítulo discute as conclusões obtidas a partir deste trabalho e sugere possíveis melhorias futuras para tornar este trabalho melhor e mais completo.

8.1. Conclusões

O Mercado de jogos eletrônicos é um dos mais lucrativos atualmente. Os bilhões de dólares faturados a cada ano comprovam este fato. Atualmente, alguns projetos de jogos chegam a custar 500 mil dólares, representando um ponto crítico para o sucesso de algumas organizações. Assim, torna-se necessária a implantação de medidas que elevem o nível de qualidade de jogos e diminua a taxa de erros de projetos dessa natureza.

Buscando entender melhor deste notável Mercado, este trabalho de graduação foi idealizado para realizar uma análise de uma abordagem já consagrada na elevação da qualidade de produtos e serviços: processos e metodologias. Assim, o objetivo deste trabalho é a realização de uma análise destas metodologias.

No 1º capítulo deste documento foram apresentados dados históricos sobre a indústria de jogos, como surgiram e sua evolução até os dias atuais. Isso para tornar o leitor, mais interessado sobre o universo de jogos. No capítulo seguinte, foram apresentados dados mercadológicos sobre a indústria de jogos. Lendo este capítulo, é possível ter uma noção da magnitude deste mercado bem-sucedido, que vem, por exemplo, faturando mais que a indústria do Cinema Americano há alguns anos.

Não esquecendo que um jogo eletrônico é uma categoria de software, o terceiro capítulo apresenta dados que mostram os problemas encontrados em projetos de jogos em geral. Este capítulo, junto aos dois anteriores, justifica a realização deste trabalho, mostrando a relevância da área de pesquisa e a necessidade de realizar um trabalho que ele eleve a qualidade de projetos de desenvolvimento de jogos eletrônicos.

Após entender as razões para a realização deste trabalho, é hora de situar o leitor no universo dos projetos de jogos. O capítulo 4 realiza esse trabalho de introdução, até chegar ao capítulo seguinte, onde são analisadas metodologias de desenvolvimento de jogos existentes em literatura. Após analisadas estas metodologias, observando papéis, fases, artefatos, dentre outros, é hora de checar se as empresas de jogos nacionais, mais especificamente pernambucanas estão aderentes às práticas observadas. O sexto capítulo apresenta resultados os questionários aplicados juntos a estas empresas.

Ao final destas análises realizadas via literatura e questionários, foi realizada uma consolidação dos resultados observados. O sétimo capítulo apresenta um “Manual de Boas Práticas” para o desenvolvimento de jogos. Este conjunto de práticas é o principal produto deste trabalho, tendo o objetivo de fornecer as empresas de jogos, principalmente as iniciantes, um guia que pode auxiliar a realização de projetos bem-sucedidos.

Assim, como principais contribuições deste trabalho podemos enumerar:

1. Análise do Mercado de Jogos;
2. Análise da situação de projetos de software em geral;
3. Apresentação de características de projetos de jogos;
4. Estudo de Metodologias de desenvolvimento de jogos;
5. Pesquisa de campo para a observação dos processos e atividades utilizadas por empresas locais;
6. Apresentação de um “Manual de Boas Práticas” para o desenvolvimento de jogos eletrônicos.

Foi possível observar, ao fim de trabalho que, mesmo com uma Indústria forte, o Mercado de jogos ainda se mostra imaturo em relação a processo de desenvolvimento, principalmente quando se trata de empresas de pequeno e médio porte, maioria das empresas Brasileiras. Assim, é necessária a realização de estudos mais aprofundados sobre o tema. A realização destes estudos poderá contribuir para a literatura sobre o assunto, que tem se mostrado ainda deficiente, e fornecer alicerces para que as empresas e seus gerentes possam melhor lidar com esta categoria de projetos de software, que se mostra como um dos mais complexos e gratificantes de toda a Indústria de Software.

