

# Uma Proposta para o Desenvolvimento Ágil de Ambientes Virtuais

Fernando E. R. Mattioli, Edgard A. Lamounier Jr., Alexandre Cardoso  
Universidade Federal de Uberlândia  
Uberlândia-MG, Brasil  
mattioli.fernando@gmail.com,  
{lamounier, alexandre}@ufu.br

Nélio M. M. Alves  
Instituto Federal do Triângulo Mineiro  
Uberlândia-MG, Brasil  
nelio@iftriangulo.edu.br

## Abstract

*Agile software process presents an iterative and incremental approach to computer systems development, which focuses on user's needs and embraces changes. Virtual Reality projects are characterized by the rapid evolution of technology and by the need for clients' feedback, during the whole project's lifecycle. In this work, a comparative evaluation of existent methodologies is presented and the application of agile software development process in Virtual Reality projects is argued. Then, a proposal for an agile software development process for Virtual Reality systems is presented and its benefits are discussed.*

## 1. Introdução

Processos ágeis de desenvolvimento têm como característica uma abordagem iterativa e incremental dos princípios de Engenharia de Software. Tal abordagem revela-se extremamente adequada aos projetos de Realidade Virtual e proporciona, por sua natureza evolutiva, grandes benefícios associados à gestão de riscos em projetos de software [3].

A palavra ágil foi relacionada pela primeira vez à Engenharia de Software em 2001, por um consórcio de especialistas em métodos de desenvolvimento, que elaboraram na ocasião o “Manifesto Ágil” [2]. Tal manifesto destaca alguns dos princípios compartilhados por diferentes metodologias de desenvolvimento, a partir de então denominados processos (ou métodos) ágeis [8]:

- Indivíduos e interações são mais importantes que processos e ferramentas;

- Software funcionando é mais importante do que documentação detalhada;
- Colaboração dos clientes é mais importante de que negociação de contratos;
- Adaptação às mudanças é mais importante do que planejamento extensivo.

Por outro lado, o desenvolvimento de aplicações em Realidade Virtual requer pleno conhecimento e entendimento de diversas disciplinas, tais como computação gráfica, modelagem geométrica, interação multimodal, entre outras [13]. Algumas características destes sistemas revelam a necessidade de um aprimoramento contínuo em seu processo de desenvolvimento. Dentre essas características, destacam-se:

- Rápida evolução da tecnologia relacionada à visualização e à capacidade gráfica dos computadores [19];
- Indecisão e mudanças de opinião por parte dos clientes, problema agravado em sistemas compostos por equipamentos de alto custo [19];
- Necessidade da implementação contínua de protótipos, visando a avaliação da viabilidade do sistema pelo cliente.

O desenvolvimento evolucionário, o planejamento adaptativo e o acolhimento de alterações nos requisitos apresentam-se como possíveis melhorias associadas ao desenvolvimento ágil de ambientes virtuais.

## 2. Desenvolvimento ágil de software

O desenvolvimento ágil de software é uma abordagem de desenvolvimento caracterizada pela adaptabilidade, ou seja, pela capacidade de absorver mudanças, sejam elas nas forças de mercado, nos requisitos do sistema, na tecnologia de implementação ou nas equipes de projeto [5].

As seções 2.1 e 2.2 apresentam uma breve descrição de dois dos principais métodos ágeis: XP e SCRUM.

### 2.1. Programação Extrema (XP)

O conceito de Programação Extrema teve sua concepção e desenvolvimento em resposta às necessidades de pequenas equipes de desenvolvimento de software envolvidas em projetos com alto grau de alteração (ou mesmo ausência) de requisitos. Trata-se de uma metodologia leve de desenvolvimento, cujos fundamentos incluem [1]:

- Testes unitários são escritos antes da parte correspondente do programa. Esses testes são mantidos e executados durante todo o ciclo de vida do projeto;
- Integração e testes do sistema como um todo, várias vezes por dia;
- O projeto começa com uma arquitetura simples, que evolui constantemente no sentido de agregar flexibilidade e reduzir a complexidade desnecessária do próprio sistema;
- Um sistema mínimo é rapidamente implementado e colocado em produção. Tal sistema evoluirá de acordo com as direções tomadas pelo projeto.

