

UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA – UESB  
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

ARLEISON MEIRA ROCHA LACERDA

**PROPOSTA DE METODOLOGIA PARA O DESENVOLVIMENTO DE  
AMBIENTES VIRTUAIS TRIDIMENSIONAIS**

VITÓRIA DA CONQUISTA – BA

2016

ARLEISON MEIRA ROCHA LACERDA

PROPOSTA DE METODOLOGIA PARA O DESENVOLVIMENTO DE AMBIENTES  
VIRTUAIS TRIDIMENSIONAIS

Monografia apresentado como requisito parcial  
para obtenção do título de Bacharel em Ciência  
da Computação, Universidade Estadual do  
Sudoeste da Bahia.

Orientador: Prof. Me. Stênio Longo Araújo  
Coorientador: Prof. Dr. Francisco dos Santos  
Carvalho.  
Coorientador: Prof. Me. Marcos Gomes Prado.

VITÓRIA DA CONQUISTA – BA

2016

ARLEISON MEIRA ROCHA LACERDA

PROPOSTA DE METODOLOGIA PARA O DESENVOLVIMENTO DE AMBIENTES  
VIRTUAIS TRIDIMENSIONAIS

Monografia apresentado como requisito parcial  
para obtenção do título de Bacharel em Ciência  
da Computação, Universidade Estadual do  
Sudoeste da Bahia.

**Banca Examinadora:**

---

**Prof. Me. Stênio Longo Araújo**

---

**Prof. Dr. Francisco dos Santos Carvalho**

---

**Prof. Me. Marcos Gomes Prado**

## AGRADECIMENTOS

Gostaria de agradecer minha família, que ao modo de cada um me apoiou e incentivou durante todos estes anos. E que sem o seu amor e suporte eu não conseguiria chegar tão longe. Agradeço em especial aos meus pais, Almerídio Lacerda e Francina Meira e ao meu irmão Flaubert Lacerda que sempre estiveram presentes comigo não só nos momentos acadêmicos, mas em diversos aspectos da minha vida. Mas família não é só onde a gente nasce é também quem a gente escolhe para a gente. Externo aqui minha gratidão ao meu irmão Marllon Oliveira, uma amizade que resistiu a todas as atribulações da vida universitária, e também a minha noiva Juliana Batista, pessoa que somou e equalizou a minha vida.

Agradeço também aos amigos que estiveram comigo durante esta jornada. Com quem compartilhei alegrias, vitórias, ansiedades e frustrações dentro da universidade. Mas, não menos importantes foram também as amizades formadas fora do meu curso, com os amigos encontrados em outros cursos e dentro da Ordem DeMolay, instituição da qual tive o privilégio de vivenciar.

Agradeço também aos professores do curso de Ciência da Computação da UESB, pela dedicação, apoio e principalmente pelos ensinamentos e conhecimentos transmitidos ao longo da graduação. E por mencionar os mestres que tive, registro aqui um agradecimento especial aos meus Mestres de Karate-Do Tradicional: Erisvaldo Santos e Hermogenes Dias e por extensão a todos amigos e companheiros do clube do qual integro, a determinação que cultivei em todos esses anos de prática foram chave essencial para esta conquista.

E como princípio e fim, acima de tudo isso, Deus. Sem o seu amor e benevolência, não teria nada ao que agradecer. Gratidão por tudo e por que sou.

*“Os homens devem moldar seu caminho. A partir do momento em que você vir o caminho em tudo o que fizer, você se tornará o caminho.”*  
(Myamoto Musashi)

*“A introdução de abstrações adequadas é a nossa única ajuda mental para reduzir o apelo à enumeração, para organizar e dominar a complexidade.”* (Edsger Dijkstra)

## RESUMO

Este trabalho apresenta a proposta de um processo de software para o desenvolvimento de sistemas de ambientes virtuais tridimensionais. Integrando, assim, conhecimentos da área da engenharia de software com as especificidades da realidade virtual. Sistemas de realidade virtual são softwares altamente interativos, imersivos ou não, com elementos multimídia e estão cada vez mais presentes nos diversos setores da sociedade, em suas várias formas de aplicação: visualização científica, representação arquitetônica, simulação, entretenimento, entre outras aplicações. Entretanto, desenvolver tal sistema agrega em si uma alta complexidade e para resolver tal questão, a prática comum de qualquer engenharia para se construir um artefato complexo é construí-lo de acordo com um planejamento. Este trabalho apresenta um processo de software para o desenvolvimento de ambientes virtuais tridimensionais. A metodologia usada nesta monografia é exploratória, uma vez que delimita os objetivos de cada fase e componente da metodologia, para depois avaliá-la segundo critérios qualitativos. Como resultado deste trabalho foi possível conceber um processo de software que visa abordar e tratar de forma adequada o desenvolvimento de ambientes virtuais tridimensionais, acredita-se ainda que tal processo careça de uma avaliação e melhoria para aceitação e consolidação nos âmbitos acadêmico e de mercado.

Palavras-Chave: Processo de Software. Realidade Virtual. Tridimensional.

## ABSTRACT

This work presents a proposal for a software process for the development of three-dimensional virtual environments systems. Thus, integrating knowledges of the area of software engineering with the specifics of virtual reality. Virtual reality systems are highly interactive software, immersive or not, with multimedia elements and are increasingly present in the various sectors of society, in its various forms of application: scientific visualization, architectural representation, simulation, entertainment, among other applications. However developing such system adds itself a high complexity and to solve this issue, the common practice of any engineering to build a complex artifact is built it according to a scheme. This paper presents a software process for developing three-dimensional virtual environments. The methodology used in this work is exploratory, once it defines the objectives of each phase and components of the methodology, for late evaluate it according to qualitative criteria. As a result of this work it was possible to conceive a software process that aims to address and treat appropriately the development of three-dimensional virtual environments, although it is believed that this process requires the evaluation and improvement for acceptance and consolidation in academic scope and market.

Keywords: Software Process. Virtual reality. Three-dimensional.

## LISTA DE FIGURAS

Figura 1: Modelo de processo em cascata. ....	16
Figura 2: Modelo de processo em espiral. ....	17
Figura 3: Modelo de processo incremental. ....	18
Figura 4: Modelo de processo de prototipação. ....	19
Figura 5: Modelo de processo RUP. ....	20
Figura 6: Atividades do processo de software. ....	23
Figura 7: Metodologia para o processo de software. ....	23
Figura 8: Modelo para definição de processos em níveis. ....	25
Figura 9: Estrutura da norma ISO/IEC 12207. ....	26
Figura 10: Representação contínua - níveis de capacitação. ....	27
Figura 11: Representação em estágios - níveis de maturidade. ....	28
Figura 12: Componentes do modelo MPS.Br. ....	29
Figura 13: Aplicação com reconstrução arquitetônica. ....	32
Figura 14: Aplicação em entretenimento com jogos. ....	32
Figura 15: Aplicação na medicina com realidade aumentada. ....	33
Figura 16: Aplicação em visualização industrial. ....	33
Figura 17: Aplicação em simulação no treinamento militar. ....	34
Figura 18: Modelo de processo de desenvolvimento de SRVs. ....	36
Figura 19: Modelo de processo de Kim. ....	37
Figura 20: Atividades da monografia. ....	42
Figura 21: Visões dos domínios. ....	45
Figura 22: Fluxo geral do MAV3D. ....	46
Figura 23: Diagrama de atividade da fase de concepção. ....	50
Figura 24: Diagrama de atividade da fase de estruturação. ....	51
Figura 25: Diagrama de atividade da fase de desenvolvimento. ....	51
Figura 26: Diagrama de atividade da fase de entrega. ....	52

## LISTA DE ABREVIATURAS E SIGLAS

<b>3D</b>	Três dimensões/ Tridimensional
<b>ABNT</b>	Associação Brasileira de Normas Técnicas
<b>AV</b>	Ambiente virtual
<b>ASD</b>	Adaptive Software Development
<b>AS-CMM</b>	CMM para Aquisição de Software
<b>CASE</b>	Computer Aided Software Engineering – Engenharia de Software Auxiliada por Computador
<b>CMM</b>	Capability Maturity Model – Modelo de Maturidade em Capacitação
<b>CMMI</b>	Capability Maturity Model – Integration Modelo de Maturidade em Capacitação – Integração
<b>DER</b>	Documento de Especificação de Requisitos
<b>DPS</b>	Documento de Projeto de Software
<b>FDD</b>	Feature Driven Development
<b>GUI</b>	Graphical User Interface – Interface Gráfica do Usuário
<b>HUD</b>	Heads Up Display tela de exibição de informações dentro de um jogo.
<b>IOGE</b>	Instituições Organizadoras de Grupos de Empresas
<b>ISO</b>	International Organization for Standardization – Organização Internacional para Padronização
<b>MA- MPS</b>	Método de Avaliação do MPS
<b>MN- MPS</b>	Modelo de Negócio do MPS
<b>MPS.Br</b>	Programa de Melhoria de Software Brasileiro
<b>MR- MPS</b>	Modelo de Referência do MPS
<b>RUP</b>	Rational Unified Process
<b>RV</b>	Realidade virtual
<b>SE-CMM</b>	CMM para Engenharia de Sistema
<b>SOFTEX</b>	Associação para Promoção da Excelência do Software Brasileiro
<b>SRV</b>	Sistemas de Realidade Virtual
<b>SW-CMM</b>	CMM para Software
<b>UML</b>	Unified Modeling Language – Linguagem de Modelagem Unificada
<b>XP</b>	Extreme Programming – M

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1	QUESTÃO CENTRAL DA PESQUISA .....	12
1.2	PROBLEMA DA PESQUISA .....	12
1.3	HIPÓTESE CENTRAL .....	13
1.4	OBJETIVO .....	13
1.4.1	<b>Objetivo Geral .....</b>	<b>13</b>
1.4.2	<b>Objetivos Específicos.....</b>	<b>13</b>
1.5	JUSTIFICATIVA .....	13
1.6	ESTRUTURA DO DOCUMENTO .....	14
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>15</b>
2.1	ENGENHARIA DE SOFTWARE .....	15
2.1.1	<b>Os modelos de processos para desenvolvimento de software .....</b>	<b>15</b>
2.1.2	<b>Metodologia para o processo de software.....</b>	<b>21</b>
2.1.2.1	Estrutura do processo de software.....	22
2.1.2.2	Avaliação do processo de software .....	26
2.2	AMBIENTE VIRTUAL.....	30
2.2.1	<b>Tipos de Realidade Virtual .....</b>	<b>31</b>
2.2.2	<b>Aplicações da Realidade Virtual .....</b>	<b>31</b>
2.3	TRABALHOS RELACIONADOS .....	35
a)	<b>Fundamentos e tecnologia de realidade Virtual e Aumentada .....</b>	<b>35</b>
b)	<b>Uma Proposta para o Desenvolvimento Ágil de Ambientes Virtuais.....</b>	<b>36</b>
c)	<b>Desenvolvimento de um ambiente para visualização tridimensional da dinâmica de <i>risers</i> .....</b>	<b>37</b>
2.3.1	<b>Contribuições correlacionadas .....</b>	<b>38</b>

a)	Contribuição à Engenharia de Requisitos de Ambientes Virtuais.....	38
b)	Desenvolvimento de Concept Art para Personagens e Video Game Design: Uma análise da estética conceitual do entretenimento digital.....	39
<b>3</b>	<b>METODOLOGIA .....</b>	<b>41</b>
3.1	TIPO DA PESQUISA QUANTO AOS OBJETIVOS, ABORDAGEM E FINS.....	41
3.2	INSTRUMENTOS E MATERIAIS DA PESQUISA .....	41
3.3	TAREFAS DO DESENVOLVIMENTO DO TRABALHO.....	41
<b>4</b>	<b>MAV3D - METODOLOGIA PARA AMBIENTES VIRTUAIS TRIDIMENSIONAIS.....</b>	<b>44</b>
4.1	VISÃO GERAL E PRINCÍPIOS .....	44
4.2	HIERARQUIA DOS COMPONENTES DA METODOLOGIA .....	46
4.2.1.1	Fases.....	46
4.2.1.2	Disciplinas.....	47
4.2.1.3	Tarefas.....	48
4.3	FLUXO DAS ATIVIDADES.....	50
4.4	ESPECIALIZAÇÃO DO MAV3D .....	52
<b>5</b>	<b>RESULTADOS E DISCUSSÕES .....</b>	<b>54</b>
5.1	CONSIDERAÇÕES SOBRE O MAV3D.....	54
5.2	PROBLEMAS ENCONTRADOS E SOLUÇÕES APLICADAS .....	55
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>56</b>
6.1	CONCLUSÃO .....	56
6.2	TRABALHOS FUTUROS .....	57
	<b>REFERÊNCIAS.....</b>	<b>58</b>

# 1 INTRODUÇÃO

## 1.1 QUESTÃO CENTRAL DA PESQUISA

Como projetar uma solução computacional adequada para o desenvolvimento de ambientes virtuais tridimensionais?

## 1.2 PROBLEMA DA PESQUISA

A vida na sociedade contemporânea está em boa parte vinculada à presença de sistemas complexos baseados em computadores, que estão aplicados em uma gama significativa de áreas que abrangem a atuação do ser humano, quer seja na esfera do trabalho ou fora dela (SOMMERVILLE, 2007). Esta realidade é reflexo de todo um processo evolutivo, onde segundo Pressman (1995), há uma considerável evolução do software.

Em uma retrospectiva histórica desse processo evolutivo, pode-se relatar que – durante a década de 60, institui-se o que foi chamado de “crise de software”, como um conjunto de problemas encontrados no desenvolvimento de software, causado pelas más condutas e/ou negligências dos responsáveis. Ao que mais tarde foi chamado de “aflição crônica de software” (PRESSMAN, 1995), já que não se trata de um marco vital crítico para os softwares, mas sim, de uma situação recorrente. Em resposta a “crise” ou “aflição”, foi estruturada uma área de conhecimento denominada de Engenharia de Software, que tem por ideia introduzir práticas que possam melhorar o processo de desenvolvimento e gerar softwares de qualidade.

Devido à complexidade no desenvolvimento, o risco de se construir um sistema que não alcance seus objetivos é alto. Para evitar tal risco, a prática comum de qualquer engenharia para se construir um artefato complexo, um sistema de software no caso, é construí-lo de acordo com um plano. Em outras palavras, utilizar de uma metodologia adequada para a elaboração do sistema.

Existem diversos contextos nos quais podem ser utilizados sistemas de software. Entre eles, pode-se citar os ambientes interativos gerados por realidade virtual (RV). A ideia geral deste trabalho apresenta a proposta de um processo de software para ambientes virtuais tridimensionais (3D) – o MAV3D. Integrando assim, conhecimentos da área da engenharia de software e da realidade virtual.

### 1.3 HIPÓTESE CENTRAL

É possível desenvolver uma metodologia direcionada a ambientes virtuais 3D por meio de processos e modelos da engenharia de software adaptados as características e necessidades destes tipos de sistema.

### 1.4 OBJETIVO

#### 1.4.1 Objetivo Geral

Propor uma metodologia de desenvolvimento de software para ambientes virtuais tridimensionais.

#### 1.4.2 Objetivos Específicos

- Apresentar os elementos e princípios da criação de um processo metodológico para softwares;
- Conhecer os modelos de processo adequados para o desenvolvimento de sistemas de realidade virtual (SRV);
- Propor uma nova metodologia;
- Descrever as vantagens da metodologia proposta.

### 1.5 JUSTIFICATIVA

A investigação a ser realizada apresenta relevância acadêmica na medida em que possibilita que o pesquisador correlacione conhecimentos teóricos com conhecimentos práticos exigidos no processo de planejamento do projeto. Este trabalho de conclusão de curso vem propor uma metodologia de desenvolvimento de software direcionada aos ambientes de realidade virtual, que são softwares altamente interativos, imersivos ou não, com elementos multimídia. Além de possuírem um grande valor em seus vários campos de aplicação: simulação militar, educação, turismo, indústria e entretenimento, por exemplo.