8.2. Trabalhos Futuros

A análise realizada neste trabalho pode ser melhorada em muitos aspectos. Dentre eles:

1. Melhoria da análise das metodologias aqui apresentadas e análise de outras metodologias de desenvolvimento que não estão presentes neste trabalho;
2. Confecção de artigos detalhando aspectos específicos como fases, papéis e processos, das metodologias de desenvolvimento aqui apresentados;
3. Mapeamento da situação de empresas de jogos nacionais fora do Estado de Pernambuco, em relação aos seus respectivos processos de desenvolvimento.

4. Analisar não somente algumas das práticas, consideradas indicadas por empresas do setor. E sim realizar também, um estudo para entender melhor o porquê do fato que, algumas das práticas consagradas no desenvolvimento de softwares de uma forma geral, não se aplicam a projetos de jogos.

Referências

[FLOOD] FLOOD, K. Game Unified Process. Disponível em: <<http://www.gamedev.net/reference/articles/article1940.asp>>. Acesso em: 12 abr. 2007.

[ARAUJO] ARAÚJO, A. R. S. Agile Game Process – Metodologia Ágil para Projetos de Advergimes. 2006. 130p. Trabalho de Graduação (Graduação em Ciências da Computação). Universidade Federal de Pernambuco.

[WIKI – I] Wikipédia – A Enciclopédia Livre. História do Videogame. Disponível em: <http://pt.wikipedia.org/wiki/Hist%C3%B3ria_do_Video-Game>. Acesso em: 28 abr. 2007

[SOMMERVILLE] SOMMERVILLE, I. Engenharia de Software. Addison-Wesley Press. Edição: 6ª. Ano: 2003.

[CARVALHO] CARVALHO, G. H. P. Um Modelo Preditivo para Desenvolvimento de Jogos de Computador. 2006. 102p. Trabalho de Graduação (Graduação em Ciências da Computação). Universidade Federal de Pernambuco.

[ESA06] 2006 Essential Facts about Computer and Video Game Industry. Pesquisa publicada pela Entertainment Software Association (ESA). Ano: 2006.

[SOFTEX] Relatório SOFTEX. Tecnologias de Visualização na Indústria de Jogos Digitais. Disponível em: <<http://observatorio.softex.br>>. Ano: 2005.

[CHAOS04] Chaos Report 1994. Pesquisa publicada pelo The Standish Group International. Disponível em: <http://www.standishgroup.com/sample_research/chaos_1994_1.php>. Acesso em: 11 de maio 2007.

[EXTREME] Extreme CHAOS. Pesquisa publicada pelo The Standish Group International. Disponível em: <<http://www.smallfootprint.com/Portals/0/Standish%20Group%20-%20Extreme%20Chaos%202001.pdf>>. Acesso em: 11 mai. 2007.

[WIKI – II] Wikipédia – A Enciclopédia Livre. Engenharia de Software. Disponível em: <http://pt.wikipedia.org/wiki/Engenharia_Software>. Acesso em: 12 mai. 2007.

[SLOPER] SLOPER, T. Following Up After the Game is Released: It's not Over when it's Over. Game Design Perspectives. 2002.

[OLSEN] OLSEN, J. 2003 Game Development Salary Survey. Gamasutra Article. Disponível em: <http://www.gamasutra.com/features/20040211/olsen_pfv.htm>. Acesso em: 4 de junho de 2007.

[PMBOK] Project Management Body of Knowledge. Project Management Institute (PMI). Edição: 1. Ano: 2000.

[WIKI – III] Wikipédia – A Enciclopédia Livre. Crise do Software. Disponível em: <http://pt.wikipedia.org/wiki/Crise_do_software>. Acesso em: 30 abr. 2007

[RUP] JACOBSON, I.; et. al. The Unified Software Development Process. Addison-Wesley Press. Edição: 1. Ano: 1999.

[BECK] BECK, K., ANDRES, C.. Extreme Programming Explained: Embrace Change. Second Edition. 2004.