Dentre os benefícios da programação extrema, pode-se destacar [1]:

- Não força a especialização prematura dos membros da equipe. Todos os membros desempenham diferentes papéis dentro da equipe de desenvolvimento, todos os dias;
- Análise e projeto são realizados durante todo o ciclo de vida do projeto, favorecendo a adaptabilidade e a rápida resposta deste às mudanças em seu ambiente;
- A infraestrutura do projeto é construída iterativamente, acompanhando sua evolução natural e atendendo às suas reais necessidades.

A Figura 1 apresenta o as principais etapas do processo XP.

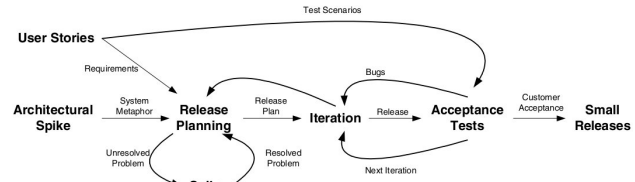


Figura 1. XP - Ciclo de vida [10].

### 2.2. SCRUM

A metodologia SCRUM tem como objetivo gerenciar e controlar a produção de software utilizando processos leves, iterativos e incrementais. Em SCRUM, cada iteração (com duração típica de 30 dias) é conhecida como *sprint*. A cada *sprint*, é associado um conjunto de tarefas, e a cada tarefa é atribuído o número de horas restantes estimado à sua finalização [18]. As tarefas de uma *sprint* podem ser realocadas caso haja disparidade entre o cronograma do projeto e o real desenvolvimento das atividades.

Em sua essência, o processo SCRUM consiste em um conjunto de regras, procedimentos e práticas, interrelacionadas entre si e visando a melhoria do ambiente de desenvolvimento [10]. Estes fatores são ilustrados na Figura 2.

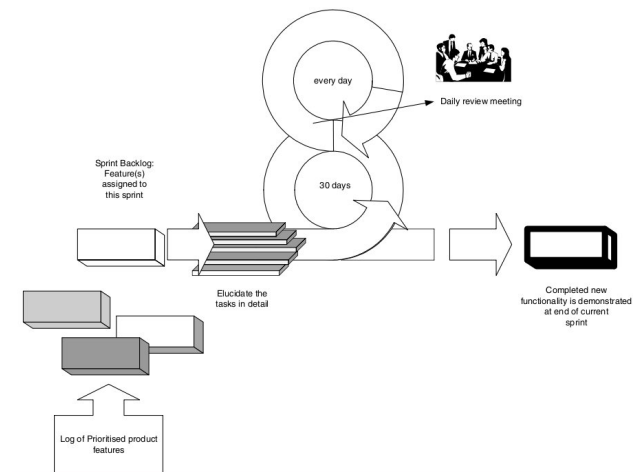


Figura 2. SCRUM [10].

Apesar de complementares, as metodologias XP e SCRUM possuem diferentes aplicações em diversos aspectos do desenvolvimento de software. Enquanto SCRUM pode ser considerada uma metodologia de gestão ágil de projetos, XP deve ser tratada como uma metodologia de engenharia, ou seja, de desenvolvimento propriamente dito. SCRUM prioriza aspectos como maior visibilidade do projeto, melhor inserção do projeto no mercado e melhor co-

laboração entre os membros da equipe. Por sua vez, a metodologia XP possui como fatores motivadores a simplificação da gestão de requisitos e o aumento da qualidade do produto. Ambas metodologias possuem uma abordagem incremental e iterativa de desenvolvimento [20].

Para o presente trabalho, decidiu-se adotar como metodologia base de desenvolvimento a metodologia XP. Alguns dos aspectos mais relevantes desta escolha são o foco no desenvolvimento e na qualidade do produto, as curtas iterações (projetos com iterações de uma a duas semanas), e o desenvolvimento orientado a testes.

### 3. Desenvolvimento de Sistemas de Realidade Virtual

O desenvolvimento de sistemas de Realidade Virtual, assim como o desenvolvimento de qualquer sistema de software, demanda um processo, uma metodologia de desenvolvimento. No entanto, tal processo deve se adequar à rápida evolução da tecnologia associada a estes sistemas.

