

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**APLICAÇÃO DO PROCESSO ÁGIL DE
GERENCIAMENTO *SCRUM* NO DESENVOLVIMENTO
DE UM JOGO DIGITAL – ESTUDO DE CASO EM
EMPRESA DE *SOFTWARE***

TRABALHO DE GRADUAÇÃO

Nº 260

Tiago Keller Ferreira

**Santa Maria, RS, Brasil
2008**

**APLICAÇÃO DO PROCESSO ÁGIL DE GERENCIAMENTO
SCRUM NO DESENVOLVIMENTO DE UM JOGO DIGITAL –
ESTUDO DE CASO EM EMPRESA DE SOFTWARE**

por

Tiago Keller Ferreira

Trabalho de Graduação apresentado ao curso de Ciência da Computação
– Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS),
como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof. Dr. Eng. Marcos Cordeiro d'Ornellas

**Santa Maria, RS, Brasil
2008**

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**APLICAÇÃO DO PROCESSO ÁGIL DE GERENCIAMENTO *SCRUM* NO
DESENVOLVIMENTO DE UM JOGO DIGITAL – ESTUDO DE CASO EM
EMPRESA DE *SOFTWARE***

elaborado por
Tiago Keller Ferreira

Como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA

Prof. Dr. Eng. Marcos Cordeiro d'Ornellas (UFSM)
(Presidente / Orientador)

Prof. Dr. Eng. Felipe Martins Müller (UFSM)

Prof. Dra. Lisandra Manzoni Fontoura (URI)

Santa Maria, 31 de janeiro de 2008

DEDICATÓRIA

À minha esposa, ao meu filho, aos meus pais e a Decadium Studios.

AGRADECIMENTOS

Agradeço acima de tudo a Deus, por conceber vida a todos nós; por conceder os amigos, as idéias, e providenciar os meios e recursos para que esse trabalho se realizasse.

Agradeço a minha família, especialmente a minha esposa e ao meu filho, por estarem sempre junto aos meus desafios e por me manterem firme aos meus objetivos.

Agradeço a Decadium Studios, meus sócios e colaboradores, pela compreensão e ajuda na realização desse trabalho.

E por fim, agradeço aos professores e ao meu orientador, pela disposição para esclarecimentos de dúvidas e orientações.

“É melhor ouvir as palavras calmas de uma pessoa sábia do que os gritos de um líder numa reunião de tolos”.
(Salomão)

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

APLICAÇÃO DO PROCESSO ÁGIL DE GERENCIAMENTO *SCRUM* NO DESENVOLVIMENTO DE UM JOGO DIGITAL – ESTUDO DE CASO EM EMPRESA DE *SOFTWARE*

Autor: Tiago Keller Ferreira
Orientador: Prof. Dr. Eng. Marcos Cordeiro d'Ornellas
Local e Data de Defesa: Santa Maria, 31 de Janeiro de 2008

O presente trabalho tem como finalidade a aplicação do método ágil de gerenciamento *Scrum* no desenvolvimento de um jogo digital. Para tal, apresenta um estudo de caso no qual o método *Scrum* foi aplicado no processo de criação de um jogo 3D, desenvolvido em uma empresa de software. Além do estudo de caso, esse trabalho explica e apresenta as principais características do método de gerência *Scrum*. O trabalho está esquematizado em três partes tratando respectivamente de: contextualização (Introdução; *Scrum* processo ágil de gerenciamento; Projeto Arena Dalponte); estudo de caso e desenvolvimento (Estudo de caso – aplicação *Scrum*; Produto Final); e a conclusão (Conclusão).

Palavras-chave: *scrum*; jogos digitais; processos ágeis; engenharia de *software*.

ABSTRACT

Trabalho de Graduação
Computer Science
Universidade Federal de Santa Maria

AGILE PROCESS SCRUM FOR DEVELOPMENT OF A DIGITAL GAME - CASE STUDY IN SOFTWARE HOUSE

Author: Tiago Keller Ferreira
Advisor: Prof. Dr. Eng. Marcos Cordeiro d' Ornellas
Place and Date of Presentation: Santa Maria, 31 the January, 2008

This work has the purpose of applying the agile method of administration Scrum in the development of a game. For such, the work shows a case of study where the Scrum method was applied in a 3D game creating, developed in a software house. In addition, this work explains and presents the basic characteristics concerning Scrum management method. The work is organized in three parts, as follows: introduction (Introdução; *Scrum* processo ágil de gerenciamento; Projeto Arena Dalponte); case of study and development (Estudo de caso – aplicação *Scrum*; Produto Final); and conclusion (Conclusão).

Keywords: scrum; games; agile development; software engineer

LISTA DE FIGURAS

Ilustração 1 – <i>Scrum</i> em um jogo de <i>Rugby</i>	18
Ilustração 2 – Lista de funcionalidades do produto	19
Ilustração 3 – Reunião Diária	20
Ilustração 4 – Lista de funcionalidades do ciclo iterativo	23
Ilustração 5 – Gráfico de andamento - Relação horas por ciclo iterativo	23
Ilustração 6 – Ciclo do <i>Scrum</i>	24
Ilustração 7 – Fluxo do jogo	27
Ilustração 8 – Controle de jogo universal	28
Ilustração 9 – Arte de referência para o projeto	29
Ilustração 10 – Lista de funcionalidades do ciclo iterativo Arena Dalponte	37
Ilustração 11 – Gráfico de andamento do ciclo iterativo 1	39
Ilustração 12 – Gráfico de andamento do ciclo iterativo 2	41
Ilustração 13 – Gráfico de andamento do ciclo iterativo 3	43
Ilustração 14 – Gráfico de andamento do ciclo iterativo 4	45
Ilustração 15 – Gráfico de andamento do ciclo iterativo 5	47
Ilustração 16 – Gráfico de andamento final do projeto	48
Ilustração 17 – Crescimento em horas totais do projeto.....	49
Ilustração 18 – Tarefas adicionadas a lista de funcionalidades.....	50
Ilustração 19 – Porcentagem de tarefas adicionadas.....	51
Ilustração 20 – Rendimento em horas da equipe	52
Ilustração 21 – Tela de escolha de jogador.....	55
Ilustração 22 – Tela de jogo	56

LISTA DE TABELAS

Tabela 1 – Lista de funcionalidades do produto	34
Tabela 2 – Lista de funcionalidades do ciclo iterativo 1	38
Tabela 3 – Lista de funcionalidades do ciclo iterativo 2	40
Tabela 4 – Lista de funcionalidades do ciclo iterativo 3	42
Tabela 5 – Lista de funcionalidades do ciclo iterativo 4	44
Tabela 6 – Lista de funcionalidades do ciclo iterativo 5	46