[DEMACHY] DEMACHY, T.. Extreme Game Development: Right on Time, Every Time. Gamasutra Article, 2003. Disponível em: <www.gamasutra.com/resource_guide/20030714/demachy_pfv.htm>. Acesso em: 12 jun. 2007.

[SCHWABER] SCHWABER, K. Agile Project Management with Scrum. Microsoft Press. Edição: 1ª. Ano: 2004.

[WIKI - IV] Wikipedia, the free encyclopedia. Scrum (development). Disponível em: <http://en.wikipedia.org/wiki/Scrum_%28development%29>. Acesso em: 25 jul. 2007

Apêndice A – Glossário

- **Advergames** – jogos criados com vocação publicitária com o objetivo de passar uma mensagem que contém propaganda.
- **Arcade** - é um videogame profissional usado em estabelecimentos de entretenimento, por vezes também chamado no Brasil de fliperama.
- **Console** – é um videogame destinado para jogadores caseiros;
- **Engine** – software que possui implementado, alguns conceitos que podem ser utilizados na criação de jogos. Por exemplo, *Engines* para simulação de física.
- **Framework** - é um conjunto de classes com objetivo de reutilização de um design, provendo um guia para uma solução de arquitetura em um domínio específico de software.
- **Jogabilidade** – atributo que mede a facilidade e o conforto que um jogador percebe ao jogar um jogo eletrônico.
- **Metodologias Preditivas** – metodologias que assumem ser possível prever os fatos que ocorreram no decorrer de um projeto.
- **MMORPG** - é um tipo de jogo de computador e/ou video game que permite a milhares de jogadores criarem personagens em um mundo virtual dinâmico ao mesmo tempo na Internet.
- **Release** – entregas (versões) de um produto que está em desenvolvimento.
- **Stakeholders** – todo e qualquer indivíduo que pode influenciar ou ter interesse, direto ou indireto, nos resultados de um projeto.
- **Touch-screen**: tipo especial de telas sensíveis ao toque do usuário. Elas são encontradas principalmente em alguns dispositivos móveis e terminais informatizados.
- **UML** – A *Unified Modeling Language* é uma linguagem de modelagem orientada a objetos semi-formal, que permite a modelagem de aspectos, visões e estruturas de softwares.
- **Usabilidade** - termo usado para definir a facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar uma tarefa específica e importante.

- **Wiki** – termo usado para identificar um tipo específico de coleção de documentos em hipertexto, feitos de forma colaborativa e que o conteúdo tenha que ser revisto antes da sua publicação.

Apêndice B – *Rational Unified Process*

Apesar da maioria das empresas de jogos não utilizarem o *Rational Unified Process* como processo padrão, por julgarem-no muito rígido e demasiadamente preditivo e orientado a documentos. Ele será apresentado aqui neste apêndice devido a sua importância no que diz respeito à criação ou compilação de práticas de desenvolvimento de software, utilizadas em outras metodologias.

A *Rational Software Corporation*, atual IBM, criou o chamado *Rational Unified Process*, em resposta aos problemas apresentados na metodologia Cascata. O processo criado une basicamente as melhores práticas de desenvolvimento de software. [CARVALHO]

Esse conjunto de “boas práticas” podem ser resumidas em 6. São estas:
[RUP]

- 1. Desenvolvimento Iterativo e incremental:** permite que grandes tarefas sejam decompostas em outras menores, facilitando o desenvolvimento. Além disso, essa prática de desenvolvimento progressivo aumenta o *feedback* do usuário quanto ao desenvolvimento do sistema;
- 2. Gerenciamento de Requisitos:** esta prática preocupa-se em atender às necessidades dos diversos *Stakeholders* do sistema, permitindo que o escopo do projeto seja definido. Além do mais, essa prática permite o gerenciamento das mudanças de Requisitos;
- 3. Arquitetura Baseada em Componentes:** o desenvolvimento baseado em componentes permite que a arquitetura do sistema seja extensível, compreensível e reusável; [ARAÚJO]