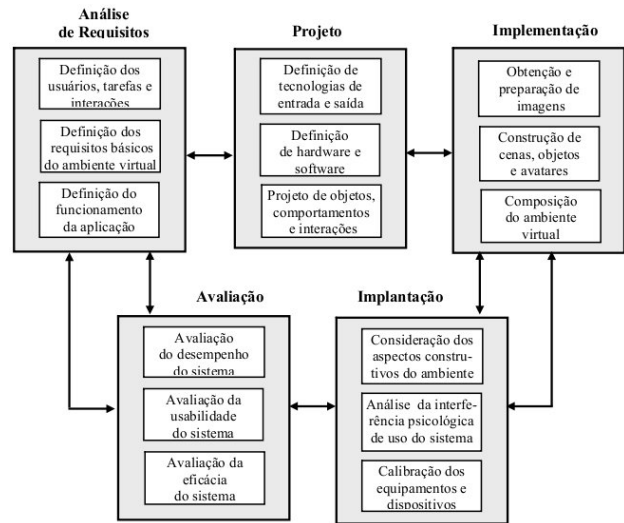
Em [19] é apresentado um processo de desenvolvimento que agrega características de prototipagem, desenvolvimento iterativo e desenvolvimento evolucionário de sistemas de software. Este processo baseia-se em conceitos e modelos da Engenharia de Software, adaptados às peculiaridades dos Sistemas de Realidade Virtual (SRV). Este processo compõe-se de cinco etapas, realizadas iterativamente: *Análise de requisitos*, *Projeto*, *Implementação*, *Avaliação* e *Implantação*.

A Figura 3 traz uma representação gráfica deste processo.

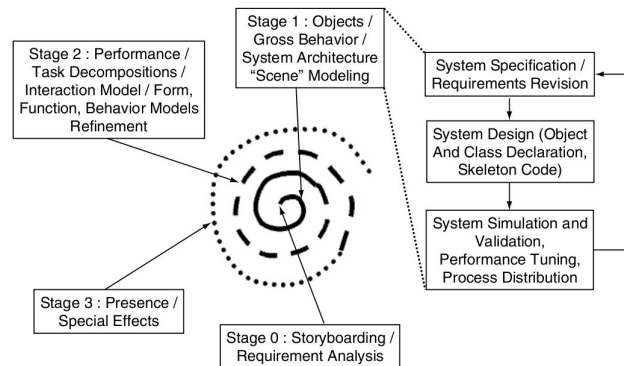
Uma outra abordagem de desenvolvimento, aplicável aos sistemas de Realidade Virtual, é apresentada em [12]. De um ponto de vista global, esta abordagem pode ser comparada ao modelo espiral clássico, adaptado às necessidades específicas dos sistemas de Realidade Virtual, por meio de elementos tais como modelos de interação, modelagem de cenários etc. Este processo de desenvolvimento é apresentado na Figura 4.

A partir dos processos apresentados anteriormente, percebe-se uma forte influência da abordagem estruturada no desenvolvimento de sistemas de Realidade Virtual. Apesar disso, algumas das características comuns a estes sistemas relacionam-se direta ou indiretamente com as práticas adotadas no desenvolvimento ágil de software. Dentre estas características, pode-se enumerar:

- A natureza evolucionária dos SRV;
- A necessidade de modelos que representem iterativamente, forma, função e comportamento dos objetos componentes dos SRV;



**Figura 3. Etapas do Processo de Desenvolvimento de SRV [14].**



**Figura 4. Modelo espiral do desenvolvimento de sistemas de Realidade Virtual [12].**

- A melhor aceitação de sistemas com participação ativa dos usuários finais no processo de desenvolvimento, devido à necessidade de avaliação e de um constante *feedback* destes usuários;
- A necessidade da realização exaustiva de testes, buscando identificar e reduzir os problemas de interação dos usuários com o SRV.

### 4. Desenvolvimento ágil de ambientes virtuais

Nenhum processo de desenvolvimento pode garantir, simplesmente por sua aplicação, o sucesso do produto de

software ao qual foi aplicado [4]. No entanto, é possível destacar algumas características comuns a alguns processos bem sucedidos [16]:

- *Desenvolvimento iterativo*: projetos maiores, com vários componentes, são mais propensos a problemas de integração. Um planejamento adequado de iterações reduz os problemas de integração e facilita o acompanhamento do processo de desenvolvimento pelos gerentes de projeto;
- *Avaliação contínua do processo*: nenhum processo de desenvolvimento de software pode garantir a total imunidade do projeto face a problemas como mudanças na equipe de desenvolvimento ou nos requisitos do usuário. A avaliação (e consequente adaptação) do processo de desenvolvimento é de fundamental importância ao longo do ciclo de vida do projeto;
- *Boas práticas ("Best practices")*: as melhorias associadas à utilização de boas práticas de desenvolvimento [11, 15] ou de padrões de projeto [9] não devem ser negligenciadas em projetos de software.