Justifica-se também a possibilidade de compreender como avaliar e o que avaliar na construção de um AV 3D, pois existem diversos tipos de ambientes virtuais e cada um exige técnicas adequadas ao seu nível de realidade e imersão.

Por se tratar de uma atividade complexa e multidisciplinar, a boa escolha da metodologia adotada, torna-se um fator crucial para o sucesso do sistema e a aceitação no mercado.

## 1.6 ESTRUTURA DO DOCUMENTO

A estrutura desta monografia está dividida em seis capítulos. O Capítulo 1 faz uma introdução geral sobre a delimitação do tema e objetivos do trabalho, apresentando de forma breve a engenharia de software e a realidade virtual. Além de trazer os objetivos, a justificativa e a metodologia de como se deu este trabalho.

O Capítulo 2 apresenta as bases teóricas sobre as quais este trabalho foi fundamentado: os modelos de processo encontrados dentro da engenharia de software; os princípios, objetivos e elementos da criação de um processo de software; conceitos e aplicações da realidade virtual e alguns trabalhos relacionados com o tema proposto.

No Capítulo 3 encontra-se a descrição da metodologia aplicada ao projeto. Sua classificação, suas ferramentas e etapas envolvidas.

O Capítulo 4 trata do cerne desta monografia, o MAV3D. Descreve seus princípios, objetivos, práticas, etapas e fluxos de trabalho.

O Capítulo 5 apresenta algumas dificuldades encontradas e soluções aplicadas e algumas considerações sobre a metodologia de software desenvolvida. O Capítulo 6 contém a conclusão do trabalho apresentado, bem como, as perspectivas de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

### 2.1 ENGENHARIA DE SOFTWARE

Primeiramente é necessário trazer o conceito de engenharia de software. Nessa perspectiva, Sommerville (2007, p.5) define engenharia de software como:

“[...] uma disciplina da engenharia relacionada com todos os aspectos da produção de software, desde estágio inicial da especificação do sistema até sua manutenção, depois que este entrar em operação”.

Já Pressman (1995) faz uma definição mais específica do que seria a engenharia de software:

É um rebento da engenharia de sistemas e de hardware. Ela abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – o que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para construção de software de alta qualidade produtivamente.

Percebe-se que os dois autores colocam a engenharia de software, como uma atividade fim-a-fim, mas apenas o segundo acrescenta a ideia de gerente. Inserindo assim, uma nova atividade profissional que teria um papel administrativo e responsável por todo o desenvolvimento do projeto, com o poder do “controle do processo”.

#### 2.1.1 Os modelos de processos para desenvolvimento de software

Por processo de software pode-se entender como um conjunto de passos, constituído por atividades, métodos, práticas e transformações, usadas para atingir uma meta (PAULA FILHO, 2009). Dentro da engenharia de software, existem algumas divergências quanto a divisão dos modelos de processo, ou também chamados “ciclos de vida de software”, mas de modo geral, pode-se citar os seguintes modelos ou metodologias mais difundidas:

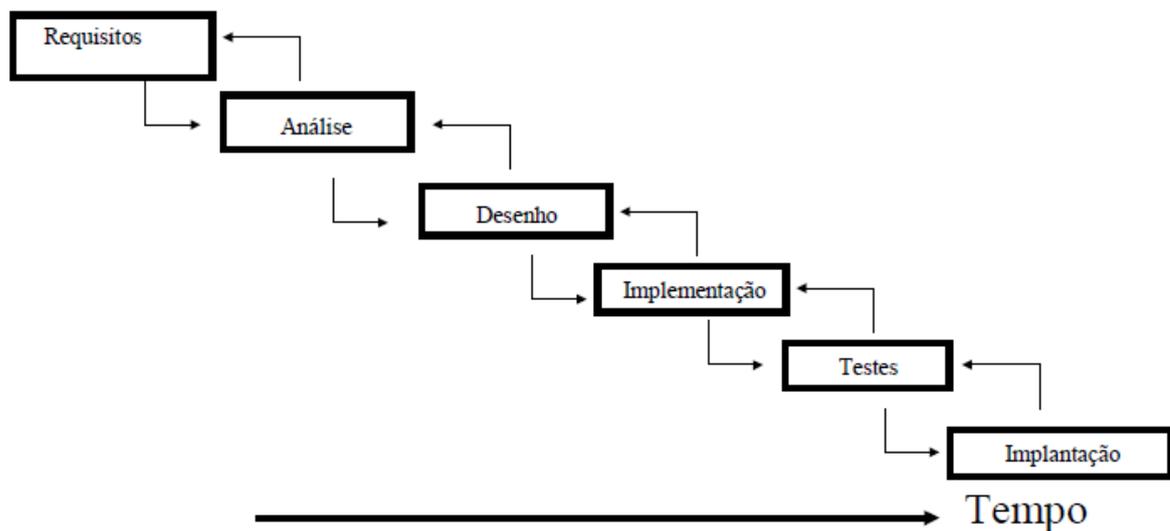
- Modelo em Cascata

Também conhecido como, “Modelo de ciclo de vida clássico”. É um modelo sistemático e sequencial, onde o resultado de uma fase serve de entrada para outra. Este foi o primeiro modelo publicado, e originou-se dos processos mais gerais da engenharia de sistemas (SOMMERVILLE, 2007). Também segundo o autor é indicado para quando os requisitos forem bem definidos e houver pouca probabilidade de uma mudança radical

durante o desenvolvimento. Em outra visão, Paula Filho (2009), afirma se tratar de um processo rígido e burocrático, onde as atividades de requisitos, análise e desenho têm de ser muito bem dominadas, pois não são permitidos erros. Onde o modelo de cascata puro é de baixa visibilidade para o cliente, que só recebe o resultado final do projeto.

Independente destas características, o modelo de vida clássica, trouxe imposições de disciplina, planejamento e gerenciamento, garantindo uma continuidade de projeto mais segura. Sua prática ainda é mais eficaz que qualquer desenvolvimento casual feito sem planejamento.

Figura 1: Modelo de processo em cascata.



Fonte: Paula Filho, 2009.

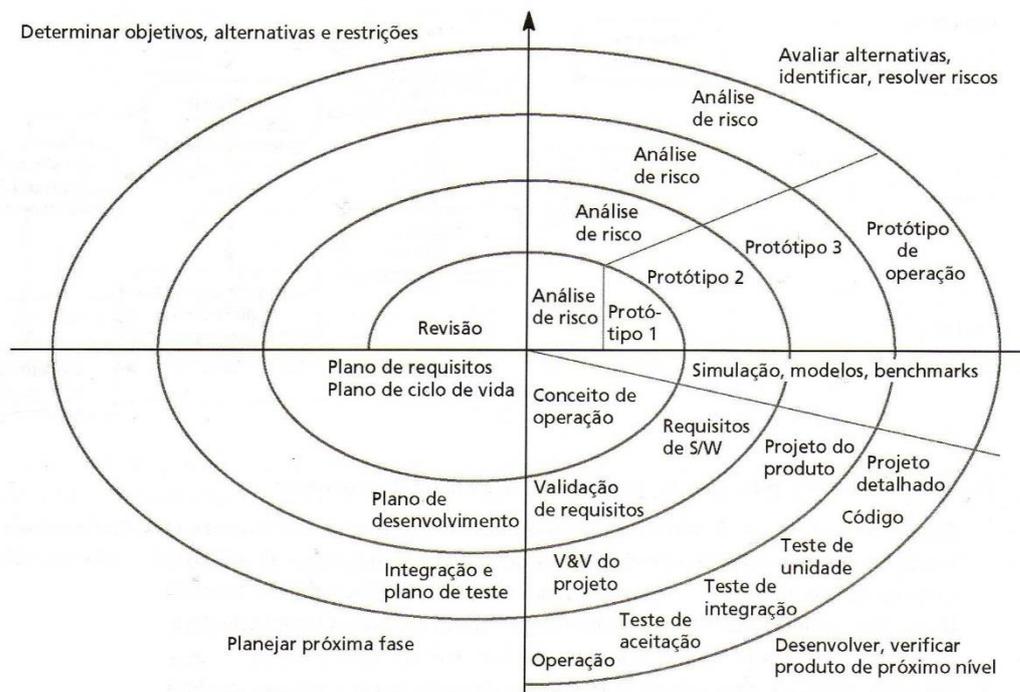
- Modelo Iterativo

Em virtude destes aspectos mutáveis, outros modelos foram surgindo. Neste ponto Pressman (2010) vem corroborar, afirmando que - hoje em dia, o trabalho de software é em ritmo rápido e sujeito a uma torrente sem fim de modificações (de características, funções e conteúdo da informação). O que força ao desenvolvedor e sua equipe a revisarem etapas anteriores e atualizarem outras. Um dos modelos que segue essa ideia é o modelo em espiral.

Proposto originalmente por Boehn em 1988, onde cada *loop* da espiral representa uma fase do processo de software (SOMMERVILLE, 2007). Este modelo traz um novo aspecto de atividade; a análise de riscos, que consiste em uma avaliação da viabilidade e impacto de uma mudança. Sendo assim, o produto é desenvolvido em uma série de iterações.

Cada nova iteração corresponde a uma volta na espiral, e resulta em uma versão do sistema. Isto permite construir produtos em prazos curtos, com novas características e recursos que são agregados na medida em que a experiência descobre sua necessidade.

Figura 2: Modelo de processo em espiral.



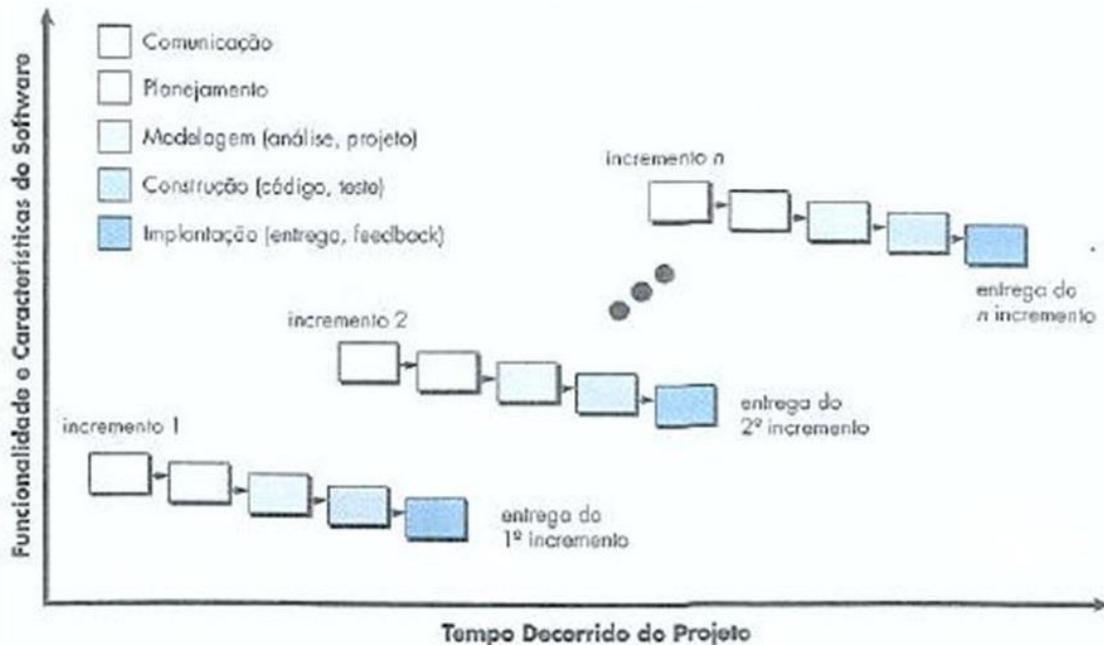
Fonte: Sommerville, 2007.

- Modelo Incremental

Construído também baseado na mutabilidade e acréscimos de requisitos. O modelo incremental combina elementos do modelo em cascata e do espiral. Onde se aplica uma sequência linear para a construção do produto. Cada sequência produz incrementos passíveis de serem entregues (PRESSMAN, 2010).

Em uma abordagem incremental o cliente identifica, em linhas gerais, os requisitos básicos que o sistema deve cumprir. O que irá gerar o produto do primeiro incremento, também conhecido como núcleo do produto. A partir daí este produto passa por uma avaliação e revisão do cliente, e um plano é montado, especificando o acréscimo de nova funcionalidade e/ou alteração do incremento anterior. Este processo se repete até que o produto final seja alcançado.

Figura 3: Modelo de processo incremental.



Fonte: Pressman, 2010.

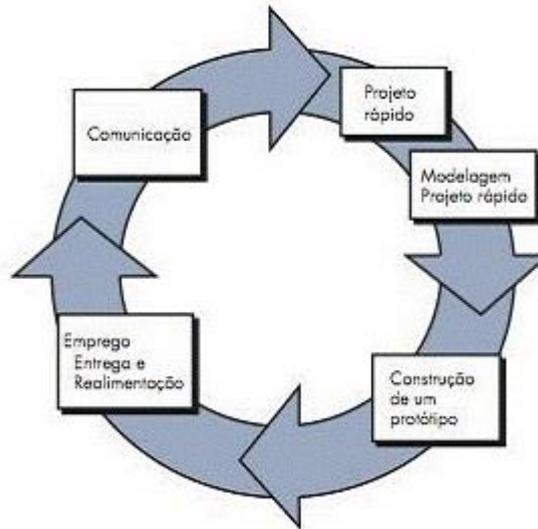
- Modelo Evolucionário

O desenvolvimento evolucionário baseia-se na de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando esse resultado por meio de várias versões até seja desenvolvido um sistema adequado. As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de separadas, com *feed back* rápido que permeia as atividades. (SOMMERVILLE, 2007, p. 45)

Uma abordagem que se enquadra neste paradigma é a de prototipagem. O modelo começa com a comunicação. Depois o engenheiro reúne-se com o cliente e defini os objetivos e identificam as necessidades iniciais. Partindo para uma modelagem rápida e construção de um primeiro protótipo, havendo retorno para o cliente que avalia e testa e devolve um *feed back* ao desenvolvedor (PRESSMAN, 2010). O modelo de prototipação é adequado quando o cliente já definiu os objetivos gerais, mas ainda não identificou detalhadamente os requisitos. Ou quando o desenvolvedor não está seguro, com algum padrão adotado, ou com a integração do sistema, seja com o usuário, ou seja, com o ambiente de implantação. Neste aspecto, Sommerville (2007), orienta que o modelo evolucionário deve ser usado em sistemas de pequeno ou médio porte (até 500 mil linhas) e que para sistemas de grande o porte o modelo

deve-se usar outra metodologia, mas agregando as melhores características do paradigma evolucionário.

Figura 4: Modelo de processo de prototipação.