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

GDC – *Game Developers Conference*

IA – *Inteligência Artificial*

JME – *jMonkey Engine*

MSF – *Microsoft Solution Framework*

RUP – *Rational Unified Process*

XML – *eXtensible Markup Language*

XP – *eXtreme Programming*

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Objetivos desse trabalho	15
1.2 Organização do texto	15
2 SCRUM PROCESSO ÁGIL DE GERENCIAMENTO	17
2.1 Origem do termo <i>Scrum</i>	17
2.2 Estrutura do método	18
2.2.1 Lista de funcionalidades do produto	19
2.2.2 Reunião diária	19
2.2.3 Ciclo iterativo	21
2.2.4 Reunião de planejamento do ciclo iterativo	21
2.2.5 Lista de funcionalidades do ciclo iterativo	22
2.2.6 Reunião final do ciclo iterativo	23
2.3 Visão geral do método	24
3 PROJETO ARENA DALPONTE	25
3.1 Conceito do jogo	25
3.1.1 Resumo	25
3.1.2 Regras do Jogo	26
3.1.3 Interface	26
3.1.4 Controles	27
3.1.5 Inteligência Artificial	28
3.1.6 Arte Conceitual	28
3.2 Processo de desenvolvimento do jogo	29
3.3 Porque utilizar <i>Scrum</i>	30
4 ESTUDO DE CASO - APLICAÇÃO DO SCRUM	32
4.1 A equipe do projeto Arena Dalponte	32
4.2 <i>Scrum</i> na empresa	33
4.3 Lista de funcionalidades do produto	34
4.4 Ciclos iterativos	36
4.4.1 Ciclo iterativo 1	37

	13
4.4.2 Ciclo iterativo 2.....	39
4.4.3 Ciclo iterativo 3.....	41
4.4.4 Ciclo iterativo 4.....	43
4.4.5 Ciclo iterativo 5.....	45
4.5 Análise dos gráficos finais do projeto.....	47
4.5.1 Gráfico de andamento do projeto	47
4.5.2 Tarefas adicionadas	49
4.5.3 Rendimento em horas dos colaboradores.....	51
4.6 Ganhos com Scrum no projeto	52
5 PRODUTO FINAL E DETALHES TÉCNICOS	54
6 CONCLUSÃO	57
7 REFERÊNCIAS BIBLIOGRÁFICAS	58
8 OUTRAS REFERÊNCIAS BIBLIOGRÁFICAS	60

1 INTRODUÇÃO

Quando um *software* de computador é bem-sucedido – ou seja, satisfaz as necessidades das pessoas que o usam, tem desempenho sem falhas por um longo período, é fácil de modificar e ainda mais fácil de usar – ele pode e efetivamente modifica as coisas para melhor. Mas, quando o *software* falha – ou sejam, seus usuários ficam insatisfeitos, quando tem tendência a erros, quando é difícil de modificar e ainda mais difícil de usar – podem e efetivamente acontecem coisas desagradáveis. Todos nós desejamos construir *softwares* que tornem as coisas melhores evitando os problemas que espreitam na sombra dos esforços mal sucedidos. Para obter sucesso, precisamos de disciplina quando o *software* é projetado e construído, sendo assim, precisamos de uma abordagem de engenharia (PRESSMAN, 2005, p.1).

Um modelo de processo de desenvolvimento de software, pode ser visto como uma representação, ou abstração dos objetos e atividades envolvidas no desenvolvimento de um software. Além disso, oferece uma forma mais abrangente e fácil de representar o gerenciamento do desenvolvimento de software e conseqüentemente o progresso do projeto.

Existem vários modelos de processo de *software*. Cada um representa uma tentativa de colocar ordem em uma atividade inerentemente caótica. A indústria de desenvolvimento de jogos digitais não foge a essa regra. Por se tratar de um sistema onde é necessária uma fase de planejamento e design, seguida por uma fase de prototipagem, seguida por uma fase de produção, seguida por uma fase de teste e, por fim, o término do projeto, o desenvolvimento de jogos digitais também necessita de práticas e processos de engenharia e gerência para colocar em ordem suas atividades.

A indústria de jogos, conforme a *Game Developers Conference* (GDC) de 2004 (LLOPIS, 2004) e recentemente a GDC 2007 (WILSON, 2007), tem concentrado seus esforços e investimentos em soluções de engenharia e gerência em desenvolvimento de jogos digitais. Como tendência natural, as empresas de desenvolvimento de jogos digitais estão buscando por soluções de desenvolvimento ágil focando principalmente em *eXtreme Programming (XP)* e *Scrum*.

Processos ágeis são métodos de desenvolvimento de produtos usando interações curtas. O *Scrum* é utilizado atualmente como fundamento para os mais importantes *frameworks* de processos de desenvolvimento de *software* do mercado. Ele é a base da disciplina de gestão de projetos da instância *open source* do *Rational Unified Process (RUP)*, o *OpenUP (OPENUP, 2007)*. É também utilizado como base da gestão de projetos do processo da Microsoft, o *Microsoft Solutions Framework (MSF) Agile (MICROSOFT DEVELOPER NETWORK, 2007)*. Além disso, está sendo muito utilizado nos EUA e Europa, inclusive e recentemente com muita força na indústria de desenvolvimento de jogos eletrônicos e na indústria de *software*.

1.1 Objetivos desse trabalho

Tendo em vista que o processo *Scrum*, com base no texto acima, é um dos processos ágeis que estão em evidência no mercado, principalmente na indústria de jogos digitais, esse trabalho irá apresentar um estudo de caso onde aplicará o processo de gerenciamento *Scrum* em um projeto de desenvolvimento de jogo digital. O estudo de caso será aplicado em um projeto de uma empresa de desenvolvimento de jogos na qual tenho função de gerente de projetos, a Decadium Studios.

A Decadium tem um projeto de um jogo de futebol 3D que deve ser desenvolvido em cinco semanas, e, atualmente, a empresa não utiliza nenhuma metodologia de gerência ágil. O objetivo desse estudo de caso é de aplicar, documentar e relatar o processo *Scrum* no desenvolvimento de um jogo digital. O jogo que será desenvolvido durante o processo não é o foco principal desse estudo de caso, e não serão avaliadas considerações técnicas do desenvolvimento tecnológico desse *software*, assim como não é objetivo desse trabalho fazer uma comparação entre métodos de gerência de desenvolvimento de *software*.

1.2 Organização do texto

O texto encontra-se organizado sob seções, sendo que na seção 2 há uma explicação sobre a definição de um processo ágil *Scrum*, a fim de contextualizar o trabalho acerca dos assuntos relevantes ao mesmo. Na seção 3, são explorados os detalhes e necessidades do produto que foi encomendado à Decadium Studios. Na seção 4, encontra-se o desenvolvimento principal desse trabalho, relatando como foi aplicado o método *Scrum* no desenvolvimento de um jogo digital. Na seção 5, há uma breve demonstração do produto final desenvolvido e especificações técnicas do projeto, e por fim, na seção 6 são apresentadas as conclusões acerca do trabalho.