A abordagem sugerida para a adaptação de um processo existente a determinado contexto consiste em personalizar um processo existente, testando e refinando-o de maneira iterativa, de acordo com as características de cada projeto [17]. No entanto, alguns princípios não podem ser menosprezados ao se adaptar um processo existente. Dentre estes, pode-se destacar [6]:

1. *Equipes maiores requerem metodologias e processos mais robustos*;
2. *Excessos no grau de complexidade de uma metodologia representam um custo adicional*;
3. *Projetos mais críticos requerem metodologias com alto nível de detalhamento*;
4. *Maior feedback e comunicação reduzem a necessidade de documentos e modelos intermediários*;
5. *Considerações legais, reguladoras ou de segurança podem exigir metodologias mais detalhadas*.

Assim, o objetivo deste trabalho é fornecer uma avaliação teórica da aplicação de processos ágeis de desenvolvimento em projetos de SRV. Para tal, a metodologia *programação extrema (XP)* foi utilizada como metodologia base, buscando-se adaptá-la às especificidades destes projetos.

Analisando o estado da arte das metodologias de desenvolvimento de SRV, buscou-se definir alguns dos princípios chave destes sistemas. Uma abordagem semelhante foi utilizada por [7], porém em um contexto voltado ao desenvolvimento WEB. No caso dos SRV, os princípios chave são:

1) A natureza evolucionária dos SRV; 2) A iteratividade da construção e alta fidelidade dos modelos; 3) A necessidade de *feedback* dos clientes; 4) A necessidade de testes de interação e usabilidade; 5) A modularização dos SRV.

Desta forma, os 5 princípios básicos da metodologia XP (*feedback* rápido, simplicidade presumida, mudanças incrementais, aceitação das mudanças e alta qualidade) podem ser aplicados também ao desenvolvimento de SRV. Deve-se destacar a importância da aceitação das mudanças para projetos de software desenvolvidos em ambientes caracterizados por constante mudança nos requisitos por parte dos clientes. A fidelidade da representação e o comportamento dos objetos virtuais representados constituem elementos de extrema importância na medida da qualidade de SRV. Além disso, um sistema de Realidade Virtual que possua falhas de interação com o usuário possuirá um escopo de utilização reduzido, ainda que suas outras funcionalidades atendam aos requisitos do sistema.

Finalmente, um processo de desenvolvimento de SRV deve englobar 6 atividades básicas: *Planejamento, Análise, Projeto, Codificação, Testes e Integração*. Todas essas atividades são desenvolvidas a cada iteração. O planejamento de uma iteração é feito a partir do *feedback* recebido das iterações anteriores e a cada iteração os modelos do sistema são atualizados.

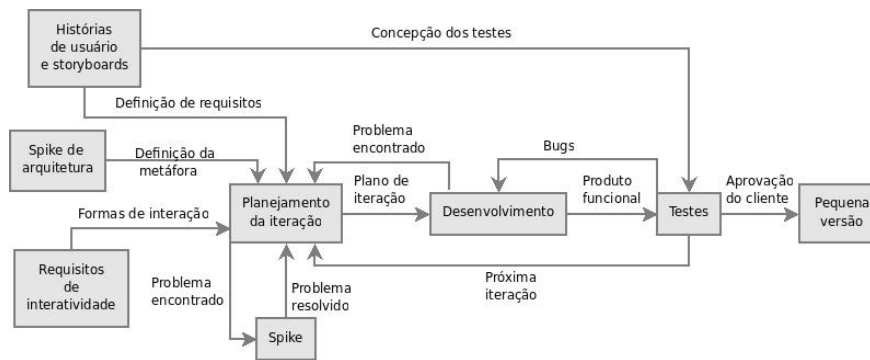
## 5. Processo ágil de desenvolvimento de SRV

A Figura 5 apresenta as principais atividades de um processo ágil para o desenvolvimento de SRV. Estas atividades são descritas a seguir.

### 5.1. Spike de arquitetura

O *spike de arquitetura* consiste em uma análise da viabilidade de implementação dos novos requisitos, obtidos a partir das histórias de usuário e dos *storyboards*. Esta etapa tem como objetivo reduzir os riscos presentes em áreas desconhecidas nos domínios da aplicação, tecnologia e do sistema propriamente dito. Do *spike de arquitetura* resulta a definição da metáfora, que será utilizada pela equipe de desenvolvimento para representar o sistema.