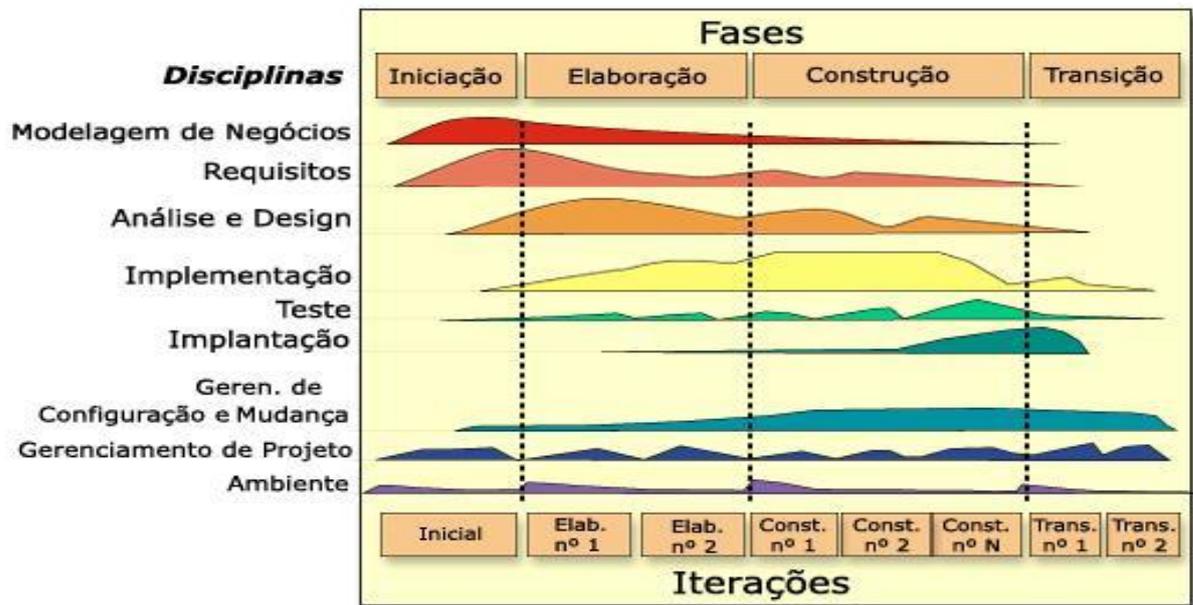
Fonte: Pressman, 2010.

- Metodologia Unificada

Diversos outros modelos e metodologias foram surgindo, com tendências a mesclar aspectos desejáveis dos que já existiam, com o Rational Unified Process (RUP) não foi diferente. Sobre este aspecto, Pessman (2010), argumenta que o processo unificado é uma tentativa de apoiar-se nos melhores recursos e características dos modelos convencionais, mas caracterizá-los de um modo que implemente muitos dos melhores princípios de desenvolvimentos ágeis.

O RUP possui as seguintes características centrais: baseado em caso de usos; é iterativo e incremental; e é centrado na arquitetura, fazendo uso da notação UML (*Unified Modeling Language*). É dividida em fases discretas, voltadas mais ao negócio do projeto, do que aos assuntos técnicos (SOMMERVILLE, 2007). Indicado para desenvolvimentos mais complexos, que necessitam de agilidade, mas requerem um certo grau formalidade.

Figura 5: Modelo de processo RUP.



Fonte: Wikipedia, 2016.

- Metodologias Ágeis

Nem sempre é viável fazer uso de uma abordagem mais estática e rigorosa. O mercado que existe hoje opera em um ambiente sujeito a rápidas mudanças, como novas oportunidades, mudanças na economia, produto e serviços concorrentes, entre outros fatores. Isso restringe o tempo para avaliação e especificações, além de exigir um desenvolvimento ágil de softwares, para atuarem como produtos nestes mercados (SOMMERVILLE, 2010).

Em decorrência dessas questões, culminou em 2001, o chamado “Manifesto para o Desenvolvimento Ágil de Software”. Um documento contendo críticas a algumas práticas e atividades mais defasadas, para o mercado atual, e contendo um conjunto de propostas para melhoria do desenvolvimento de software. De modo geral uma filosofia ágil deve ressaltar quatro tópicos chave: a importância de equipes auto-organizadas que possuem controle sobre o trabalho que executam; comunicação e colaboração entre membros das equipes, profissionais e clientes; reconhecimento de quem modificações representam uma oportunidade; e uma ênfase na entrega rápida de softwares que satisfaçam o cliente (PRESSMAN, 2010).

Depois da veiculação deste documento, surgiram e formalizaram vários modelos, alguns mais conhecidos e aplicados no mercado são: Extreme Programming (XP), Scrum,

Crystal, Adaptive Software Development (ASD), Feature Driven Development (FDD). Mas ainda existem vários outros modelos em uso.

Após a apresentação de algumas abordagens para o ciclo de vida de um software. Pode-se observar que existem diversas opções, no mercado e na literatura, para se aplicar em um projeto. Mas a grande questão é avaliar qual modelo é mais apropriada para o desenvolvimento do sistema, levando em conta: a finalidade do sistema, recursos disponíveis (tecnológicos e humanos), prazos, entre outros aspectos. Pois uma escolha inadequada pode comprometer todo o andamento do produto.

### **2.1.2 Metodologia para o processo de software**

Como afirmado anteriormente, a questão é saber: qual modelo é o mais adequado para o sistema a ser desenvolvido? Mas, e se a solução não for encontrada nos processos convencionais? Uma abordagem possível e mais simples é adaptação de um processo ou até mesmo um aprimoramento, tendo em vista a atualização das tecnologias de ferramentas e comunicação existentes. Uma segunda abordagem, mais complexa, seria a elaboração de uma metodologia própria. Mas, como realizar tal demanda? Em sua obra Sommerville (2007, p.42) já comenta sobre esta questão.

Os processos de software são complexos e, como todos os processos intelectuais e criativos, dependem de julgamento humano. Por causa da necessidade de utilizar o julgamento e a criatividade, a tentativa de automatização dos processos de software têm tido sucesso limitado.

Para resolver esta questão a engenharia de software propõe a padronização de processos. Ainda segundo Sommerville (2007), a padronização é um passo inicial importante na introdução de novos métodos e técnicas de engenharia de software e também nas boas práticas na engenharia de software. Sua prática promove o aprimoramento da comunicação e redução no tempo de treinamento e faz com que o apoio ao processo automatizado seja mais econômico.

Dentro desta perspectiva a engenharia de software aborda dois pontos: como definir um processo de software e como avaliar um processo. O primeiro ponto contempla a estrutura da metodologia, componentes, atividades e artefatos. O segundo diz respeito a qualidade do processo, de acordo com métricas de avaliação já estabelecidas.

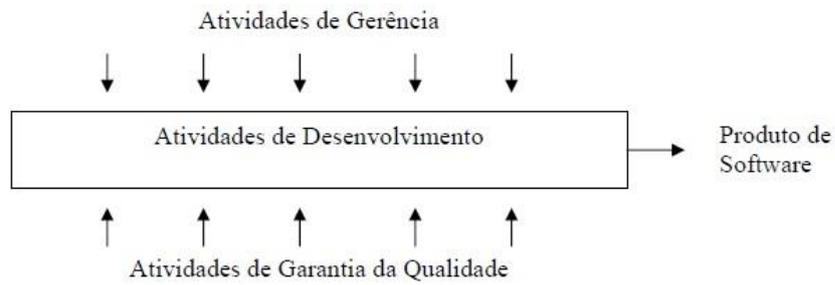
### 2.1.2.1 *Estrutura do processo de software*

De acordo com Falbo (2005), as atividades de um processo de software podem ser classificadas em seus propósitos, da seguinte forma:

- **Atividades (ou Técnicas) de Desenvolvimento (ou de Construção):** são as atividades diretamente relacionadas ao processo de desenvolvimento do software, ou seja, que contribuem diretamente para o desenvolvimento do produto de software a ser entregue ao cliente. São exemplos de atividades de desenvolvimento: especificação e análise de requisitos, projeto e implementação.
- **Atividades de Gerência:** são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto, tais como realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.
- **Atividades de Garantia da Qualidade:** são aquelas relacionadas com a garantia da qualidade do produto em desenvolvimento e do processo de software utilizado, tais como revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento.

É possível afirmar que as atividades de desenvolvimento formam a espinha dorsal do desenvolvimento e, de maneira geral, são realizadas segundo uma ordem estabelecida no planejamento. As atividades de gerência e de controle da qualidade são, muitas vezes, ditas atividades de apoio, pois não estão ligadas diretamente à construção do produto final: o software a ser entregue para o cliente, incluindo toda a documentação necessária. Essas atividades, normalmente, são realizadas ao longo de todo o ciclo de vida, sempre que necessário ou em pontos pré-estabelecidos durante o planejamento, ditos marcos ou pontos de controle (FALBO, 2005). A relação entre esses tipos de atividades é apresentada na figura 6.

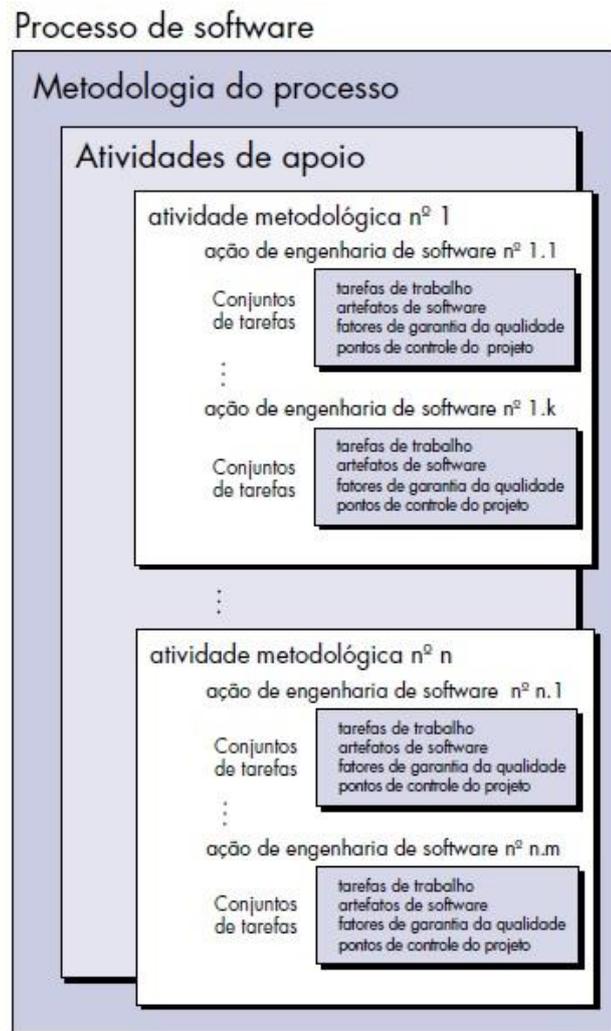
Figura 6: Atividades do processo de software.



Fonte: Falbo, 2005.

Em uma visão mais detalhada Pressman (2011), apresenta em sua obra um esquema contendo a hierarquia metodológica para processos de software.

Figura 7: Metodologia para o processo de software.



Fonte: Pressman, 2011.

De acordo com a figura 7, cada atividade metodológica é composta por um conjunto de ações de engenharia de software. Cada ação é definida por um conjunto de tarefas, as quais identificam as tarefas de trabalho a serem completadas, os artefatos de software que serão produzidos, os fatores de garantia da qualidade que serão exigidos e os marcos utilizados para indicar progresso. Sendo assim, é importante ao menos apresentar alguns conceitos simples relacionados com cada elemento que compõe o processo de software.

**Atividade** – Pode ser compreendida como “mini-projetos”. Pois possuem início e fim de atividades, produzem artefatos definidos e consomem um volume finito de recursos. Representa uma evidência do progresso do desenvolvimento.

**Tarefa** – Ação desempenhada por uma pessoa visando a realização ou monitoramento do projeto. Não representa uma evidência de progresso no desenvolvimento e também consome recursos (esforço, equipamento, financeiro).

**Artefato** – Está vinculado a entrada e saída de uma atividade ou tarefa e pode ser dividida em dois tipos: insumo, elemento de entrada necessário à realização da atividade/tarefa; e produto, resultado (saída) de uma atividade/processo.

Outro aspecto existente, apontado por Pressman (2011), dentro de um processo de software é o **fluxo de software**. O fluxo descreve como são organizadas as atividades metodológicas, bem como as ações e tarefas que ocorrem dentro de cada atividade em relação à sequência e ao tempo. O autor apresenta quatro abordagens para o fluxo de software, são elas.

**Fluxo linear** – executa cada uma das atividades metodológicas em sequência, ou seja, uma atividade só é realizada após a conclusão de sua antecessora;

**Fluxo iterativo** – repete uma ou mais das atividades antes de prosseguir para a seguinte;

**Fluxo evolucionário** – executa as atividades de uma forma “circular”. Cada volta pelas atividades conduz a uma versão mais completa do software;

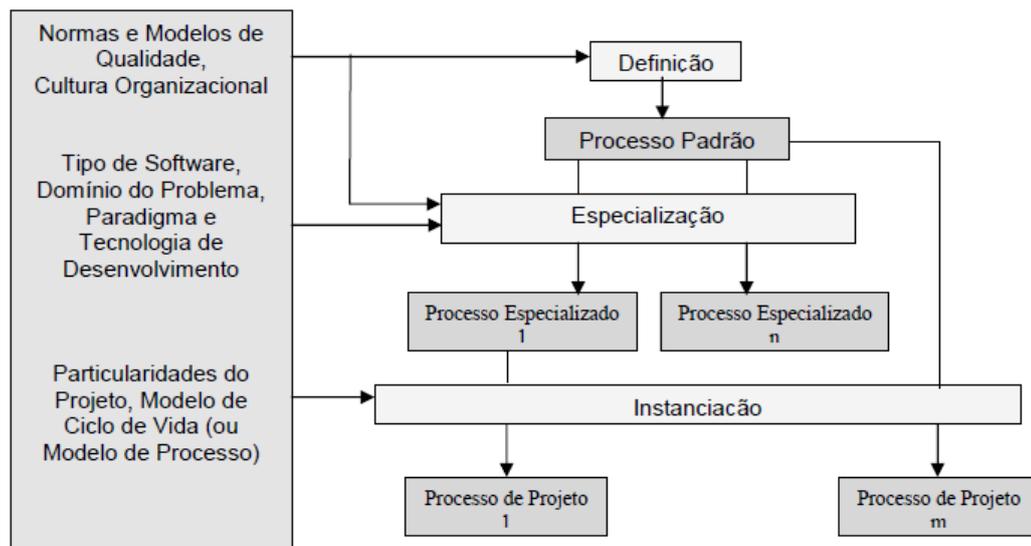
**Fluxo paralelo** – executa uma ou mais atividades em paralelo com outras atividades (por exemplo, a modelagem para um aspecto do software poderia ser executada em paralelo com a construção de outro aspecto do software).

Já Falbo (2005, p.8), traz outra visão sobre a estruturação do processo de software.

De fato, o modelo de definição de processos baseado em processos padrão pode ser estendido para comportar vários níveis. Primeiro, pode-se definir um processo padrão da organização, contendo os ativos de processo que devem fazer parte de **todos** os processos de projeto da organização. Esse

processo padrão pode ser especializado para agregar novos ativos de processo, considerando aspectos, tais como tecnologias de desenvolvimento, paradigmas ou domínios de aplicação. Assim, obtêm-se processos mais completos, que consideram características da especialização desejada. Por fim, a partir de um processo padrão ou de um processo especializado, é possível instanciar um processo de projeto, que será o processo a ser utilizado em um projeto de software específico. Para definir esse processo devem ser consideradas as particularidades de cada projeto. (Grifo do autor)

Figura 8: Modelo para definição de processos em níveis.