2 SCRUM PROCESSO ÁGIL DE GERENCIAMENTO

O método ágil *Scrum* tem como objetivo definir um processo de gerenciamento de projetos onde há grande probabilidade de mudanças em seus requisitos. Esse processo baseia-se em princípios como: equipes pequenas de, no máximo sete pessoas; requisitos que são pouco estáveis ou desconhecidos; iterações curtas e práticas incrementais (SCHWABER, 2004). O *Scrum* é considerado um método específico para o gerenciamento de desenvolvimento de software. O método tem sua base fundamentada nos princípios de desenvolvimento ágil. Conforme o manifesto ágil, temos:

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho, passamos a valorizar:

Indivíduos e interação entre eles mais que processos e ferramentas.

Produto em funcionamento mais que documentação abrangente.

Colaboração com o cliente mais que negociação de contratos.

Responder a mudanças mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda (AGILE MANIFESTO, 2001).

2.1 Origem do termo *Scrum*

A palavra *Scrum* está associada ao jogo inglês *Rugby* que teve sua origem em 1823 (ASSOCIAÇÃO BRASILEIRA DE RUGBY, 2008). O *Scrum*, em um jogo de *Rugby*, acontece quando os dois times se concentram em um objetivo comum, que é a posse da bola. Este é um momento de definição e que ocorre diversas vezes durante o jogo, e podemos visualizar um *Scrum* em uma partida de *Rugby* na Ilustração 1. O dicionário Babylon (BABYLON, 1997) define o termo *Scrum* como “formação concentrada de jogadores que lutam para tomar posse da bola (*Rugby*)”.

O *Scrum* como método de gerência de desenvolvimento, também representa de modo figurado um *Scrum* de *Rugby*, onde uma equipe está focada em um único objetivo, e durante todo o jogo continua buscando esse objetivo. A utilização da palavra

Scrum para o método se deve por esse fato, que consegue de forma simples e resumida exemplificar o que o método *Scrum* propõe.



Ilustração 1 – *Scrum* em um jogo de *Rugby*

2.2 Estrutura do método

Nessa seção será detalhada a estrutura e como funciona o método *Scrum*.

O *Scrum* não requer ou fornece qualquer técnica ou método específico para a fase de desenvolvimento de software, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto (JEVEAUX, 2007).

As práticas gerenciais do *Scrum* podem ser divididas em:

- a) Lista de funcionalidades do produto (*Product backlog*);
- b) Reuniões diárias (*Daily Scrum*);
- c) Ciclo iterativo (*Sprint*);
- d) Reunião de planejamento do ciclo iterativo (*Sprint planning meeting*);
- e) Lista de funcionalidades do ciclo iterativo (*Sprint backlog*);
- f) Reunião final do ciclo iterativo (*Sprint review meeting*).

As pessoas envolvidas no *Scrum* são classificadas como:

- a) Equipe (*Scrum team*);
- b) Cliente (*Product owner*);
- c) Coordenador *Scrum* (*Scrum master*).

2.2.1 Lista de funcionalidades do produto

A lista de funcionalidades do produto contém todas as funcionalidades desejadas para um produto ou *software*. O conteúdo desta lista é definido pelo cliente, o qual deve participar no processo de desenvolvimento. A lista de funcionalidades do produto não precisa estar completo no início de um projeto. Pode-se começar com tudo aquilo que é mais óbvio em um primeiro momento. Com o tempo, a lista de funcionalidades do produto cresce e muda à medida que se aprende mais sobre o produto e seus usuários (TELES, 2007). Na Ilustração 2 pode-se visualizar uma lista de funcionalidades do produto.

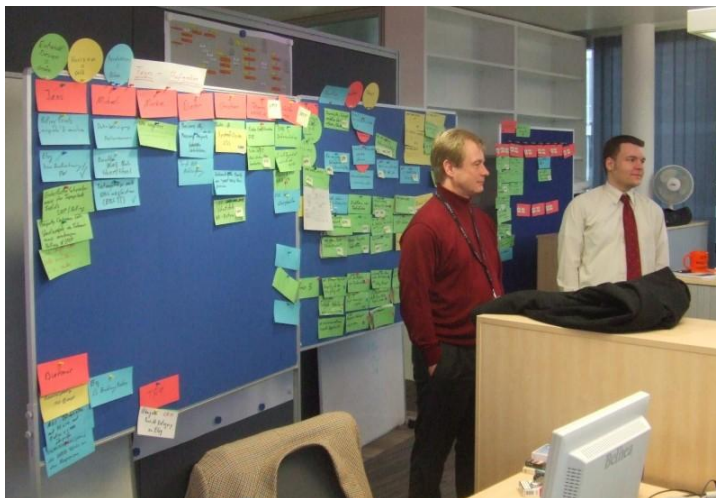


Ilustração 2 – Lista de funcionalidades do produto

2.2.2 Reunião diária

As reuniões diárias têm como objetivo disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia. As reuniões diárias preferencialmente devem ser realizadas no mesmo lugar e na mesma hora do dia. Idealmente são realizadas na parte da manhã, para ajudar a estabelecer as prioridades do novo dia de trabalho. Todos os membros da equipe devem participar das reuniões diárias. Outras pessoas também podem estar presentes, mas só poderão escutar. Isso torna as reuniões diárias uma excelente forma para uma equipe disseminar informações sobre o estado do projeto (TELES, 2007).

Não é propósito das reuniões diárias serem usadas como uma reunião para resolução de problemas. Ela deve ser uma reunião rápida de até 15 minutos e com todos os participantes em pé, como mostrado na Ilustração 3. Questões levantadas devem ser levadas para fora da reunião e normalmente tratadas por um grupo menor de pessoas que tenham a ver diretamente com o problema ou possam contribuir para solucioná-lo.



Ilustração 3 – Reunião Diária

Durante a reunião diária, cada membro da equipe provê respostas para cada uma destas três perguntas:

- a) O que você fez ontem?
- b) O que você fará hoje?
- c) Há algum impedimento no seu caminho?

Concentrando-se no que cada pessoa fez ontem e no que ela irá fazer hoje, a equipe ganha uma excelente compreensão sobre que trabalho foi feito e que trabalho ainda precisa ser feito. A reunião diária não é uma reunião de reporte na qual um chefe fica coletando informações sobre quem está atrasado. Ao invés disso, é uma reunião na qual membros da equipe assumem compromissos perante os demais. Os impedimentos identificados na reunião diária devem ser tratados pelo coordenador *Scrum* o mais rapidamente possível (TELES, 2007).

O coordenador *Scrum* é basicamente o gerente do projeto, porém com um enfoque diferente. Ao invés de delegar e cobrar por tarefas, ele trabalha mais como um facilitador e protetor da equipe. É papel do coordenador *Scrum* não permitir que a equipe se comprometa excessivamente e remover os obstáculos que aparecem durante o *ciclo iterativo*. (JEVEAUX, 2007).

2.2.3 Ciclo iterativo

O termo original utilizado para ciclo iterativo é *Sprint*. Não há uma palavra na língua portuguesa que corresponda à palavra *Sprint*. Conforme o Dicionário Collins Gem (MIKE, 1992, p.282) *Sprint* é definido como: “corrida de pequena distância, correr a toda velocidade”.