No *spike de arquitetura* ocorre essencialmente a exploração dos recursos disponíveis para a implementação dos requisitos da aplicação. A importância desta abordagem na construção de SRV está na existência de uma etapa dentro do processo de desenvolvimento que permite e incentiva a experimentação, fator de grande importância e constante utilização na modelagem de objetos e cenários virtuais. Este módulo, juntamente com os requisitos de interatividade, está fortemente relacionado aos avanços tecnológicos em RV.



**Figura 5. Desenvolvimento ágil de SRV: ciclo de vida.**

## 5.2. Requisitos de interatividade

A interatividade constitui o elemento central de diversos sistemas de Realidade Virtual, exercendo um papel de fundamental importância na determinação da usabilidade destes sistemas. Sendo assim, a análise detalhada e definição dos requisitos de interatividade, que estão intimamente relacionados à crescente tecnologia em ambientes de RV, ocupa uma posição de destaque dentro do processo de desenvolvimento de SRV.

Nesta etapa do processo de desenvolvimento, podem ser utilizadas aplicações de teste que implementem as formas de interação desejadas. Tais aplicações permitirão aos desenvolvedores e ao cliente a análise da viabilidade e a validação das formas de interação propostas.

## 5.3. Planejamento da iteração

O planejamento da iteração, realizado ao início de cada iteração, é feito a partir dos requisitos apresentados pelo usuário por meio das histórias de usuário e dos *storyboards*. Cada vez que um problema é encontrado durante o planejamento da iteração, é realizado um *spike* que tem como objetivo levantar soluções candidatas ao problema encontrado. Dentro do *spike*, podem ser implementados modelos ou mesmo aplicações de teste para constatar a viabilidade das soluções apresentadas.

Deve-se ressaltar a importância do caráter adaptativo do planejamento da iteração. A flexibilidade do planejamento da iteração deve ser buscada visando acolher eventuais mudanças nos requisitos relativos à iteração atual e ainda solucionar problemas encontrados durante o desenvolvimento.

## 5.4. Desenvolvimento

Uma vez definido o plano de iteração tem início o desenvolvimento do sistema, por sua vez constituído de 4 atividades principais: *análise, projeto, codificação e testes*.

A análise e o projeto têm como objetivo estruturar a maneira mais simples de se implementar os requisitos propostos no plano de iteração. Para tal, devem ser analisados os testes propostos pelos próprios clientes, a partir das histórias de usuário e dos *storyboards*. Estes testes servirão para orientar a equipe de desenvolvimento para a implementação das funcionalidades de maior importância do ponto de vista do cliente. Ao final das atividades de análise e projeto, os desenvolvedores elaboram seus próprios testes unitários, necessários à implementação das funcionalidades da iteração atual. De posse dos testes, inicia-se a codificação. O desenvolvimento termina quando o sistema executa com sucesso os testes propostos para a iteração atual.

É importante evidenciar o caráter evolutivo da atividade de desenvolvimento. O sistema constitui-se, inicialmente, de modelos simplificados, que representam os objetos constituintes do ambiente virtual. À medida que o sistema evolui, ocorre o refinamento destes modelos, o que resulta em objetos virtuais com forma e comportamento cada vez mais fiéis aos respectivos objetos reais.

## 5.5. Testes

Terminado o desenvolvimento, têm início os testes do sistema. As modificações efetuadas durante o desenvolvimento só serão integradas ao sistema principal após a comprovação de seu funcionamento pelos testes, definidos tanto pelos clientes como pelos desenvolvedores.

Se durante os testes são encontradas anomalias ou bugs, o projeto volta à atividade de desenvolvimento, para correção destes. Se o novo sistema é composto apenas por pequenas modificações e passa nos testes previstos, tem início o planejamento da próxima iteração. Novos requisitos serão então definidos de acordo com as prioridades dos clientes.

Um novo sistema, com um número significativo de modificações, que passe nos testes propostos é submetido a uma avaliação do cliente. Quando o cliente aprova as modificações efetuadas, pode ser entregue uma pequena ver-

são (versão parcial) do sistema, que implementa satisfatoriamente parte dos requisitos propostos.

Para o caso dos Sistemas de Realidade Virtual, deve-se ressaltar a importância da realização exaustiva de testes específicos de interação. A eficiência e a usabilidade de um SRV ficarão comprometidas, caso este sistema possua anomalias ou falhas de interação.

## 6. Conclusões

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.