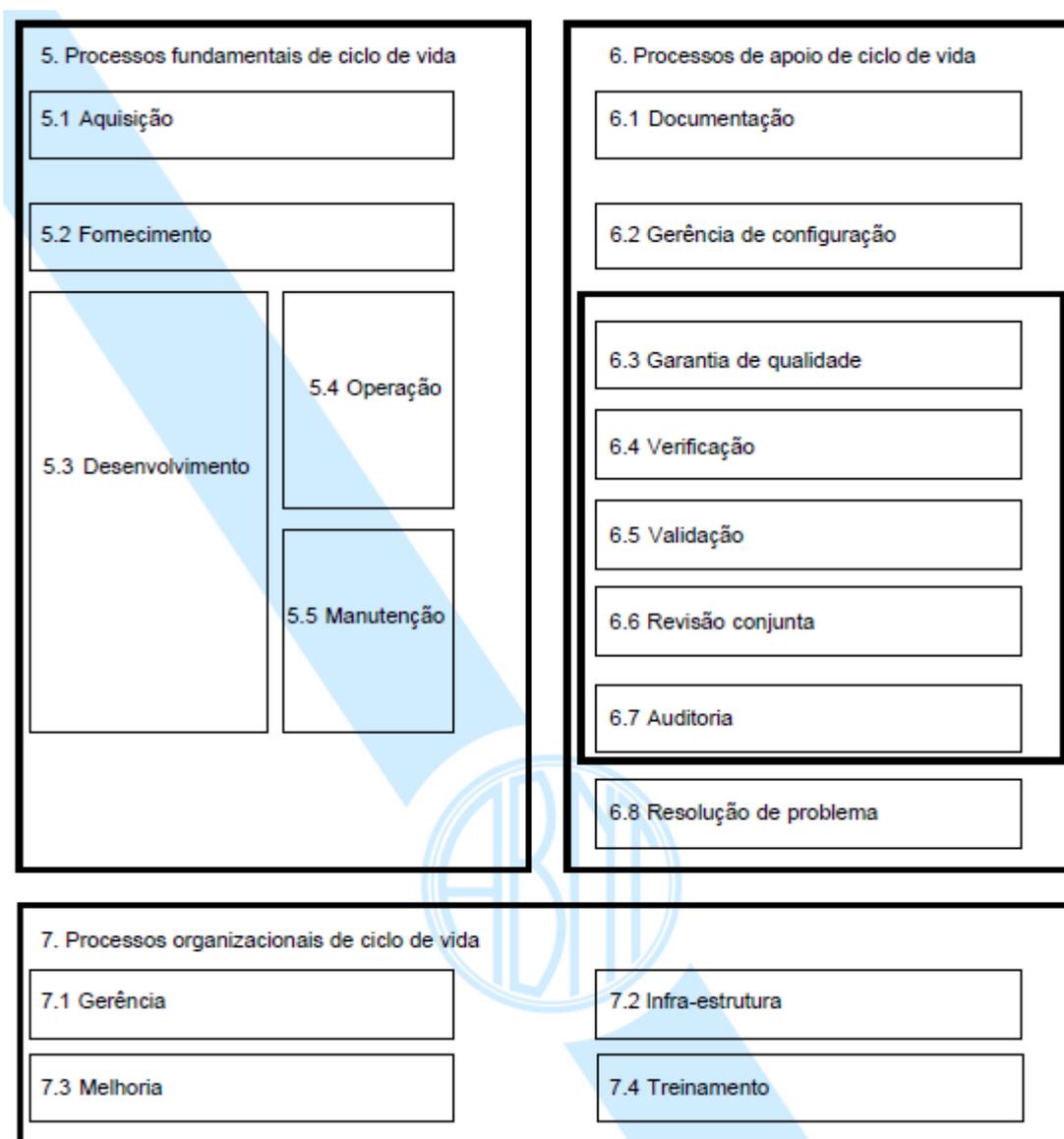


Fonte: Falbo, 2005.

Por fim, para concluir a discussão sobre a estrutura do processo de software é possível fazer referência à norma ISO/IEC 12207 – Processos de ciclo de vida de software. Este modelo de referência foi definido pela *International Organization for Standardization (ISO)* e tem como principal objetivo fornecer uma estrutura única para que o adquirente, desenvolvedor e todos os demais envolvidos com o desenvolvimento de software utilizem uma linguagem comum que é estabelecida na forma de processos bem definidos (LAHOZ e SANT’ANNA, 2016).

Dessa forma a ISO/IEC 12207 é capaz de gerar uma estrutura comum, apresentada na figura 9, flexível e adaptável às necessidades de quem a utiliza. Ela cobre o ciclo de vida do software desde a concepção de ideias até a descontinuação do software e processo de adaptação consiste na supressão de processos, atividades e tarefas não aplicáveis.

Figura 9: Estrutura da norma ISO/IEC 12207.



Fonte: ABNT NBR ISO/IEC 12207.

### 2.1.2.2 Avaliação do processo de software

Uma vez então abordado sobre a estrutura do processo de software é necessário então discorrer sobre as métricas e normas para a avaliação de um processo.

A existência de um processo de software não garante que o software será entregue dentro do prazo, que estará de acordo com as necessidades do cliente ou que apresentará características técnicas que conduzirão a características de qualidade de longo prazo. Os padrões de processo devem ser combinados com uma prática de engenharia de software consistente. Além disso, o próprio processo pode ser avaliado para assegurar que está de

acordo com um conjunto de critérios de processo básicos comprovados como essenciais para uma engenharia de software de sucesso (PRESSMAN, 2011).

Para apoiar esta atividade diversas normas e modelos foram surgindo ao longo das décadas, entre elas estão: CMMI, ISO/IEC 15504 e MPS.Br. Os próximos parágrafos discorrem em uma visão geral sobre cada uma delas.

O primeiro modelo a ser abordado é o Modelo de Maturidade em Capacitação – Integração (CMMI). Ele surgiu no final da década de 90, como proposta do Pentágono para unificação dos modelos de melhoria de software já existentes, como o Modelo de Maturidade em Capacitação (CMM), o modelo de maturidade para Software (SW-CMM), para engenharia de sistema (SE-CMM), para aquisição de software (AS-CMM), entre outros modelos e normas vigentes na época (PAULA FILHO, 2009, p.68).

As práticas recomendadas pelo CMMI estão agrupadas em áreas de processos, que são conjuntos de práticas relacionadas a uma área, que quando executadas satisfazem um conjunto de metas consideradas importantes para a melhoria nessa área. O CMMI trabalha com dois modelos de representação em sua arquitetura, a representação continuada (Figura 10) e a por estágios (Figura 11). Na representação contínua os níveis de capacitação proveem uma ordem recomendada para a abordagem da melhoria. Esta abordagem permite uma indicação de progresso individual por área de processo, além de ser uma abordagem mais flexível, pois os desenvolvedores podem aplicar mais recursos em áreas consideradas mais relevantes. Em contra partida, isso exige um maior conhecimento das interdependências de cada área. Já na apresentação em estágios, as metas de um conjunto de áreas de processo estabelecem um nível de maturidade em cada nível proveem a fundação para os níveis subsequentes (*ibidem*, 2009, p.69). Esta abordagem, segundo Paula Filho, é a mais simples de entender e de implantar, sendo a mais adotada dentro das empresas.

Figura 10: Representação contínua - níveis de capacitação.

Número	Nome (Tradução)	Nome (Original)	Descrição
0	Incompleto	<i>Incomplete</i>	Processo não é executado ou é executado parcialmente. Uma ou mais metas específicas não são atingidas.
1	Executado	<i>Performed</i>	Processo é executado, mas não é instucionalizado. São satisfeitas as metas específicas, mas não as genéricas.
2	Gerido	<i>Managed</i>	Processo é executado com infra-estrutura que o mantém mesmo em situações adversas.
3	Definido	<i>Defined</i>	Processo gerido que é derivado do conjunto de processos padrão da organização, consistente entre projetos e descrito formalmente.
4	Gerido quantitativamente	<i>Quantitatively managed</i>	Processo definido que é gerido com técnicas quantitativas, entendido em termos estatísticos, com objetivos quantitativos de qualidade e desempenho.
5	Otimizante	<i>Optimizing</i>	Processo quantitativamente gerido que está em melhoria constante, com base no entendimento das causas comuns de variação.

Fonte: Paula Filho, 2009.

Figura 11: Representação em estágios - níveis de maturidade.

Número	Nome (Tradução)	Nome (Original)	Descrição
1	Inicial	<i>Performed</i>	Processos informais e <i>ad-hocs</i> , às vezes caóticos.
2	Gerido	<i>Managed</i>	Processos planejados e executados conformes políticas.
3	Definido	<i>Defined</i>	Processos bem caracterizados, entendidos e padronizados.
4	Gerido quantitativamente	<i>Quantitatively managed</i>	Processos geridos em função de objetivos quantitativos de qualidade e desempenho.
5	Otimizante	<i>Optimizing</i>	Processos em melhoria contínua.

Fonte: Paula Filho, 2009.

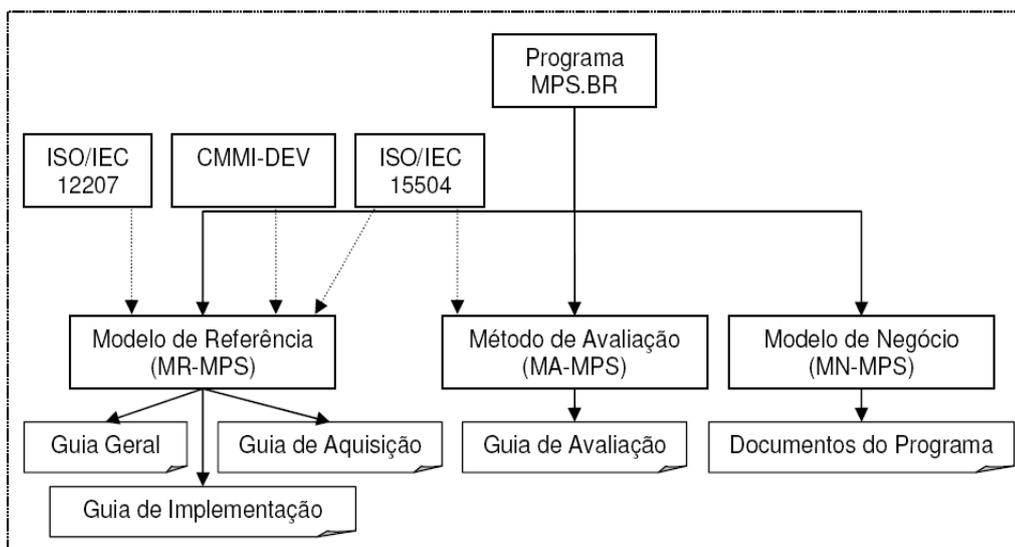
Outra abordagem de avaliação que pode ser empregada é a norma ISO/IEC 15504 – Tecnologia da informação – Avaliação de Processos. Este *framework* organiza e classifica as melhores práticas em duas dimensões: categorias de processo e níveis de capacidade. Cada uma das categorias de processo é detalhada em processos mais específicos, ou subcategorias (Cliente-fornecedor, Engenharia, Projeto, Suporte e Organização) (LAHOZ e SANT’ANNA, 2016). Esta norma visa também avaliar a capacidade da organização em cada processo, permitindo assim sua melhoria. Cada um dos processos deve ser classificado em níveis (incompleto, executado, gerenciado, estabelecido, previsível e otimizado).

Seu modelo de referência serve de base para o processo de avaliação como um conjunto padronizado de processos fundamentais, que orientam para uma boa engenharia de software e pode ser usado de forma complementar com a ISO/IEC 12207.

Por fim, uma alternativa aos padrões internacionais é o Programa de Melhoria de Software Brasileiro (MPS.Br). Ele foi criado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), em 2003. Seu principal foco de atuação são as micro, pequenas e médias empresas de software brasileiro que possuem poucos recursos para melhoria de processos, mas que estão diante de uma necessidade de fazê-lo (SOFTEX, 2012 *apud* ASSUNÇÃO; VASCONCELLOS, 2016).

O MPS.Br foi idealizado com duas metas iniciais: A criação do Modelo de Referência (MR-MPS), Método de avaliação (MA-MPS) e Modelo de Negócio (MN-MPS); e Implementação e avaliação do Modelo MPS dentro de organizações, públicas e privadas, em todas as regiões do país (MPS.BR 2008 *apud* SILVA, 2008). Estas metas podem ser representadas pela figura abaixo.

Figura 12: Componentes do modelo MPS.Br.



Fonte: SOFTEX, 2012 *apud* ASSUNÇÃO; VASCONCELLOS.

O Modelo de Referência representa o modelo ideal para uma organização, isto é, ele indica quais são as metas a que uma organização deve visar para atingir certo grau de qualidade. Ele é descrito através de atributos do processo, que indicam de uma forma global, como o processo deve se comportar; e os resultados esperados, que indicam os resultados que cada processo específico devem produzir. Esses atributos e resultados evoluem conforme se alcança novos níveis de maturidade. O essencial do MR-MPS é descrito no Guia Geral, que é apoiado pelo Guia de Aquisição e Guia de Implementação (SILVA, 2008).

O Modelo de Avaliação institucionaliza a forma como se deve proceder a avaliação de uma empresa, verificando assim se a mesma está de acordo com as normas especificadas pelo MR-MPS. Como resultado desse processo, a empresa terá sua qualidade certificada. Ele é descrito pelo Guia de Avaliação (*ibidem*, 2008).

O Modelo de Negócio, descrito pelos Documentos do Programa, explicita: as regras de negócio para a implementação do MR-MPS pelas Instituições Implementadoras; a avaliação MA-MPS feita pelas Instituições Avaliadoras; a implementação do MR-MPS realizado pelas organizações de grupos de empresas; e, por fim, a avaliação realizada pelas Instituições Organizadoras de Grupos de Empresas (IOGE) (*ibidem*, 2008).

Ainda existem no mercado mundial, diversos outros modelos de avaliação para processos de software, mas através dos modelos ora apresentados já é possível ter uma noção da importância da maturidade nas atividades de avaliação de processos de software.

## 2.2 AMBIENTE VIRTUAL

O primeiro passo para entender sobre ambientes virtuais (AV) é entender sobre realidade virtual. Do que dita isso, Tori, Kirner e Siscoutto (2006) definem:

A Realidade Virtual é, antes de tudo, uma “interface avançada do usuário” para acessar aplicações executadas no computador, tendo como características a visualização de, e movimentações em ambientes tridimensionais em tempo real e a interação com elementos desse ambiente.

Neste espaço o usuário atua de maneira multissensorial, que é o ponto chave da RV, aliado aos aspectos da imersão, interação e envolvimento (RODRIGUES; PORTO, 2013).

Por imersão se tem a ideia da medida (métrica) que o usuário está inserido dentro do espaço tridimensional, ou seja, o quanto ele está imerso. Todavia, este aspecto está diretamente relacionado com as tecnologias que transmitem ao usuário essa sensação. A interação está associada à capacidade do computador detectar as entradas do usuário e modificar em tempo real o mundo virtual e as ações sobre ele. Por fim o aspecto do envolvimento está ligado ao grau de participação e comprometimento do usuário na atividade realizada dentro do ambiente.

Com todos estes pontos ressaltados anteriormente, pode-se perceber que o desenvolvimento de um AV está ligado ao uso de tecnologias multimídias/multissensoriais para suas aplicações. O conceito de multimídia reflete à integração controlada por computador, de textos, imagens, vídeos, animações, áudio e outras mídias, que possam representar armazenar, transmitir e processar informações de forma digital (MARSHAL, 2001 *apud* TORI; KIRNER; SISCOOTTO, 2006). Tendo como base esta definição, pode-se perceber certo impasse, pois o uso demasiado da multimídia limita a experiência do usuário a dados pré-processados, o que vai de encontro ao maior interesse da realidade virtual, que é o trabalho de imersão em tempo real.

### 2.2.1 Tipos de Realidade Virtual

É justamente neste ponto de conflito que a RV se divide em dois tipos: a imersiva e não-imersiva. A RV é imersiva, quando o usuário é transportado predominantemente para o domínio da aplicação, através de dispositivos sensoriais (capacete ou caverna, por exemplo) que capturam seus movimento e comportamentos. A RV é classificada como não-imersiva quando, o usuário é transportado parcialmente ao mundo virtual, através de uma janela (monitor ou projeção, por exemplo), mas continua a sentir-se predominantemente no mundo real (TORI; KIRNER; SISCOOTTO, 2006).

Ainda sobre as classificações, Netto, Machado e Oliveira (2016), apresentam que um sistema de RV pode proporcionar uma sessão sob três formas diferentes: passiva, exploratória ou interativa.