4. **Modelagem Visual:** a modelagem visual do sistema é uma abstração que permite o melhor entendimento das partes interessadas no sistema. O RUP recomenda o uso do UML (*Unified Modeling Language*) para modelar as diversas visões do sistema;
5. **Busca Contínua da Qualidade:** diferentemente do Modelo Cascata, onde a verificação da Qualidade era feita apenas no final do desenvolvimento do sistema, aqui a verificação é feita de forma contínua, envolvendo todos da equipe de desenvolvimento.
6. **Controle de Mudanças:** Todos os projetos de software sofrem mudanças. O Processo Unificado define métodos para controlar, acompanhar e monitorar mudanças. [ARAÚJO]

Além das boas práticas citadas anteriormente, a estrutura do RUP é composta ainda por fases e disciplinas. A figura abaixo apresenta esta arquitetura:

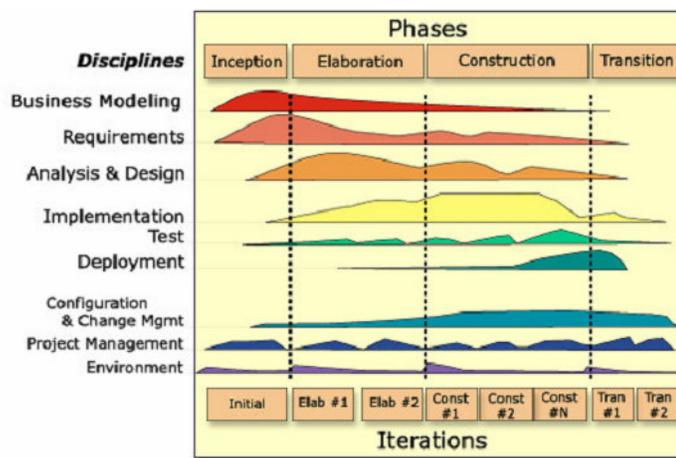


Figura 37. Arquitetura do RUP

Uma descrição detalhada das disciplinas é apresentada na tabela a seguir:

Disciplina	Descrição
Modelagem de Negócios	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.
Implementação	Faz parte desta disciplina a implementação propriamente dita dos componentes, os testes unitários para cada um dos componentes implementados e a integrações entre estes.
Testes	Aqui são realizados testes sobre o código desenvolvido. São realizados testes de integração e testes de conformidade com os Requisitos especificados. Caso problemas sejam identificados, eles devem ser resolvidos antes da implantação.
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 de 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 mudanças solicitadas e ocorridas.
Gerência de Projeto	Os objetivos desta disciplina são planejar e gerenciar o ciclo de desenvolvimento.
Ambiente	O objetivo desta disciplina é apoiar o desenvolvimento do software com ferramentas e processos.

Tabela 13. Disciplinas do RUP¹⁰

¹⁰ Retirada de CARVALHO, G. H. P. Um Modelo Preditivo para Desenvolvimento de Jogos de Computador

O RUP é composto também por fases, nas quais as disciplinas são distribuídas. A tabela a seguir apresenta uma descrição destas fases:

Fases	Descrição
Concepção	Nesta fase são definidos: o Escopo, os Custos e Cronograma do Projeto.
Elaboração	Esta fase tem o objetivo de obter o entendimento do domínio do projeto e que seja realizada a Elicitação de Requisitos.
Construção	É nesta fase em que ocorre mais intensamente a execução do projeto propriamente dita (codificação).
Transição	Por último é realizada a implantação do sistema desenvolvido, em seu ambiente de execução.

Tabela 14. Fases do RUP.

É importante observar que o RUP se destina comumente a projetos de grande porte, com grandes equipes. Num cenário com estas características, um grande nível de documentação e estruturação é desejável.