Os *spikes* de arquitetura constituem um importante melhoramento do processo de desenvolvimento de projetos de Realidade Virtual, na medida em que facilitam o acolhimento das mudanças associadas à rápida evolução da tecnologia dos SRV. A correta utilização de *storyboards* na atividade de análise de requisitos revela-se também como uma atividade importante no desenvolvimento destes sistemas, por antecipar o *feedback* dos usuários e fornecer uma representação aproximada do sistema antes de sua modelagem ou implementação. É importante destacar que esta atividade, associada aos *spikes* de arquitetura, deve ser cuidadosamente trabalhada tendo em mente as adaptações que são feitas no sistema devido ao avanço tecnológico. Finalmente, em projetos de Realidade Virtual, a correta definição dos requisitos de interatividade do sistema é de total relevância na determinação da usabilidade e da eficiência do mesmo, constituindo um fator primordial para o sucesso do projeto.

Uma sugestão para trabalhos futuros é a definição de métricas de avaliação, que permitam uma avaliação mais objetiva da aplicação do processo proposto neste trabalho em projetos de Realidade Virtual. A extensão do processo proposto aos projetos de Realidade Aumentada também constitui um tema importante a ser desenvolvido em trabalhos futuros.

## Referências

- [1] K. Beck. *Programação extrema explicada: acolha as mudanças*. Bookman, Porto Alegre, 2004.
- [2] K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. J. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development. Disponível em: <<http://www.agilemanifesto.org>>, 2001. Acesso em: 25 mar. 2009.
- [3] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, C. Jim, and H. K. A. *Object-oriented analysis and design with applications*. John Wiley & Sons, Westford, 3 edition, 2007.
- [4] F. Brooks. No silver bullet: essence and accidents of software engineering. *IEEE computer*, 20(4):10–19, 1987.
- [5] A. Cockburn. *Agile software development*. Addison-Wesley, Boston, 2002.
- [6] A. Cockburn. *Agile software development: the cooperative game (agile software development series)*. Addison-Wesley Professional, Boston, 2006.
- [7] G. Concas, M. Di-Francesco, M. Marchesi, R. Quaresima, and S. Pinna. An agile development process and its assessment using quantitative object-oriented metrics. In *9th International Conference on Agile processes and eXtreme programming in Software Engineering*, pages 83–93. Springer, 2008.
- [8] E. G. d. Costa-Filho, R. Penteado, J. C. A. Silva, and R. T. V. Braga. Padrões e métodos ágeis: agilidade no processo de desenvolvimento de software. *SugarLoafPLOP' 2005 5th Latin American Conference on Pattern Language of Programming*, 5:156–169, 2005.
- [9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Padrões de projeto: soluções reutilizáveis de software orientado a objetos*. Bookman, Porto Alegre, 2000.
- [10] J. Hunt. *Agile software construction*. Springer, London, 2006.
- [11] B. W. Kernighan and R. Pike. *The practice of programming*. Addison-Wesley, Reading, 1999.
- [12] G. J. Kim. *Designing virtual reality systems: the structured approach*. Springer, London, 2005.
- [13] G. J. Kim, K. C. Kang, H. Kim, and L. Jiyoung. Software engineering of virtual worlds. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 131–138, 1998.
- [14] T. G. Kirner and V. F. Martins. A model of software development process for virtual environments: Definition and a case study. In *ASSET '99: Proceedings of the 1999 IEEE Symposium on Application - Specific Systems and Software Engineering and Technology*, page 155, Washington, DC, USA, 1999. IEEE Computer Society.
- [15] A. Oram and G. Wilson, editors. *Beautiful code*. O'Reilly Media, Sebastopol, 2007.
- [16] D. Pilone and R. Miles. *Head first software development*. O'Reilly Media, Sebastopol, 2008.
- [17] J. Shore and S. Warden. *A arte do desenvolvimento ágil*. Alta Books, Rio de Janeiro, 2008.
- [18] V. Subramaniam and A. Hunt. *Practices of an agile developer*. Pragmatic Bookshelf, Dallas, 2006.
- [19] R. Tori, C. Kirner, and R. Siscoutto, editors. *Fundamentos e tecnologia de realidade virtual e aumentada*. SBC, Porto Alegre, 2006.
- [20] K. Waters. Extreme programming versus scrum. Disponível em: <<http://www.agile-software-development.com/2008/extreme-programming-versus-scrum>>, 2008. Acesso em: 08 abr. 2009.