Os ciclos iterativos são definidos em um espaço de tempo de duas a quatro semanas, no qual um conjunto de funcionalidades extraídas da lista de funcionalidades do produto são passadas para a lista de funcionalidades do ciclo iterativo e devem ser concluídas até o período final estabelecido pelo ciclo iterativo.

2.2.4 Reunião de planejamento do ciclo iterativo

A Reunião de planejamento do ciclo iterativo é realizada sempre antes de cada ciclo iterativo. Durante essa reunião, o cliente descreve as funcionalidades de maior prioridade para a equipe. A equipe faz perguntas durante a reunião de modo que seja capaz de quebrar as funcionalidades em tarefas técnicas, após a reunião. Essas tarefas darão origem a lista de funcionalidades do ciclo iterativo.

2.2.5 Lista de funcionalidades do ciclo iterativo

Segundo Teles (2007), a lista de funcionalidades do ciclo iterativo são extraídas da lista de funcionalidades do produto, pela equipe, com base nas prioridades definidas pelo cliente e a percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades, pode-se visualizar uma lista de funcionalidades do ciclo iterativo na Ilustração 4. Cabe à equipe determinar a quantidade de itens da lista de funcionalidades do produto que serão trazidos para a lista de funcionalidades do ciclo iterativo, já que é ela quem irá se comprometer a implementá-los. Durante um ciclo iterativo, o coordenador *Scrum* mantém a lista de funcionalidades do ciclo iterativo atualizando-a para refletir que tarefas são completadas e quanto tempo a equipe acredita que será necessário para completar aquelas que ainda não estão prontas. A estimativa do trabalho que ainda resta a ser feito no ciclo iterativo é calculada diariamente e colocada em um gráfico, resultando em um gráfico de andamento conforme modelo na Ilustração 5.



Ilustração 4 – Lista de funcionalidades do ciclo iterativo

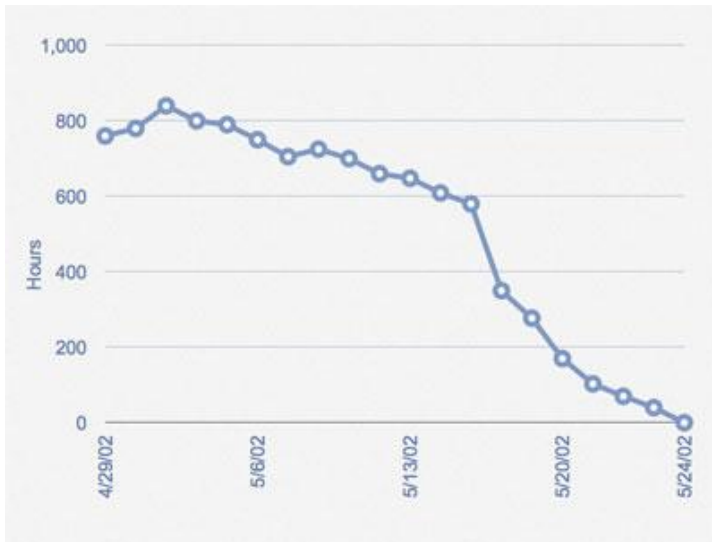


Ilustração 5 – Gráfico de andamento - Relação horas por ciclo iterativo

2.2.6 Reunião final do ciclo iterativo

Ao final de cada ciclo iterativo é feita uma reunião final do ciclo iterativo. Durante esta reunião, a equipe mostra o que foi alcançado durante o ciclo iterativo. Tipicamente, isso tem o formato de uma demonstração das novas funcionalidades.

2.3 Visão geral do método

Na Ilustração 6 pode-se contemplar uma visão geral que demonstra todo o ciclo de processo ágil iterativo *Scrum*.

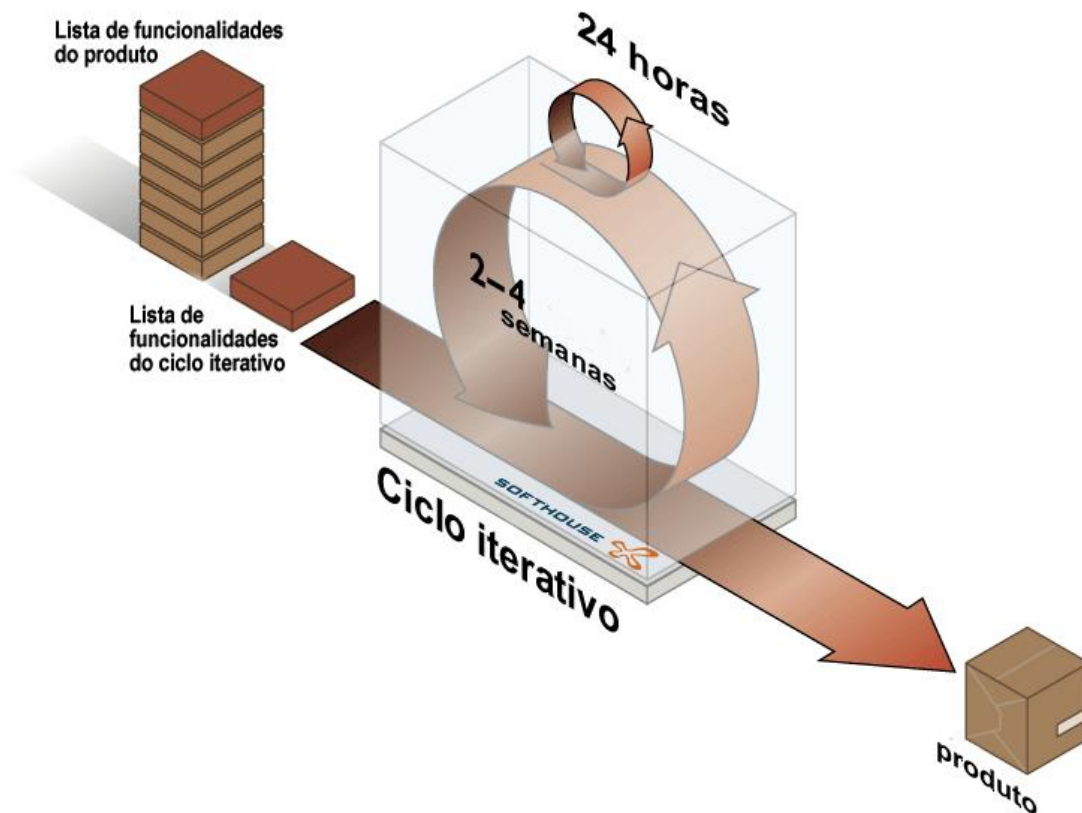


Ilustração 6 – Ciclo do *Scrum*

3 PROJETO ARENA DALPONTE

Nesta seção estão detalhados os requisitos e especificações de um jogo digital desenvolvido pela Decadium Studios. Por motivos contratuais e por se tratar de um produto que foi disponibilizado ao mercado, alguns dados como valores financeiros e informações de contrato não serão especificados nesse relatório.