Uma sessão passiva proporciona ao usuário uma exploração do ambiente automática e sem interferência. A rota e os pontos de observação são explícitos e controlados exclusivamente pelo *software*. O usuário não tem controle algum, exceto talvez, para sair da sessão. Uma sessão exploratória proporciona uma exploração do ambiente dirigida pelo usuário. O participante pode escolher a rota e os pontos de observação, mas não pode interagir de outra forma com entidades contidas na cena. Já uma sessão interativa proporciona uma exploração do ambiente dirigida pelo usuário e, além disso, as entidades virtuais do ambiente respondem e reagem às ações do participante. Por exemplo, se o usuário move o ponto de observação em direção à porta, esta pode parecer abrir-se, permitindo ao participante passar por ela.

### 2.2.2 Aplicações da Realidade Virtual

Muito se evoluiu desde a concepção do primeiro dispositivo de imersão, em 1950, e da formalização do conceito de Realidade Virtual, definido em 1980, por Jaron Lenier (TORI; KIRNER; SISCOOTTO, 2006). De forma que se extrapolaram os limites dos laboratórios de pesquisa e integraram-se as atividades cotidianas. As áreas de aplicação dos sistemas de realidade virtual vão desde a visualização de dados até a operação de cirurgias em modelos de realidade aumentada.

A figura 13 trata da reconstrução de ambientes arquitetônicos. Esta aplicação viabiliza a visualização e navegação dentro de ambientes já danificados, ou até extintos, pelo tempo.

Figura 13: Aplicação com reconstrução arquitetônica.



Fonte: Rosa Junior, 2003.

A figura 14 traz um exemplo da utilização de ambientes virtuais na indústria do entretenimento. Esta é uma área que está em constante crescimento, pois é grande tendência de criação de jogos e simuladores com presença de realidade virtual ou realidade aumentada.

Figura 14: Aplicação em entretenimento com jogos.



Fonte: Ribeiro et al, 2011.

Na figura 15 encontra-se um modelo tridimensional de realidade aumentada em uma aplicação de cirurgia médica. Este tipo de aplicação realiza um mapeamento de realidade aumentada e o próprio paciente se torna o ambiente virtual.

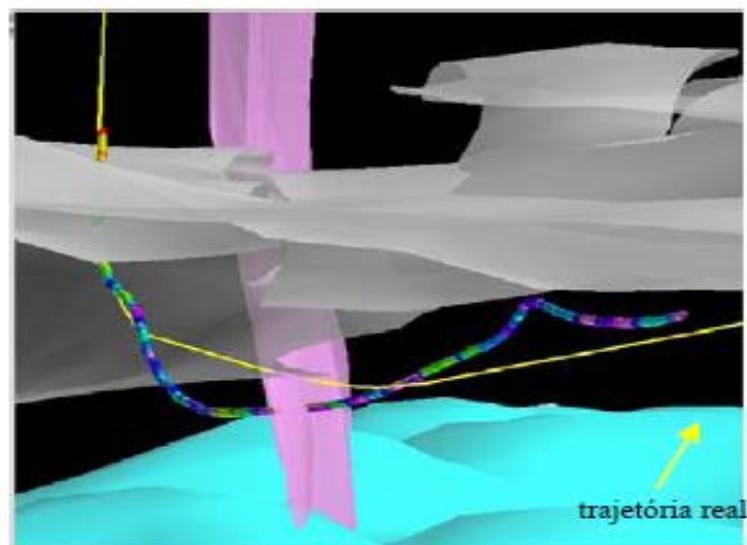
Figura 15: Aplicação na medicina com realidade aumentada.



Fonte: Tori; Kirner; Siscoutto, 2006.

A visualização de cenários, para as indústrias, por meio de ambientes virtuais, auxilia a tomada de decisões, além de minimizar os custos dentro das empresas. A figura 16 apresenta esta aplicação.

Figura 16: Aplicação em visualização industrial.



Fonte: Netto; Machado; Oliveira, 2002.

Outra aplicação possível é criação de ambientes virtuais imersivos para simulação de treinamento militar (Figura 17). Este tipo de prática reduz o gasto com recursos e é capaz de retratar várias situações de ambientes diferentes.

Figura 17: Aplicação em simulação no treinamento militar.



Fonte: Tori; Kirner; Siscoutto, 2006.

Por fim, pode-se concluir que muitas terminologias foram surgindo para definir o conceito de inserção de pessoas em ambientes sintéticos e seus modelos de interação, foram definidos termos como ambiente virtual, espaço cibernético, virtualização, presença virtual, entre outros. Alguns termos se confundem e outros possuem definições diferenciadas, mas mantendo uma congruência (ROSA JUNIOR, 2003).

Um AV tridimensional, além de representar a informação de forma mais realística, facilita a compreensão do comportamento humano e tem se revelado um desafio permanente ao próprio homem em função de sua complexidade e diversidade (OSÓRIO *et al*, 2015).

## 2.3 TRABALHOS RELACIONADOS

Diversos trabalhos podem ser relacionados ao desenvolvimento de ambientes virtuais, ou de realidade virtual, alguns tratando de abordagens mais genéricas, outros direcionando a aplicações mais específicas, como para ambientes de rede, indústrias, ou para jogos. Este tópico reuni a descrição de alguns trabalhos que serviram de base para a metodologia proposta nesta monografia. Apresenta-se abaixo uma lista nomeada pelos títulos das obras.

### a) Fundamentos e tecnologia de realidade Virtual e Aumentada

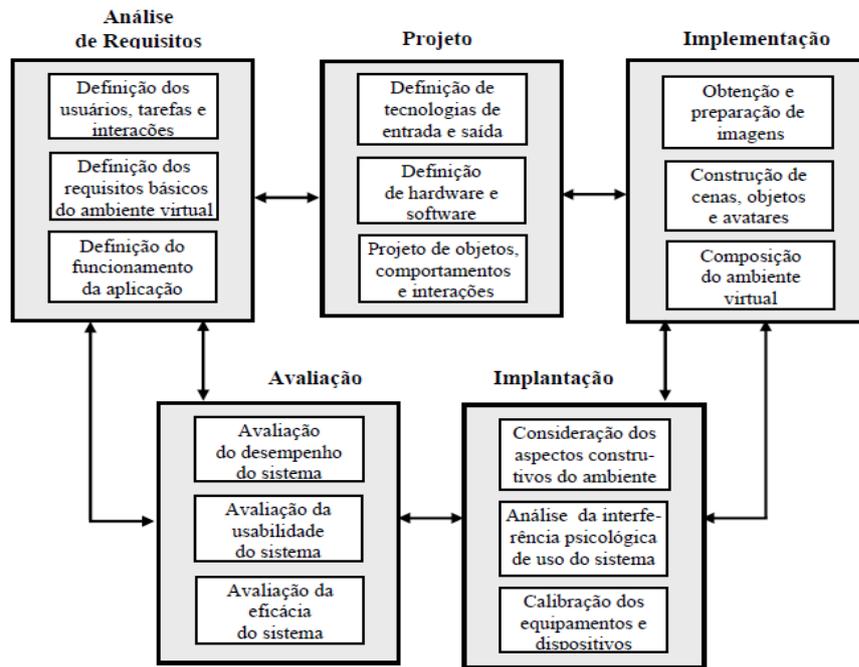
Esta obra é um livro apresentado no lançamento do pré-simpósio do *VII Symposium on Virtual Reality* e reuni a contribuição de diversos estudiosos. De maneira geral, este livro trata sobre vários aspectos que permeiam o estudo da realidade virtual e da realidade aumentada, mas é na segunda parte do livro, intitulada “Ambientes Virtuais”, na sessão oito, que os autores Tori, Kiner e Siscoutto tratam dos processos de desenvolvimento de sistemas de realidade virtual.

Os autores ainda apresentam uma análise sobre os modelos de processos mais utilizados e suas vantagens e desvantagens na aplicação para ambientes virtuais. Sintetizando sobre o desenvolvimento de SRVs é possível encontrar a seguinte afirmação apresentada na obra:

O Desenvolvimento de Sistemas de Realidade Virtual (SRV) teve suas origens no desenvolvimento de sistemas de software, com a utilização das metodologias tradicionais da engenharia de software [McConnell, 1996], adaptadas à criação de sistemas multimídia, com a incorporação de recursos para análise e projeto de sistemas distribuídos e de tempo real. Além disso, questões de criação e de produtos na indústria cinematográfica têm contribuído para o desenvolvimento de SRV, principalmente no que tange à criação do mundo virtual. (TORI, KIRNER, SISCOOTTO, 2006, p.109-110; McConnell, (1996) *apud ibidem*).

Apoiando-se então nesta perspectiva, os autores apresentam um modelo de processo (Figura 18) que agrega características da prototipagem, aliadas ao modelo iterativo. Este modelo busca atender algumas características: facilitar a participação dos usuários e produzir um sistema de fácil utilização e uma manutenção, dentro de um período apropriado e com custos aceitáveis.

Figura 18: Modelo de processo de desenvolvimento de SRVs.



Fonte: Tori; Kirner; Siscoutto, 2006.

## b) Uma Proposta para o Desenvolvimento Ágil de Ambientes Virtuais

Diferente do trabalho apresentado anteriormente, que aborda mais as chamadas ‘metodologias clássicas’, neste artigo Mattioli *et al* focam apenas nas metodologias ágeis. A princípio, esclarecem sobre o que é o Manifesto Ágil e seu impacto sobre a área da engenharia de software. Discorrem ainda sobre diferentes metodologias, como o XP, SCRUM, a metodologia de Tori, Kiner e Siscoutto, ora já apresentada e outra proposta por Kim (2005) - na obra *Designing virtual reality systems: the structured approach* – ratificando as boas características e suas implicações para o desenvolvimento de SRVs.

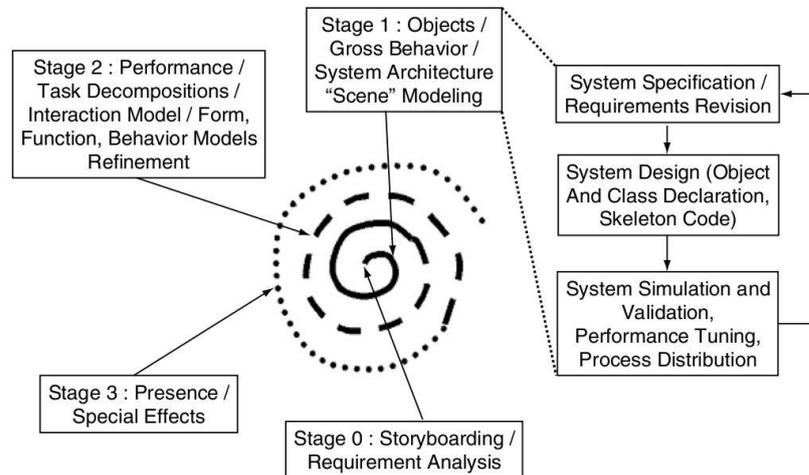
Posteriormente os autores abordam a metodologia do SPIKE, para a arquitetura do sistema, que consiste em uma análise da viabilidade de implementação dos novos requisitos, obtidos a partir das histórias de usuário e dos *storyboards*. Defendendo que esta prática tem como objetivo reduzir os riscos presentes em áreas desconhecidas nos domínios da aplicação, tecnologia e do sistema propriamente dito (MATTIOLI *et al*, 2016).

E por fim, concluem:

Processos ágeis de desenvolvimento se apresentam como uma importante alternativa às metodologias clássicas aplicadas atualmente ao desenvolvimento de sistemas de Realidade Virtual. Em especial, o caráter iterativo, a boa aceitação de mudanças nos requisitos e a importância dada ao

*feedback* dos clientes nestes processos representam os pontos fortes de sua aplicação, ao mesmo tempo em que são compatíveis com as características dos projetos de Realidade Virtual. (*Ibidem*, 2015).

Figura 19: Modelo de processo de Kim.



Fonte: Kim *apud* Mattioli *et al*, 2016.

### c) Desenvolvimento de um ambiente para visualização tridimensional da dinâmica de *risers*

Esta obra trata-se uma dissertação apresentada por Bernades Júnior (2004), em que o autor se fundamenta nos princípios da realidade virtual e de ambientes virtuais, da computação gráfica e visualização científica, e no estudo dos *risers* (estrutura que liga os poço de exploração submersos às plataformas ou navios na superfície) e exploração marítima do petróleo a fim gerar uma solução para o desenvolvimento e implementação de um protótipo de ambiente virtual interativo para a visualização tridimensional dos *risers*.

Para tanto o autor utilizou-se da metodologia dos Processos Unificados (UP), justificando-se com o seguinte argumento.

Como pesquisa científica que é, este projeto deve contar fortemente com duas características: a boa documentação do projeto bem como mecanismos que auxiliem em sua repetibilidade, ou seja, que permitam que outros pesquisadores possam alcançar resultados semelhantes seguindo aproximadamente os mesmos passos trilhados aqui.

Para conseguir estas características num projeto que trata, em grande parte, de desenvolvimento de software, optou-se por utilizar uma metodologia formal para este desenvolvimento, que norteie os passos do processo e que esteja bem clara tanto para o desenvolvedor quanto para o leitor (BERNADES JÚNIOR, 2004).

Esta obra contribui com a fundamentação deste trabalho à medida que se aproxima do conteúdo abordado nela, corroborando e fornecendo uma visão do processo de desenvolvimento e aplicação de um ambiente virtual.

### 2.3.1 Contribuições correlacionadas

Além das áreas abordadas na fundamentação e dos trabalhos mencionado anteriormente, três trabalhos correlacionados ao tema desta monografia foram de importante contribuição ao que tange alguns aspectos específicos para a elaboração da metodologia proposta. O primeiro trata da especificação de requisitos para AVs e os outros dois das atividades realizadas no desenvolvimento de jogos (*game design*). Apresenta-se abaixo uma lista nomeada pelos títulos das obras.

#### a) *Contribuição à Engenharia de Requisitos de Ambientes Virtuais*

O primeiro trabalho é um artigo produzido por Kirner e Salvador (2015). Ele trata da realidade virtual, conceitos e ferramentas; da engenharia de requisitos, definições e atividades; e por fim estabelece uma ligação entre as duas áreas, culminando na especificidade que existe nos requisitos de um ambiente virtual, sejam eles funcionais ou não.

Kirner e Salvador (2015) apresentam aos seguintes resultados sobre os requisitos funcionais:

Quando se aborda ambientes virtuais, é necessário considerar três enfoques possíveis,

que impactam de maneira distinta a especificação funcional do sistema:

- Enfoque 1. Quando o ambiente virtual reproduzirá tarefas e ações executadas por usuários em um ambiente real e se pretende retratar esta situação como ela é, ou seja, com realismo.
- Enfoque 2. Quando os usuários desempenham determinadas tarefas no mundo real, mas se decide que o ambiente virtual reproduzirá tais tarefas de forma diferente ao que ocorre na realidade.
- Enfoque 3. Quando o ambiente virtual pretendido compreenderá a realização de tarefas e interações que não são desempenhadas no mundo real.