A Arena Dalponte é um projeto de marketing e propaganda da empresa Dalponte Calçados. O projeto teve como objetivo realizar um campeonato digital de futebol 3D, onde os compradores dos calçados Dalponte participaram de um campeonato de jogo de futebol 3D com a marca Dalponte.

A Dalponte Calçados contratou a Decadium Studios para o desenvolvimento do jogo para o evento. A Decadium Studios é uma empresa que tem seu foco de atuação no desenvolvimento de jogos digitais, participante da Incubadora Tecnológica de Santa Maria, na qual atuou como gerente de desenvolvimento.

O projeto, conforme contrato estipulado, determinou o prazo de cinco semanas para o desenvolvimento completo do projeto.

3.1 Conceito do jogo

Nessa subseção estão descritos os requisitos conceituais do projeto Arena Dalponte solicitados pelo cliente e discutidos junto à Decadium Studios. A seguir a descrição conceitual e requisitos do projeto o para jogo de futebol de salão 3D.

3.1.1 Resumo

Conforme documento firmado entre o cliente e a Decadium Studios, ficou definido desenvolver um jogo de futebol 3D para computador em tela cheia, *multiplayer* e com a aplicação de jogo controlada a partir de um servidor central. Todo o campeonato foi controlado por um servidor, que teve a obrigação de estipular as chaves do campeonato, indicar próximos jogos, liberar acessos e armazenar os resultados dos jogos de cada uma das estações de jogo. O prazo de desenvolvimento foi de cinco semanas.

3.1.2 Regras do Jogo

O jogo se baseou nas regras e especificações oficiais da Confederação Brasileira de Futsal, tendo o uso obrigatório de cinco jogadores para cada time e a quadra com as dimensões oficiais. Algumas regras especiais foram especificadas para se adequar ao evento proposto e pela limitação do tempo de desenvolvimento.

Regras do jogo:

- a) Partidas com tempo único de cinco minutos;
- b) Não há lateral;
- c) Não há falta;
- d) Não há escanteio;
- e) Não há impedimento;
- f) Não há juiz;
- g) Goleiro controlado pelo computador.

Critérios de desempate em ordem de prioridade:

- a) Número de gols;
- b) Número de chutes a gol;
- c) Tempo de posse de bola.

3.1.3 Interface

A interface do jogo é simples e seu fluxo controlado pelo servidor. A interface do jogo está dividida em três estados conforme a Ilustração 7.



Ilustração 7 – Fluxo do jogo

No estado de escolha, o jogador pode confirmar sua presença e escolher um dos uniformes Dalponte. O jogador irá ficar nesse estado até o servidor mandar o sinal para início de jogo.

No estado de jogo, com a duração de cinco minutos, cada um dos jogadores irá controlar seu time, e ao final a aplicação irá enviar ao servidor as informações do jogo e o mesmo irá progredir para o próximo estado.

No estado de resultados, são mostradas as estatísticas do jogo e se aponta o jogador vencedor. O servidor organiza os próximos jogos e depois de determinado tempo redireciona as estações de jogo novamente para o estado de escolha.

3.1.4 Controles

O jogador tem as seguintes ações disponibilizadas no jogo:

- a) Movimentar o jogador pela quadra;
- b) Correr;
- c) Chutar;

- d) Passar a bola;
- e) Roubar a bola;
- f) Trocar de jogador;

O jogador humano pode controlar seus jogadores digitais com o uso de um *joystick* universal padrão, conforme o modelo da Ilustração 8.



Ilustração 8 – Controle de jogo universal

3.1.5 Inteligência Artificial

A inteligência artificial do jogo é simples. O goleiro é controlado pelo computador e operará sobre um sistema randômico de defesa com base no tipo e dados do chute. Não haverá time controlado pelo computador, os jogadores digitais que não são momentaneamente controlados pelo jogador humano, têm uma inteligência básica de aproximação da bola e do jogador digital controlado pelo jogador humano, para realizar alguma ação esses jogadores virtuais devem ser selecionados pelo jogador humano.

3.1.6 Arte Conceitual

O jogo apresenta uma arte conceitual humanóide, mas não tem a pretensão de ser tão ricamente animado e graficamente pesado como os jogos de futebol lançados comercialmente. Foi requisitada uma arte gráfica de boa qualidade e com a utilização de recursos computacionais atuais, porém limitando-se a um *advergame* (jogos com fins publicitários). Como referência de arte conceitual o cliente passou a Ilustração 9 à Decadium Studios, como base mínima em termos de visualização e gráficos.



Ilustração 9 – Arte de referência para o projeto

3.2 Processo de desenvolvimento do jogo

Por se tratar de um *advergame*, algumas fases importantes de desenvolvimento de jogos foram eliminadas, tais como: *storyboard*; narração; ajuda e interface customizada. Antes de se definir o plano de desenvolvimento, a Decadium Studios levantou os requisitos e as possíveis fases de desenvolvimento para estipulação de

custos, e o levantamento inicial, com base no conceito do jogo, foi definido da seguinte maneira:

- a) Reuniões com cliente.
- b) Game Design;
- c) Planejamento técnico e modelagem;
- d) Programação da interface;
- e) Programação do jogo;
- f) Programação do servidor;
- g) Programação de sons;
- h) Programação de efeitos;
- i) Criação de arte conceitual;
- j) Criação da arte 3D;
- k) Animações;
- l) Composição de sons e músicas;
- m) Testes;

3.3 Porque utilizar *Scrum*

O ponto crítico do projeto Arena Dalponte com certeza foi o prazo de desenvolvimento de cinco semanas. No mercado de desenvolvimento de *advergames*, o tempo médio para produção de jogos digitais 3D é de três à seis meses, sendo assim, a Decadium Studios tinha em suas mãos o desafio de conseguir desenvolver o jogo digital em tempo hábil. Até então, a Decadium Studios não havia feito nenhum projeto de jogo completo e em 3 dimensões em menos de três meses, e não adotava nenhum tipo de metodologia ágil para desenvolvimento e gerência.

Como sócio da Decadium Studios e finalizando meu curso de graduação pensando em me aprofundar na área de engenharia de software, decidi aliar meu projeto de graduação ao problema da empresa, sugerindo a utilização de uma nova metodologia de gerência de projetos de desenvolvimento de jogos capaz de trazer

resultados satisfatórios ao projeto Arena Dalponte da Decadium Studios. Como solução para o problema tivemos dois focos principais:

- a) Utilizar o método *Scrum* para gerenciamento e relacionamento com o cliente;
- b) Enxugar as funcionalidades do projeto e trabalhar com uma lista de funcionalidades de produto compacta e que suportasse modificações futuras do cliente.

A escolha do método *Scrum* para gerência do projeto se deve ao fato de a empresa já ter uma pequena experiência com *XP* e pesquisa realizada junto a empresas de desenvolvimento de jogos digitais no Brasil e no exterior.