Sobre os requisitos não funcionais, os autores continuam:

[...] os ambientes virtuais agregam peculiaridades que demandam a consecução de requisitos não funcionais adicionais. Neste sentido, espera-se que um ambiente virtual seja:

- Sintético. Significa que o ambiente virtual é gerado em tempo-real pelo sistema computacional, o que não ocorre com sistemas multimídia, nos quais as apresentações são gravadas previamente, antes de serem executadas.
- Tri-dimensional. Significa que o ambiente “físico” que circunda o usuário deve ser representado tridimensionalmente, e que o usuário pode andar por este ambiente.
- Multi-sensorial. Significa que o ambiente deve incluir recursos que estimulem diferentes sentidos humanos, como visão, audição, tato, senso de espaço, profundidade, etc.
- Imersivo. Significa que, além de visualizar as cenas e ouvir os sons, o usuário deve ter a impressão de que está realmente dentro do ambiente virtual.
- Interativo. Tal requisito possibilita a detecção de estímulos (solicitações) enviadas pelo usuário e, instantaneamente, promover alterações/ações em cenas e objetos do ambiente virtual.
- Realístico. Refere-se ao nível de precisão e conformidade que o ambiente virtual apresenta em relação à representação do mundo real enfocado.
- Com presença. Caracteriza-se por um sentido subjetivo de que o usuário está fisicamente no ambiente

*b) Desenvolvimento de Concept Art para Personagens e Video Game Design:*

*Uma análise da estética conceitual do entretenimento digital*

Essas duas obras propostas por Takahashi & Andreo (2015) e Tocchio (2007), respectivamente, contribuíram na fundamentação da atividade de projeto denominada ‘Projeto Audiovisual’ – que será melhor explicada na subseção ‘Projetando um ambiente virtual tridimensional’.

Tal qual o desenvolvimento de SRVs o desenvolvimento de jogos digitais possui uma metodologia própria, que entrelaça os aspectos da engenharia de software com a criação artística (design), gerando documentos característicos como: o *game concept*, *game design* e *biblie design*. Por conta deste entrelace é viável estabelecer uma correlação entre o desenvolvimento de AVs 3D e de jogos. Já que todos os dois correspondem a softwares que primam pela interatividade e imersão do usuário. Além disso, Tocchi (2007, p. 66, p. 67), em sua obra, declara sobre a imersão em AVs no futuro dos jogos.

No que se refere às possibilidades de refinamento de técnicas para o aprimoramento dos jogos, para que o jogador tenha a real sensação de imersão, um longo caminho ainda está por ser trilhado. [...] O princípio de imersão mostra-se de fundamental relevância neste sentido, uma vez que oferece possibilidades reais de diálogo fluído com novas realidades e mostra potencial não apenas no campo fictício, mas também na qualidade de solução para muitas questões do cotidiano, com aplicações na área de design residencial e de mobiliário, turismo e medicina, entre outros.

Partindo desta visão é possível destacar o processo de design chamado de *Art Concept* (Arte Conceitual), sobre sua definição, influência e impacto no projeto, Takahashi e Andreo afirmam:

O *concept art*, pode ser considerado como representações visuais que buscam materialização de conceitos idealizados para a indústria de entretenimento (como filmes de animação, ação livre e jogos eletrônicos), de forma a auxiliar de forma relevante no desenvolvimento dos projetos, pois acelera e torna mais coesa a produção, tornando possível incorporar, alterar e interagir visualmente com as ideias propostas.

Sua utilização, além de definir e estabelecer os primeiros passos do projeto possui grande potencial para enriquecer a apresentação do projeto para possíveis investidores, por se tratar de algo concreto e que torna as ideias mais palpáveis e convincentes.

Completando a influência do desenvolvimento de jogos digitais neste trabalho, um segundo aspecto precisa ser abordado: a sonorização.

“Também a sonorização mostra-se fundamental para o desenvolvimento de uma ambientação imersiva nos jogos e é elemento imprescindível para a condução narrativa de certos games.” (TOCCHI, 2007, p.44).

Essa mesma afirmativa é aplicável ao desenvolvimento de AVs 3D e somada aos aspectos visuais conseguem contemplar os aspectos, imersivos, realísticos e multissensorial que são natos a um ambiente virtual.

### 3 METODOLOGIA

#### 3.1 TIPO DA PESQUISA QUANTO AOS OBJETIVOS, ABORDAGEM E FINS

Este trabalho segue como um resumo de assuntos e foi direcionado, em seu objetivo, na linha da pesquisa exploratória, já que se faz necessário delimitar os objetivos e métodos usados na criação do processo de software, para posteriori explicitar os resultados obtidos, gerando assim maior familiaridade com o caso abordado (GIL, 2002).

Para tanto foi realizada uma abordagem qualitativa, já que esta ressalta aspectos indutivos, onde o pesquisador desenvolve conceitos, ideias e entendimentos a partir de padrões encontrados (MORESI, 2003). O que se adequa à aplicação de um estudo de caso e ao propósito de uma monografia, que é uma experiência de trabalho por meio da qual o aluno aprende a organizar uma bibliografia sobre determinado assunto, bem como suas próprias ideias, de modo a apresentá-las, por escrito, de forma coerente, inteligível e encadeada (LOUREIRO E CAMPOS, 1999 *apud* PAGNEZ, 2007).

#### 3.2 INSTRUMENTOS E MATERIAIS DA PESQUISA

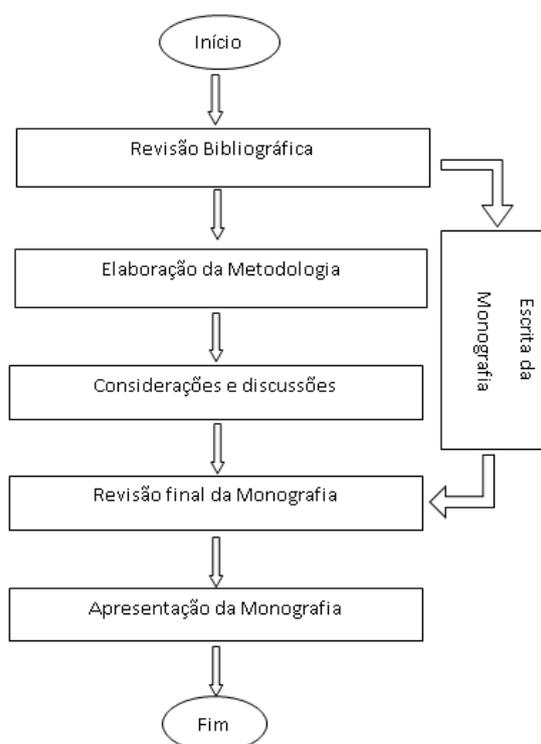
Dentre as fontes de pesquisa, foram utilizados livros das diversas áreas abordadas durante a execução do trabalho, consultas a artigos científicos e notas de aulas relacionadas ao tema, dissertações acadêmicas relacionadas ao tema proposto. Foram feitas buscas também em normas de padronização técnicas, tanto para este modelo de documento (monografia), quanto para a padronização de processos de software.

Quanto ao aspecto instrumental, as ferramentas utilizadas foram: o Inkscape, software livre de código aberto, utilizado para produção de imagens e editoração vetorial. E o Astah Community, software proprietário *shareware* (possui versão livre, mas com restrições) para plataforma Windows. O Astah é uma ferramenta CASE (*Computer Aided Software Engineering*) para o desenvolvimento de diferentes tipos de diagramas UML, organogramas e fluxogramas.

#### 3.3 TAREFAS DO DESENVOLVIMENTO DO TRABALHO

Esta subseção traz de forma simplificada uma descrição da sequência de atividades realizadas pelo autor durante a escrita deste trabalho.

Figura 20: Atividades da monografia.



Fonte: Elaborado pelo autor, 2016.

- Revisão Bibliográfica

Esta tarefa consiste no levantamento bibliográfico em livros, artigos, teses e dissertações acerca dos aspectos que foram definidos como fundamentos teóricos desta monografia, tanto em seus conceitos e definições, como em suas aplicações.

- Elaboração da Metodologia

A metodologia proposta pelo autor foi elaborada a partir do estudo dos modelos de processo de software encontrados durante a revisão bibliográfica, bem como em orientações existentes no mercado para criação de padrões de processo. Buscou-se avaliar e agregar os melhores aspectos, segundo a visão do autor, que contemplassem o desenvolvimento para ambientes virtuais tridimensionais. Esta atividade compreende a primeira parte prática desenvolvida pelo autor, já que, se pode levar em conta que ela tem como artefato um modelo metodológico.

- Considerações e discussões

Esta etapa corresponde à algumas ponderações acerca do MAV3D, suas vantagens, desvantagem, e diferencias em relações a outras metodologias existentes no mercado. Traz também algumas pontuações sobre as dificuldades encontradas na elaboração do MAV3D.

- Revisão final da Monografia

Conclusão e revisão final para submissão para a defesa da monografia. Feita em conjunto com o orientador.

- Apresentação da Monografia

Esta última etapa consiste na apresentação e defesa da monografia, frente à banca examinadora.

## 4 MAV3D - METODOLOGIA PARA AMBIENTES VIRTUAIS TRIDIMENSIONAIS

De acordo com o que já foi apresentado na seção do referencial teórico, do que trata dos modelos de processos de software – tanto dos modelos já existentes, quanto da criação de padrões de processo - e das características inerentes a um ambiente virtual. Percebe-se a real necessidade de uma abordagem adequada aos ambientes virtuais tridimensionais, pois as metodologias clássicas e ágeis, não são capazes de realizar de forma satisfatória esta demanda. A fim de contemplar esta questão, o autor propõe a criação de uma metodologia própria, o MAV3D. As próximas subseções descrevem o propósito geral, princípios, visões, elementos e o fluxo de atividades desta metodologia.

### 4.1 VISÃO GERAL E PRINCÍPIOS

A metodologia proposta agrega características do RUP, do modelo proposto por Tori, Kiner e Siscoutto e do desenvolvimento de jogos. Gerando um modelo iterativo e incremental que visa um desenvolvimento mais rápido, mas preservando certa formalidade em seu processo. Desta forma, o MAV3D pode ser enquadrado dentro das metodologias clássicas.

De forma geral, o processo integra quatro fases que podem ser concebidas a partir de um nível de abstração mais alto, são elas: Concepção, Estruturação, Desenvolvimento e Entrega. Por sua vez, cada fase possui uma ou mais disciplinas, e estas se especializam em tarefas distintas.

Quanto à aplicação desta metodologia, alguns princípios foram adotados a fim de reduzir os riscos e melhorar a eficiência e qualidade durante o desenvolvimento do sistema. Estes princípios são:

Comunicação – É um fator chave que difere a produção de software. Uma boa comunicação entre os membros da equipe e entre o cliente garante maior fluidez durante o processo de desenvolvimento.

Modelagem visual – A metodologia é pautada no uso da notação UML para representar visões e soluções para o sistema. Já que relatórios e documentos com linguagens técnicas pode dificultar o entendimento de outros componentes da equipe.

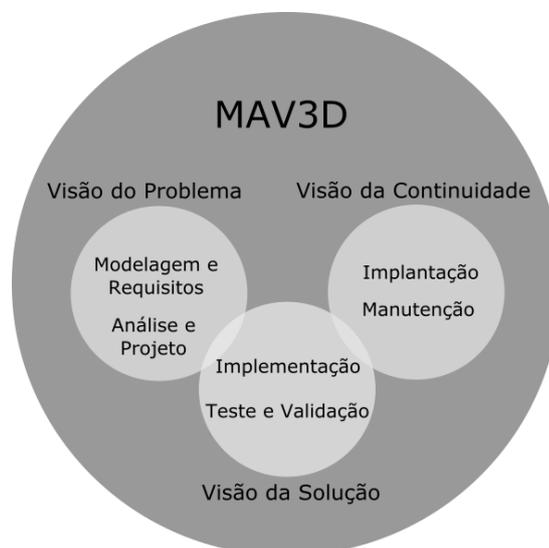
Versionamento – Este aspecto garante o foco da equipe em uma meta, além de permitir uma versão apresentável ao cliente. É a partir dela que se tem uma visão mais clara do andamento do projeto, tanto por conta dos testes das funcionalidades e integrações, quanto por conta do *feedback* apresentado pelo cliente.

Flexibilidade – Este aspecto se refere à transição das fases e disciplinas. Apesar de se tratar de uma metodologia iterativa, nem sempre é viável ou necessária a aplicação de todas as disciplinas, bem como, às vezes é necessário o retrocesso a uma etapa anterior em caso de falha.

Fundamentado no projeto – esta metodologia apoia-se na ideia de que um bom planejamento e projeto evitam futuras falhas e inconsistências no processo de desenvolvimento, além de garantir uma melhor modularidade, reutilização de componentes e escalabilidade para o sistema. Além dos projetos arquitetural, de dados, procedimental e de interface do usuário, esta metodologia adota a realização do projeto audiovisual, que será melhor explicado adiante.

Uma visão possível sobre o MAV3D é quanto aos domínios contemplados por cada fase. Este tipo de abordagem facilita a compreensão do funcionamento da metodologia onde atua cada fase, a figura 21 destaca essas visões. As fases de concepção e estruturação tratam do domínio do problema. Ou seja, são responsáveis por interpretar e entender o ambiente e os fatores externos e traduzi-los de forma que os profissionais envolvidos no processo de desenvolvimento possam construir o software. Uma vez entendido o problema, entra-se no domínio da solução, contemplado pela fase de desenvolvimento, que tem por objetivo implementar e testar uma versão de software aceitável para ser entregue ao cliente. A última visão constitui o domínio da continuidade contemplado pela fase de entrega, esta visão lida quanto aos aspectos da melhoria e da continuidade do sistema como todo, não somente do software, mas também a preparação das tecnologias e o treinamento de pessoas.

Figura 21: Visões dos domínios.

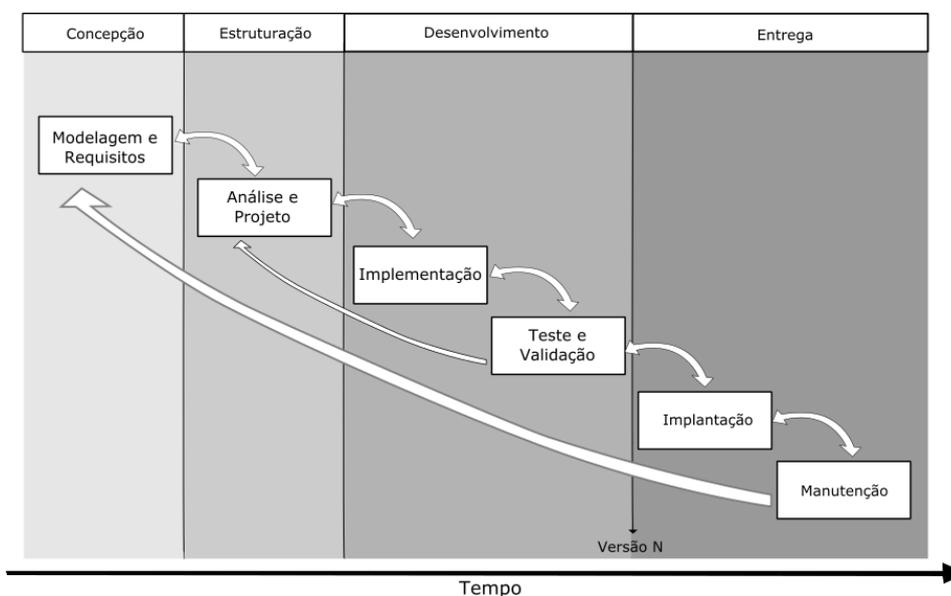


Fonte: Elaborado pelo autor, 2016.

## 4.2 HIERARQUIA DOS COMPONENTES DA METODOLOGIA

Como foi mencionada anteriormente, a metodologia segue uma linha de hierarquia onde cada componente cumpre uma camada em um nível de abstração. É aceitável também enquadrar esta hierarquia dentro da proposta apresentada por Pressman na Figura 07 (vide referencial teórico, subseção engenharia de software). A *fase* corresponde ao escopo maior (atividade metodológica), que visa o cumprimento de um determinado objetivo. Cada fase contém uma ou mais *disciplinas* (ação da engenharia de software), que são o conjunto de diferentes *tarefas*, para a realização de um determinado aspecto em uma área no ciclo do desenvolvimento.

Figura 22: Fluxo geral do MAV3D.



Fonte: Elaborado pelo autor , 2016.

A figura 22 apresenta uma visão geral do fluxo do processo de software proposto, decomposto em suas fases e disciplinas. Segue abaixo a explicitação das fases, disciplinas e tarefas da metodologia, em seus níveis e subníveis de abstração.