Estudos acadêmicos (BARROS, 2007), conferências mundiais da indústria de jogos digitais (LLOPIS, 2004) (WILSON, 2007), e relatórios da Associação Brasileira de Desenvolvedores de Jogos Eletrônicos (ABRAGAMES, 2004) apontam para uma crescente utilização de métodos ágeis no desenvolvimento de jogos digitais, principalmente dos métodos *Scrum* e *XP*.

Com as informações coletadas, e com uma apresentação resumida da metodologia *Scrum* à equipe de desenvolvimento e aos sócios da Decadium, a empresa decidiu aplicar o método de gerência *Scrum* no projeto Arena Dalponte, acreditando ser uma solução viável e necessária para realização do projeto Arena Dalponte.

4 ESTUDO DE CASO - APLICAÇÃO DO SCRUM

Nessa seção está descrito como o processo de gerência *Scrum* foi aplicado no projeto Arena Dalponte. Primeiramente será descrita a equipe formada para desenvolvimento do projeto e a implantação do *Scrum* na empresa, em seguida está lista de funcionalidades do produto, e logo após é detalhado cada um dos ciclos iterativos. Ao final, são repassados alguns detalhes específicos e gráficos de análise gerados no término do projeto. Na seção 6 tem-se a conclusão sobre todo o trabalho.

4.1 A equipe do projeto Arena Dalponte

No desenvolvimento do projeto Arena Dalponte, estabeleceu-se uma equipe formada por seis colaboradores:

- a) Coordenador *Scrum*, função realizada pela minha pessoa.
- b) Cliente, Vicente Souza, representante da Dalponte Calçados e sócio-gerente da empresa Quanta Studio, sediada em Porto Alegre, RS, Brasil.
- c) Carlos Edmilson da Silva Maia, membro da equipe, atuante na área de programação e computação gráfica.
- d) Fernando Bevilacqua, membro da equipe, qualificado na área de programação e efeitos 3D.
- e) Jeferson Ornelas Wendt, membro da equipe, especializado na área de programação e documentação.
- f) Ricardo Bess, membro da equipe, artista qualificado em ilustrações 2D e modelagem 3D.

Logo na definição da equipe, houve a necessidade de fazer alguns desvios no método de gerência *Scrum*. Preferencialmente em um projeto que adota o *Scrum* como método de gerenciamento, o cliente deve estar presente dentro do projeto pessoalmente durante todo o processo.

O primeiro problema que encontrado foi que o cliente do projeto tem sua sede de trabalho em Porto Alegre, sendo que toda a equipe de criação do jogo encontra-se em Santa Maria, numa distância de, aproximadamente, 300 KM. Não foi possível que o cliente estivesse presente durante o processo de desenvolvimento, por motivos financeiros e também pelo fato de o mesmo ter outros projetos em paralelo. A primeira solução encontrada foi a de o cliente participar remotamente das reuniões diárias (telefone ou Internet), e pessoalmente para as reuniões de planejamento e finalização de ciclo iterativo. Porém, a participação remota nas reuniões diárias logo se mostrou ineficiente, sendo necessário utilizar-se uma abordagem diferente, onde o coordenador *Scrum*, após a reunião diária repassaria, em uma reunião especial, o andamento do projeto ao cliente.

Durante todo o projeto tivemos cinco encontros com cliente para as reuniões de planejamento e finalização de ciclo iterativo. Desses encontros, duas vezes a equipe movimentou-se para Porto Alegre e nas outras três o cliente movimentou-se para Santa Maria.

4.2 Scrum na empresa

Um dos grandes desafios de qualquer processo ágil, principalmente o *Scrum*, é a implantação na prática do método e a aceitação da equipe de desenvolvimento. Não houve resistência para aplicação do novo método na Decadium Studios, a equipe já tinha conhecimento sobre essa necessidade e já se demonstrava comprometimento com o projeto, um dos pontos fortes e base do *Scrum*.

A Decadium Studios estabeleceu o horário das reuniões diárias às 18 horas. Normalmente no *Scrum* define-se o horário das reuniões diárias para o turno matutino, mas em nosso processo preferiu-se utilizar o horário vespertino, por se adequar melhor ao cronograma dos membros da equipe. Observou-se que não ocorreram problemas com o horário das reuniões diárias.

Além do comprometimento da equipe é necessário o envolvimento do cliente com o processo. Apesar do cliente não estar presente pessoalmente no processo, apresentou comprometimento com o método de gerência, demonstrando satisfação com o método e com os resultados gerados durante e ao final do projeto.

4.3 Lista de funcionalidades do produto

No dia 15 de outubro de 2007, ocorreu a primeira reunião do projeto onde se estabeleceram as funcionalidades e tarefas junto ao cliente. A lista de funcionalidades do produto normalmente é composta por funcionalidades básicas requisitadas pelo cliente, algo semelhante ao que temos descrito na subseção 3.1. Porém nesse projeto especificamos detalhadamente nossa lista de funcionalidades do produto em tarefas técnicas, pelo fato do cliente ter um conhecimento técnico regular e também para facilitar a montagem futura das listas de funcionalidades dos ciclos iterativos. Durante o levantamento dessas tarefas não se levou em consideração requisitos, prioridades, detalhes técnicos e questões de engenharia de *software*. Foram só colocadas e agrupadas, de forma geral, o que a equipe e o cliente acreditavam compor o projeto. Na Tabela 1 temos a lista das tarefas que definiram nossa lista de funcionalidades do produto inicial.

Tabela 1 – Lista de funcionalidades do produto

Título da tarefa	Horas
<i>Extensible Markup Language</i> (XML) do servidor	5
Programação do servidor	50
<i>Script</i> para gerar ordem dos jogos	10
Testes cliente-servidor	25
Estudo da <i>jMonkey Engine</i> (JME)	100

Titulo da tarefa	Horas
Planejamento dos estados de jogo	20
Modelagem em 3 dimensões da arena	25
Modelagem do jogador em 3 dimensões	30
Animação do jogador em 3 dimensões	10
Interface do usuário	10
Interface do servidor	5
Criação dos HUDs	10
Tela de escolha de personagens	10
Aplicar física a JME	5
Interface do <i>joystick</i>	15
Esqueleto do jogo	15
Posicionamento dos jogadores	10
Troca de personagens	10
Paredes imaginárias	5
Inicialização do cenário	5
Movimentação de personagens	15
Inteligência artificial	40
Converter texturas	5
Carregar modelos gráficos	5
Sistema de câmeras	15
Implementar passe	10
Implementar chute a gol	10
Implementar corrida	10
Retorno de chamadas da aplicação cliente	20
Estrutura dos estados de jogo	15
Pixel shaders	20
Fôlego em corrida	10
Cruzamento (baixa prioridade)	20
Implementar goleiro	30

Título da tarefa	Horas
Posse da bola	15
Vetor direção do jogador	5
Blocos físicos do cenário	10
Programar estados do jogo	30
Animações de game	50
Tela de resultados	20
Criação dos sons	30
Programação dos sons	20
Roubo de bola	5
Estados da bola	10
Textos da tela de resultado	5
Pontuação	5
Barra de força do chute	10
Reposição de bola	10
Contagem regressiva no jogo	5
Rodada de testes do jogo	100
Instalador	5

A lista de funcionalidades do produto inicial foi fechada com um total de 930 horas em tarefas.

Como o prazo de entrega é de cinco semanas, optou-se por dividir o processo de desenvolvimento em ciclos iterativos de uma semana. Com uma equipe ativa de desenvolvimento formada por cinco pessoas, cada ciclo iterativo teve uma carga limite de horas estipulada em até duzentas horas por ciclo iterativo.

4.4 Ciclos iterativos

Nessa seção estão detalhadas as tarefas executadas em cada ciclo iterativo. Na Ilustração 10, há uma foto da lista de funcionalidades do ciclo iterativo usado no projeto Arena Dalponte.

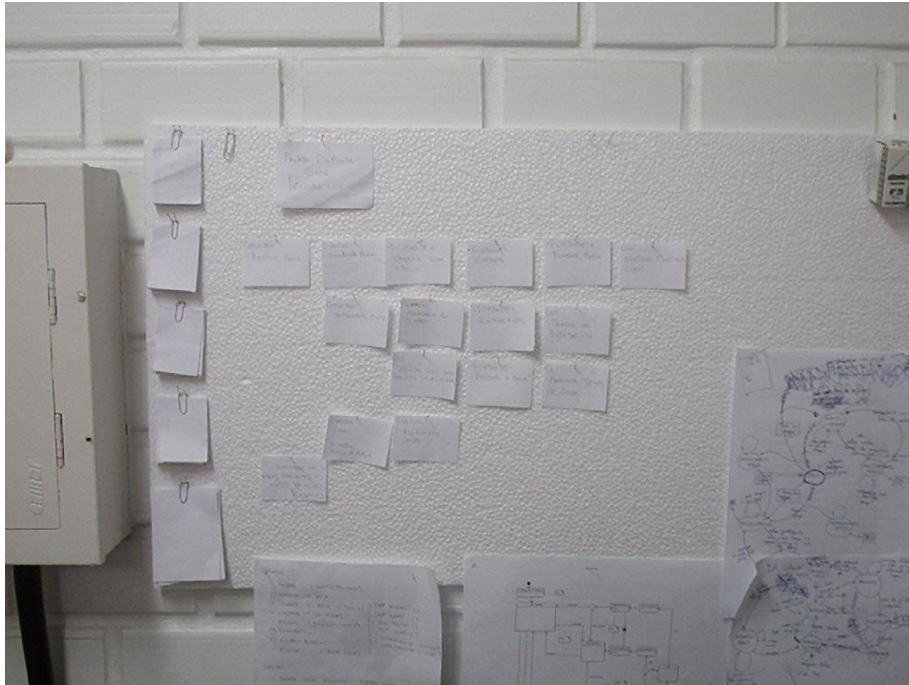


Ilustração 10 – Lista de funcionalidades do ciclo iterativo Arena Dalponte

4.4.1 Ciclo iterativo 1

O ciclo iterativo 1 foi realizado no período de 15 de outubro à 20 de outubro. E teve um total de 190 horas de tarefas retiradas da lista de funcionalidades do produto. Na Tabela 2 temos as tarefas da lista de funcionalidades do ciclo iterativo 1.

No ciclo iterativo 1 ocorreu tudo dentro do esperado, não houveram problemas com as tarefas estipuladas, e nenhuma tarefa adicional foi criada. Isso se deve ao fato que metade da carga de serviço do ciclo iterativo 1 estava em uma tarefa de estudo, o

que normalmente ocorre sem muitos percalços. Na Ilustração 11 podemos observar o gráfico de andamento do ciclo iterativo 1.

Tabela 2 – Lista de funcionalidades do ciclo iterativo 1

Título da tarefa	Dono da tarefa	Horas
<i>Extensible Markup Language</i> (XML) do servidor	Carlos	5
Script para gerar ordem dos jogos	Carlos	10
Estudo da <i>jMonkey Engine</i> (JME)	Todos	100
Planejamento dos estados de jogo	Jeferson	20
Interface do usuário	Tiago	10
Interface do servidor	Tiago	5
Criação dos HUDs	Ricardo	10
Tela de escolha de personagens	Ricardo	10
Estrutura dos estados de jogo	Jeferson	15
Vetor direção do jogador	Fernando	5

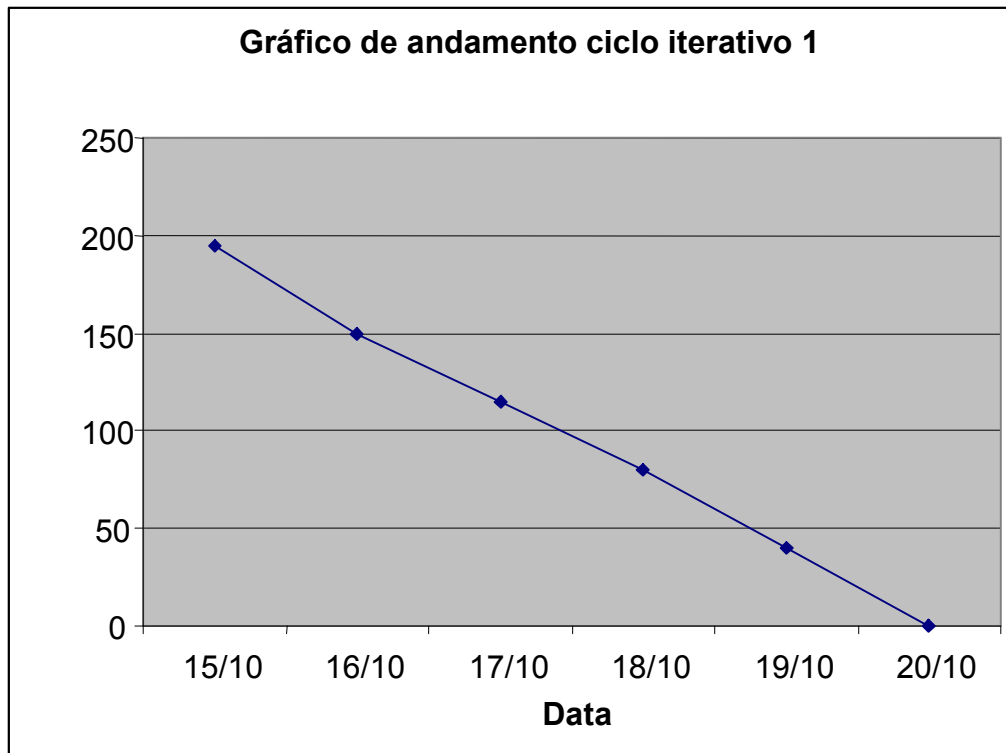


Ilustração 11 – Gráfico de andamento do ciclo iterativo 1

4.4.2 Ciclo iterativo 2

O ciclo iterativo 2 foi realizado no período de 22 a 27 de outubro, tendo um total de cento e noventa horas de tarefas retiradas da lista de funcionalidades do produto.