### 4.2.1.1 Fases

**Concepção** – Esta é a fase inicial do projeto que tem por finalidade definir os requisitos do sistema, delimitar seu escopo e elaborar sua modelagem inicial, fornecendo as visões necessárias para a próxima fase. Esta fase é marcada pela comunicação e participação do cliente e tem como marco o Documento de Especificação de Requisitos (DER).

**Estruturação** – Esta é uma fase fundamental da metodologia, ela busca complementar o levantamento e documentação da fase de concepção. A partir dela será definido o Documento de Projeto de Software (DPS), que será o marco que servirá de base para todo o sistema.

**Desenvolvimento** – Fase responsável pela codificação dos componentes e funcionalidades do ambiente, além dos processos da produção artística do ambiente virtual. É nesta fase que é gerada as versões funcionais (marco da fase) do ambiente virtual e a partir daí podem ser realizados vários testes para a validação do projeto, bem como testes alfa para o sistema em si.

**Entrega** – O objetivo desta fase é tornar o ambiente disponível e compreendido para o cliente. Nela é realizado o plano de implantação e entrega, acompanhamento, bem como o teste beta. Esta fase também prevê a manutenção do sistema, em seus aspectos de correção, implantação de novas funcionalidades e suporte. Esta fase possui como marco a implantação do sistema dentro do ambiente de aplicação.

#### 4.2.1.2 *Disciplinas*

**Modelagem** – Refere-se a modelagem inicial do sistema através do diagrama de caso de uso da UML, tendo como base os requisitos apurados anteriormente. Seu artefato é o diagrama de casos de uso.

**Requisitos** – Trata dos aspectos da elicitação de requisitos. Nela são definidos os usuários, tarefas, interações e o grau de imersividade e realismo, estabelecendo assim o escopo do ambiente virtual. Esta disciplina possui como artefato o DER.

**Análise** – Avaliação da documentação gerada anteriormente, com o intuito de particionar os aspectos (visões) do sistema, otimizando assim as atividades do projeto e definindo o funcionamento geral do sistema. Podem ser definidos aqui também quais os profissionais necessários para a formação da equipe. Possui como artefato relatórios e/ou modelos conceituais.

**Projeto** – É nesta disciplina que são fechadas as bases estruturais, comportamentais e conceituais para a implementação em si. Tem por objetivo definir a viabilidade do sistema como todos em todos seus aspectos (recursos de hardware, software). Esta disciplina possui como artefato o DPS, que é um conjunto dos aspectos: estrutural, comportamental e artístico. O projeto arquitetural é contemplado pelo diagrama de classe e de pacotes, o projeto

procedimental pelos diagramas de sequência, de atividade e de estado; o projeto de dados pelo diagrama de entidade relacionamento e dicionário de dados; o projeto de interface do usuário pelos esboços das telas – dentro (HUD) e fora do ambiente virtual; e o projeto audiovisual pelos conceitos artísticos dos elementos tridimensionais e ficha descritiva dos elementos sonoros presentes no ambiente virtual.

**Implementação** – É através desta disciplina, que de fato, a codificação é feita, bem como a modelagem dos modelos tridimensionais (avatars, objetos e cenários) e composições das trilhas e efeitos sonoros. Como fruto desta produção vários artefatos são concebidos, como: códigos objeto, arquivos (em diversos formatos) e na documentação, relatórios podem ser elaborados. Além da própria versão de software a ser apresentada ao cliente.

**Teste e Validação** – Esta disciplina tem por finalidades: Verificar a interação entre objetos; Verificar a integração adequada de todos os componentes do software; Verificar se todos os requisitos foram corretamente implementados; Identificar os defeitos antes da implantação do software; e Garantir que todos os defeitos são corrigidos, reanalisados e fechados. A medida que os testes e validações são realizados relatórios podem ser expedidos relatórios pelos membros responsáveis.

**Implantação** – O objetivo da implantação é o de produzir com sucesso lançamentos de produtos e entregar o software para seus usuários finais. Ele cobre uma vasta gama de atividades, incluindo a produção de releases externos do software, distribuição do software, instalação do software e calibração dos equipamentos. Caso seja necessário, relatórios podem ser expedidos neste processo.

**Manutenção** – Esta disciplina preza pela prestação da assistência aos usuários (suporte), implantação de novas funcionalidades e correção de falhas (*bugs*) e mudanças dentro sistema. Seus artefatos de saída podem ir desde relatórios até mesmo uma nova versão do sistema.

#### 4.2.1.3 Tarefas

Em um nível mais detalhado de cada fase, está as tarefas. Que de fato representam a ação realizada pelo desenvolvedor ou algum *stakeholder* participante no processo. Abaixo encontra-se uma lista das atividades, separadas por cada fase.

*Concepção:* Modelagem de sistema; elitação de requisitos; definição dos requisitos; definição dos níveis de imersão e realidade; estimativa de custos e prazos; definição de prioridades; e definição das tecnologias que serão usadas.

*Estruturação:* Análise da modelagem; análise de riscos; definição das visões; projeto arquitetural; projeto de dados; projeto de interfaces; projeto procedimental; e projeto audiovisual.

*Desenvolvimento:* Codificação dos componentes do sistema; modelagem tridimensional do cenário, objetos e avatares; composição da sonorização; desenho das GUIs; plano de testes; teste de unidade; teste de integração; teste de usabilidade; teste de desempenho e teste de sistema.

*Entrega:* Integração ao ambiente; treinamento; entrega de material de suporte; calibração dos dispositivos e *releases*.

Quadro 1: Resumo da hierarquia dos componentes.

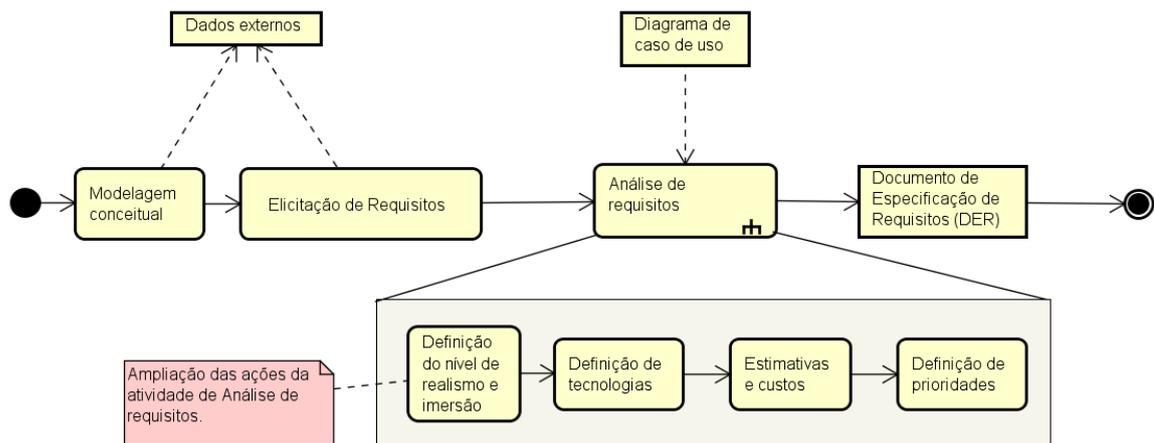
<b>Fase:</b> Concepção.
<u>Disciplina (s):</u> Modelagem e Requisitos.
<i>Tarefas:</i> Modelagem conceitual do sistema; elitação de requisitos; definição dos requisitos; definição dos níveis de imersão e realidade; estimativa de custos e prazos; definição de prioridades; e definição das tecnologias que serão usadas.
<b>Fase:</b> Estruturação.
<u>Disciplina (s):</u> Análise e Projeto.
<i>Tarefas:</i> Análise da modelagem; análise de riscos; definição das visões; projeto arquitetural; projeto de dados; projeto de interfaces; projeto procedimental; e projeto audiovisual.
<b>Fase:</b> Desenvolvimento.
<u>Disciplina (s):</u> Implementação; Teste e Validação.
<i>Tarefas:</i> Codificação dos componentes do sistema; modelagem tridimensional do cenário, objetos e avatares; composição da sonorização; desenho das GUIs plano de testes; teste de unidade; teste de integração; teste de usabilidade; teste de desempenho e teste de sistema.
<b>Fase:</b> Entrega.
<u>Disciplina (s):</u> Implantação e Manutenção.
<i>Tarefas:</i> Integração ao ambiente; treinamento; entrega de material de suporte; calibração dos dispositivos e <i>releases</i> .

Fonte: Elaborado pelo autor , 2016.

### 4.3 FLUXO DAS ATIVIDADES

Esta subseção tem por objetivo apresentar o diagrama de atividade de cada fase com seus respectivos processos, subprocessos, entradas e saídas. O primeiro diagrama de atividade a ser apresentado é de concepção (Figura 23). Como mencionado anteriormente essa fase é marcada pela presença do cliente, através do princípio da comunicação, pois a participação dele nos processos facilita a modelagem da visão do problema, o que é retratado através da dependência entre as ações de modelagem conceitual e elicitação de requisitos com os dados externos (informações do cliente). Através do diagrama de atividade é possível perceber também que o processo de análise requisitos é decomposto em outras tarefas e que somente após o término delas é viabilizado o DER.

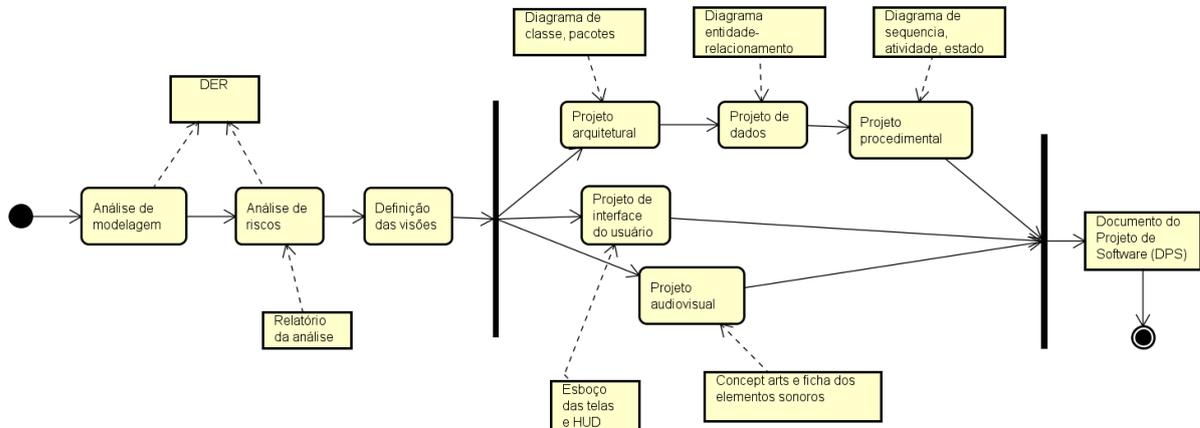
Figura 23: Diagrama de atividade da fase de concepção.



Fonte: Elaborado pelo autor, 2016.

Com o diagrama de atividade de estruturação (Figura 24) fica claro perceber que os documentos produzidos na fase anterior servem de insumo nesta fase. Outro ponto importante a ser destacado é quanto as definições das visões do sistema. Nota-se que os projetos arquitetural, audiovisual e de interface do usuário não necessitam serem realizados sequencialmente, equipes diferentes podem trabalhar em suas produções, otimizando assim o tempo durante esta fase. Em contrapartida, os projetos de dado e procedimental necessitam das definições estabelecidas no projeto arquitetural. Outro ponto seriam as produções dos artefatos relativos a cada projeto que a posteriori são integrados, formando o DPS.

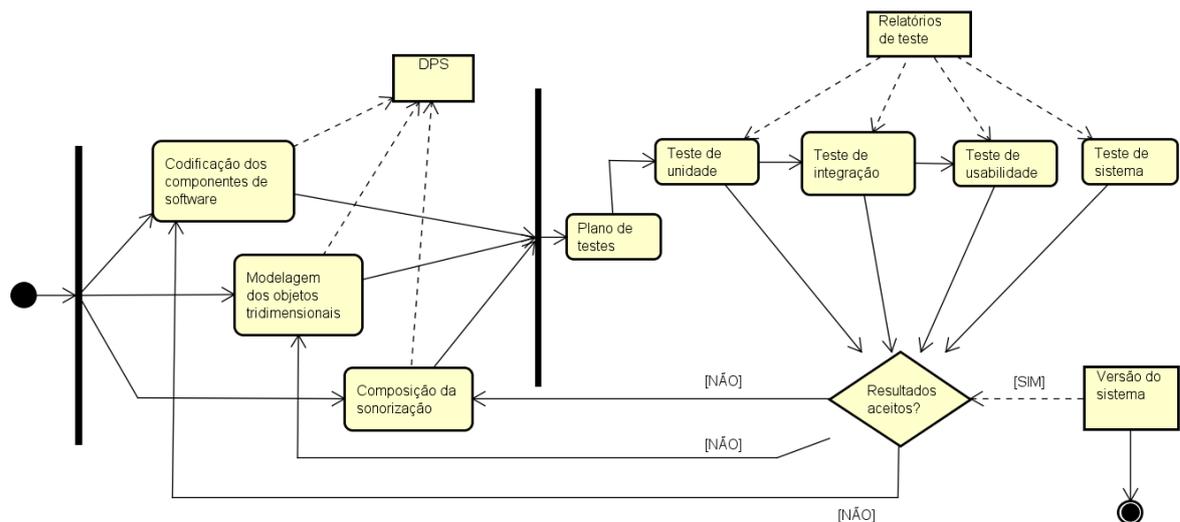
Figura 24: Diagrama de atividade da fase de estruturação.



Fonte: Elaborado pelo autor, 2016.

Assim como na fase de estruturação, o artefato produzido na fase anterior, neste caso o DPS, serve de base para as etapas iniciais na fase do desenvolvimento. De acordo com o apresentado na figura 25, percebe-se que os processos iniciais de codificação, modelagem tridimensional e composição sonora também mantêm independências em suas implementações e que após suas conclusões é possível realizar o plano de testes. Cada teste pode gerar um relatório e que apenas com resultados satisfatórios nos testes uma versão do sistema é liberada.

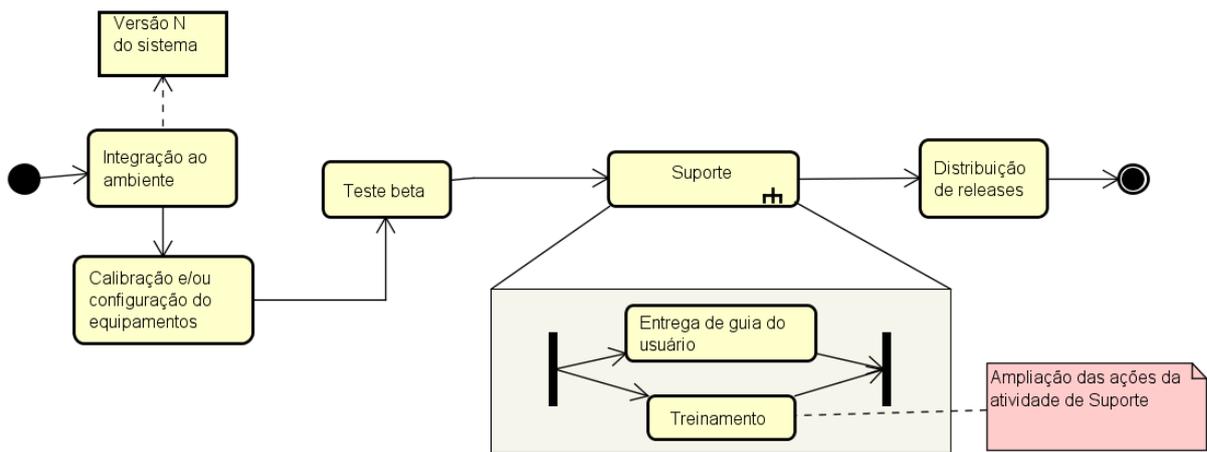
Figura 25: Diagrama de atividade da fase de desenvolvimento.



Fonte: Elaborado pelo autor, 2016.

O último diagrama de atividade, apresentado na figura 26, é referente à fase de entrega. Uma vez que uma versão foi liberada ela pode ser integrada ao ambiente de aplicação, onde estará sujeita a calibração e/ou configuração dos equipamentos e ao teste beta. Depois é possível realizar as ações de suporte e à medida que o sistema evolui *releases* podem ser implantados.

Figura 26: Diagrama de atividade da fase de entrega.



Fonte: Elaborado pelo autor, 2016.

#### 4.4 ESPECIALIZAÇÃO DO MAV3D

Segundo a visão apresentada por Falbo na figura 08, é possível realizar a melhoria de software tendo como base uma metodologia genérica e a partir dela gerar uma metodologia especializada, a fim de atender os propósitos específicos de um determinado domínio. Tendo essa ideia como base é viável afirmar que o MAV3D partiu do conjunto de diferentes processos de softwares e especializou-se quanto aos ambientes virtuais tridimensionais.

Esta subseção vem enaltecer e explicitar as atividades e tarefas especializadas, que fazem o diferencial dentro desta metodologia.

O primeiro item a ser abordado é **definição dos níveis de realismo imersão**. Esta tarefa permite logo na etapa de requisitos estabelecer os critérios de imersão e realidade, isto por consequência afeta a definição das tecnologias que serão usadas dentro do sistema, pois elas devem dar conta do processamento, que pode ser em tempo-real, e não devem permitir atrasos demasiados no tempo de resposta das interações. Além de permitir aos responsáveis pela modelagem 3D estabelecerem quais técnicas deverá ser usadas para o nível de detalhe das malhas tridimensionais.

Uma vez definidos os requisitos é possível partir para o **projeto audiovisual**. Esta atividade de projeto se refere a conceitualização dos aspectos visuais e de sonorização presentes dentro do sistema.

O objetivo é contemplar os requisitos não-funcionais em uma atividade de projeto, no que tange ao cenário, personagens e elementos interativos do ambiente, somado a descrição dos estilos de músicas e efeitos sonoros presentes dentro do ambiente, no que Kirner e Salvador elegeram em seu trabalho como os aspectos tridimensional, multisensorial e realístico.

As atividades realizadas neste projeto, no caso os *concept arts*, formam os esboços dos elementos tridimensionais definidos como interativos e o esboço do mundo virtual (cenário) e do avatar - representante do usuário dentro do ambiente virtual. Também deverá ser feita uma descrição textual dos aspectos sonoros, informando: nome do arquivo; sua categoria - trilha ou efeito; onde ele será aplicado; quando será aplicado; e uma descrição de qual intenção o som deve contemplar.

O próximo ponto trata da **modelagem tridimensional do cenário, objetos e avatares** propriamente ditos, realizada durante a fase de desenvolvimento. Como mencionado anteriormente, quanto mais se aproximar o nível de detalhe da realidade maior será a complexidade da modelagem tridimensional e do comportamento da física no mundo. Esta tarefa tem como base os *concept arts* do projeto audiovisual e os requisitos definidos no DER.

O último item trata da **composição da sonorização**, também realizada na fase de desenvolvimento. A partir das fichas estabelecidas no projeto audiovisual é possível a criação e edição das trilhas e efeitos sonoros que farão parte do ambiente. Uma vez que, como destacado anteriormente, a sonorização mostra-se fundamental para o desenvolvimento de uma ambientação imersiva, realística e multisensorial, que são aspectos natos de um sistema de realidade virtual.

## 5 RESULTADOS E DISCUSSÕES

### 5.1 CONSIDERAÇÕES SOBRE O MAV3D

Esta seção apresenta algumas considerações sobre o MAV3D com base no que foi apresentado no capítulo anterior. Para tanto, foi listado abaixo alguns pontos que descrevem suas vantagens e aplicações de uso.

- É uma metodologia especialista. O MAV3D foi elaborado para atender uma demanda específica. Sua metodologia não se enquadra no desenvolvimento de qualquer tipo software ou sistema, mas busca resolver de forma eficaz a que propõem. Traz o problema do “tridimensional” para camadas iniciais da solução, com especificação de seus requisitos e projeto próprio.
- Baseada em métodos e metodologias consagradas. O MAV3D tem como sua maior base o RUP, que é uma metodologia já avaliada, conhecida e aceita no mercado mundial. Além disso, agrega práticas do *game design*, outro modelo de práticas já bem definidas, em relação ao processo e concepção dos elementos visuais e sonoros.
- Permite vários níveis de visões. Visões, neste contexto referem-se às abordagens de abstrações existentes dentro da metodologia, tanto no quesito dos domínios, já mencionados anteriormente, quanto ao fato das hierarquias existente dentro da proposta.
- É flexível. De maneira geral o MAV3D garante a flexibilidade ao adotar um fluxo iterativo e incremental em seu processo. De maneira específica ele prevê o “salto” de atividades, pois compreende que nem todos os SRVs possuem a mesma necessidade, por exemplo: sistemas não-imersivos não necessitam de calibração de equipamentos na etapa de implantação.
- Baseada no projeto. Neste ponto ele diferencia-se no RUP, que tem seu princípio apoiado na arquitetura do sistema. Como um sistema tridimensional necessita ser apoiado em outros aspectos, além do estrutural, ser baseado no projeto garante essa amplitude.

Ratificando o que foi escrito nesta seção é válido afirmar que, uma abordagem que trate de forma concisa e coerente o desenvolvimento de AVs tridimensionais é mais válida

que uma abordagem genérica que necessita ser readaptada em vários momentos do desenvolvimento.

## 5.2 PROBLEMAS ENCONTRADOS E SOLUÇÕES APLICADAS

Antes de passar para as considerações finais deste trabalho é importante destacar alguns questões referentes as adversidades e complicações as quais o autor passou ao elaborar a proposta do MAV3D.

No início do trabalho o autor estava familiarizado com alguns aspectos dos processos de software e acreditava que apenas um estudo dos modelos de software já existentes, juntamente com os fundamentos e conceitos da realidade virtual seriam suficientes para elaborar uma proposta de metodologia. De forma que, além da proposta de metodologia outra ação seria realizada, a construção de um projeto de software, que seria elaborada segundo as recomendações definidas pelo MAV3D.

Porém ao decorrer da execução do trabalho, a consolidação da proposta metodológica necessitou de maiores fundamentos, o que levou o autor a incluir em seu fundamento um estudo específico sobre as metodologias e padrões para processos de software, e a melhoria e avaliação de processos de software. Consequentemente este aprofundamento consumiu mais recurso de tempo do autor, de modo que, a realização do projeto ficou comprometida levando a um impasse dentro da realização deste trabalho. Finalizar a metodologia e dar seguimento ao projeto ou aprofundar na metodologia e abrir mão da realização do projeto?

Optou-se então pelo aprofundamento da metodologia, por duas razões: a primeira é que este aprofundamento é capaz de prover um melhor embasamento e fundamentação ao MAV3D, o garante um melhor respaldo e facilita uma posterior avaliação do processo; a segunda razão é que a realização de um projeto de **qualidade** e bem elaborado demandaria mais tempo.

O autor acredita que uma avaliação da metodologia é um fator fundamental para sua real finalização e que esta ainda deve ocorrer. Mas compreende que esta demanda exige um trabalho de estudo específico para este propósito, repassando então esta tarefa para planos futuros.

## 6 CONSIDERAÇÕES FINAIS

### 6.1 CONCLUSÃO

Elaborar um processo de software, independente de seu propósito, que atenda as necessidades do desenvolvedor e que seja capaz de proporcionar um produto de software que corresponda os anseios do cliente, não é uma tarefa simplória. Mesmo que esta metodologia, como no caso do MAV3D, seja apoiada em outras existentes.

A realização de tal tarefa não seria possível sem o correto embasamento teórico, no caso deste estudo, trata-se da engenharia de software – que proveu os conhecimentos necessários sobre os ditos processos de softwares clássicos e ágeis, bem como os sobre as melhorias e padrões de processos; da realidade virtual – que consistiu do domínio do problema, uma vez que, a metodologia proposta foi direcionada a ambientes virtuais tridimensionais; e dos trabalhos relacionados, pelos quais foi possível direcionar a proposta do MAV3D e posteriormente realizar um comparativo entre os modelos metodológicos.

Ao decorrer deste estudo dificuldades foram encontradas, principalmente do que trata da real avaliação do processo de software proposto neste trabalho, já que a realização desta tarefa demanda da implementação de vários projetos pilotos, o que necessitaria de mais recursos e esforços de maneira geral. Mas, apesar destas adversidades, é possível afirmar que este trabalho logrou êxito em seu propósito, pois foi capaz de atender todos os objetivos propostos além de contemplar de forma eficaz do que tange um trabalho monográfico, como destacado no capítulo da metodologia.

Por fim, conclui-se que este trabalho por si só contribui para o cenário acadêmico, principalmente por conta de seu caráter multi-disciplinar. Outro ponto relevante é quanto a demanda no mercado mundial, que oferece muitas oportunidades, uma vez que as produções de software convergem cada vez mais ao uso de ambientes virtuais 3D, necessitando assim de metodologias específicas. Contudo, para que possa haver uma aceitação efetiva do MAV3D dentro do mercado outros estudos precisam ser realizados, gerando assim um aprofundamento na melhoria e qualidade do processo de software.

## 6.2 TRABALHOS FUTUROS

Como proposta para a continuidade deste trabalho algumas ações poderão ser realizadas em maior ou menor escala de complexidade. Como o próprio aprimoramento da metodologia proposta, a fim de atingir uma maior eficiência e maturidade. A implementação de diferentes ambientes virtuais tridimensionais, imersivos ou não, ou até mesmo com presença de realidade aumentada, que serviriam de objeto para um real teste e avaliação da aplicação da metodologia. Durante estas ações algumas publicações e apresentações acadêmicas também seriam possíveis, agregando assim maior legibilidade e credibilidade ao MAV3D.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520:2002** – Informação e Documentação – Citações em Documentos – Apresentação. 2002.

\_\_\_\_\_. **NBR 6023:2002** – Informação e Documentação – Referências – Elaboração. 2002.

\_\_\_\_\_. **NBR ISO/IEC 12207** – Tecnologia de Informação – Processos de ciclo de vida de software. 1998.

ASSUNÇÃO, Rodrigo P.; VASCONCELLOS, Fabrício P.. **MPS.Br – A experiência e os benefícios de implantar o modelo nos níveis G E F: um estudo de caso.** Disponível em: <[http://revistapensar.com.br/tecnologia/pasta\\_upload/artigos/a50.pdf](http://revistapensar.com.br/tecnologia/pasta_upload/artigos/a50.pdf)>. Acesso em: 05 set 2016.

BERNADES JÚNIOR, João Luiz. **Desenvolvimento de um ambiente para visualização tridimensional da dinâmica de risers.** São Paulo, SP. 2004. Originalmente apresentado como dissertação de mestrado, Escola Politécnica da Universidade de São Paulo.

FALBO, Ricardo de Almeida. **Engenharia de Software: notas de aula.** 2005. Disponível em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>>. Acesso em: 05 set. 2016.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa.** 4. Ed. São Paulo: Atlas, 2002.

KIRNER, Tereza G.; SALVADOR, Valéria F. M.. **Contribuição à engenharia de requisitos de ambientes virtuais.** Disponível em: <[http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_wer04/Tereza\\_Kirner.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_wer04/Tereza_Kirner.pdf)>. Acesso em: 25 nov. 2015.

LAHOZ, Carlos; SANT'ANNA, Nilson. **Os padrões ISO/IEC 12207 e 15504 e a modelagem de processos da qualidade de software.** Disponível em: <[http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/worcap/2003/10.20.14.01/doc/LahozWorkcap\\_versaofinal.pdf](http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/worcap/2003/10.20.14.01/doc/LahozWorkcap_versaofinal.pdf)>. Acesso em: 05 set. 2016.

MATTIOLI, Fernando E. L. *et al.* **Uma proposta para o desenvolvimento ágil de ambientes virtuais**. Disponível em: <[www.lbd.dcc.ufmg.br/colecoes/wrva/2009/0021.pdf](http://www.lbd.dcc.ufmg.br/colecoes/wrva/2009/0021.pdf)>. Acesso em: 19 abr. 2016.

MORESI, Eduardo. **Metodologia da pesquisa**. Brasília: UCB, 2003.

NETTO, Antônio V.; MACHADO, Liliane dos A.; OLIVEIRA, Maria C.. **Realidade virtual – definições, dispositivos e aplicações**. Disponível em: <[www.di.ufpb.br/liliane/publicacoes/2002\\_reic.pdf](http://www.di.ufpb.br/liliane/publicacoes/2002_reic.pdf)>. Acesso em: 19 abr. 2016.

OSÓRIO, Fernando S. *et al.* **Ambientes virtuais interativos e inteligentes: Fundamentos, implementação e aplicações práticas**. Disponível em: <<http://osorio.wait4.org/publications/Osorio-et-al-JAI2004.pdf>>. Acesso em: 30 dez. 2015.

PAGNEZ, Karina Soledad M. M.. **Apostila de metodologia do trabalho científico**. São Paulo. 2007.

PAULA FILHO, Wilson de Pádua. **Engenharia de software: fundamentos, métodos e padrões**. 3. ed. Rio de Janeiro: LTC, 2009.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Pearson Makron Books, 1995.

\_\_\_\_\_. **Engenharia de software**. 6. ed. Porto Alegre: AMGH Editora Ltda, 2010.

\_\_\_\_\_. **Engenharia de software: uma abordagem profissional**. 7. ed. AMGH Editora Ltda, 2011.

RIBEIRO, Marcos Wagner S.; Zorzal, Ezequiel Roberto (Orgs). Simpósio de Realidade Virtual e Aumentada, 13. 2011. Uberlândia. **Realidade virtual e aumentada: aplicações e tendências – SVR 2011 - Livro do Pré-Simpósio**.

RODRIGUES, Gessica P.; PORTO, Cristiane de M.. Interfaces Científicas – Educação.V.01 N.03. **Realidade virtual: conceitos, evolução, dispositivos e aplicações**. Aracajú ,jun, 2013.

ROSA JUNIOR, Onivaldo. **LRVCHAT3D, Desenvolvimento de um ambiente virtual tridimensional multiusuário para a internet.** Florianópolis, SC. 2003. Originalmente apresentado como dissertação de mestrado, Universidade Federal de Santa Catarina.

SILVA, Thiago Jamir. **O MPS.Br como alternativa a micro e pequenas empresas:** um estudo de caso. Recife, PE. 2008. Originalmente apresentado como trabalho de conclusão de curso, Universidade Federal de Pernambuco.

SOMMERVILLE, Ian. **Engenharia de software.** 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

TAKAHASHI, Patrícia K.; ANDREO, Marcel C.. **Desenvolvimento de concept art para personagens.** Disponível em: <  
[www.sbgames.org/sbgames2011/proceedings/sbgames/papers/art/full/92122.pdf](http://www.sbgames.org/sbgames2011/proceedings/sbgames/papers/art/full/92122.pdf) >. Acesso em: 22 set. 2015.

TOCCHIO, Luis Cláudio. O. **Video game design:** Uma análise da estética conceitual do entretenimento digital. São Paulo, SP. 2007. Originalmente apresentado como trabalho de conclusão de curso de especialização, Centro Universitário SENAC.

TORI, R.; KIRNER, C.; SISCOOTTO, R.. **Fundamentos e tecnologias da realidade virtual e aumentada.** Belém. 2006. Livro do Pré-Simpósio - VII Symposiun on Vitual Reality.