Na reunião de planejamento do ciclo iterativo duas tarefas foram anexadas a nossa lista de funcionalidades do produto, e a tarefa “Programação do servidor” teve sua estimativa de tempo reduzida de 50 horas para 40 horas. As tarefas adicionadas foram:

- a) Estudo da ferramenta de animação *MilkShape*;
- b) Conversão modelos MD3.

No decorrer do segundo ciclo iterativo, o cliente modificou as regras para geração das chaves do campeonato, com isso a tarefa “*Script* para gerar ordem dos jogos”, que já tinha sido executada, teve que ser novamente inserida na lista de

funcionalidades do produto. Para solução do problema ignoramos a tarefa nesse ciclo iterativo e colocamos em um ciclo iterativo posterior para não interferir na tarefa “Programação do servidor” no andamento do ciclo iterativo 2.

Ao fim do ciclo iterativo a tarefa “Modelagem 3D da arena” não foi concluída, o cliente não aprovou a arte e solicitou modificações, a tarefa foi estendida e passada para o próximo ciclo iterativo.

Na Tabela 3 é possível visualizar a lista de funcionalidades final do ciclo iterativo 2 e na Ilustração 12 o gráfico de andamento do segundo ciclo iterativo.

Tabela 3 – Lista de funcionalidades do ciclo iterativo 2

Título da tarefa	Dono da tarefa	Horas
Programação do servidor	Carlos e Tiago	40
Modelagem em 3 dimensões da arena	Ricardo	25
Estudo da ferramenta de animação	Ricardo	
<i>MilkShape</i>		5
Conversão dos Modelos MD3	Ricardo	5
Aplicar JME <i>physics</i>	Fernando	5
Esqueleto do jogo	Fernando	15
Paredes imaginárias	Tiago	5
Carregar modelos 3 dimensões de testes	Tiago	5
Sistema de câmeras	Tiago	15
Retorno de chamadas da aplicação	Fernando	
cliente		20
Blocos físicos do cenário	Jeferson	10
Programar estados de jogo	Jeferson	30

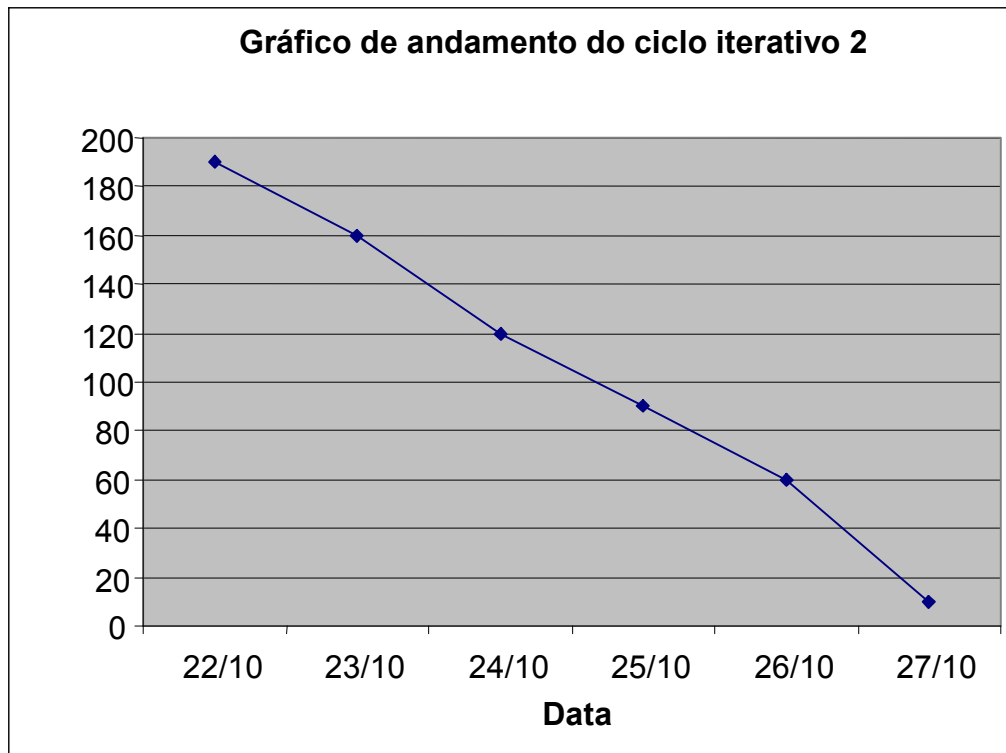


Ilustração 12 – Gráfico de andamento do ciclo iterativo 2

4.4.3 Ciclo iterativo 3

O ciclo iterativo 3 foi realizado no período de 29 a 03 de novembro. E teve um total de cento e noventa e cinco horas de tarefas retiradas da lista de funcionalidades do produto.

Na reunião de planejamento do ciclo iterativo tivemos quatro tarefas anexadas a lista de funcionalidades do produto. As tarefas adicionadas foram:

- a) Re-estruturar XML;
- b) Ajustes na modelagem da Arena;
- c) Conversão dos sons;
- d) Modelo físico da Bola.

A tarefa de “Re-estrutura XML” e “Ajustes na modelagem da Arena” foram adicionadas por serem uma necessidade apresentada no ciclo iterativo 2. A tarefa “Modelo físico” foi criada por ser um pré-requisito da tarefa “Estados da bola”.

Já a tarefa “Conversão de sons” foi inserida após a reunião de planejamento do ciclo iterativo, por ser uma necessidade apresentada pela tarefa “Criação dos sons”. A tarefa foi inserida durante o ciclo iterativo por conseguir se encaixar na lista de funcionalidades do ciclo iterativo sem prejudicar o andamento do mesmo.

Ao fim do ciclo iterativo todas as tarefas foram concluídas, e apesar de durante o ciclo apresentar tarefas com pré-requisitos, não houve atrasos e conflito de tarefas.

Na Tabela 4 podemos visualizar a lista de funcionalidades do ciclo iterativo 3 e na Ilustração 13 o gráfico de andamento do terceiro ciclo iterativo.

Tabela 4 – Lista de funcionalidades do ciclo iterativo 3

Título da tarefa	Dono da tarefa	Horas
Script para gerar ordem dos jogos	Carlos	5
Re-estruturar XML	Carlos	5
Testes cliente-servidor	Carlos	25
Ajustes na modelagem da Arena	Tiago	20
Modelagem do jogador 3 dimensões	Ricardo	30
Animação do jogador 3 dimensões	Ricardo	10
Converter texturas	Tiago	5
Carregar modelos 3 dimensões	Tiago	5
Vetor direção do jogador	Tiago	5
Conversão de sons	Jeferson	10
Criação dos sons	Jeferson	30
Implementar goleiro	Fernando	30
Modelo físico da Bola	Tiago	5
Estados da bola	Fernando	10

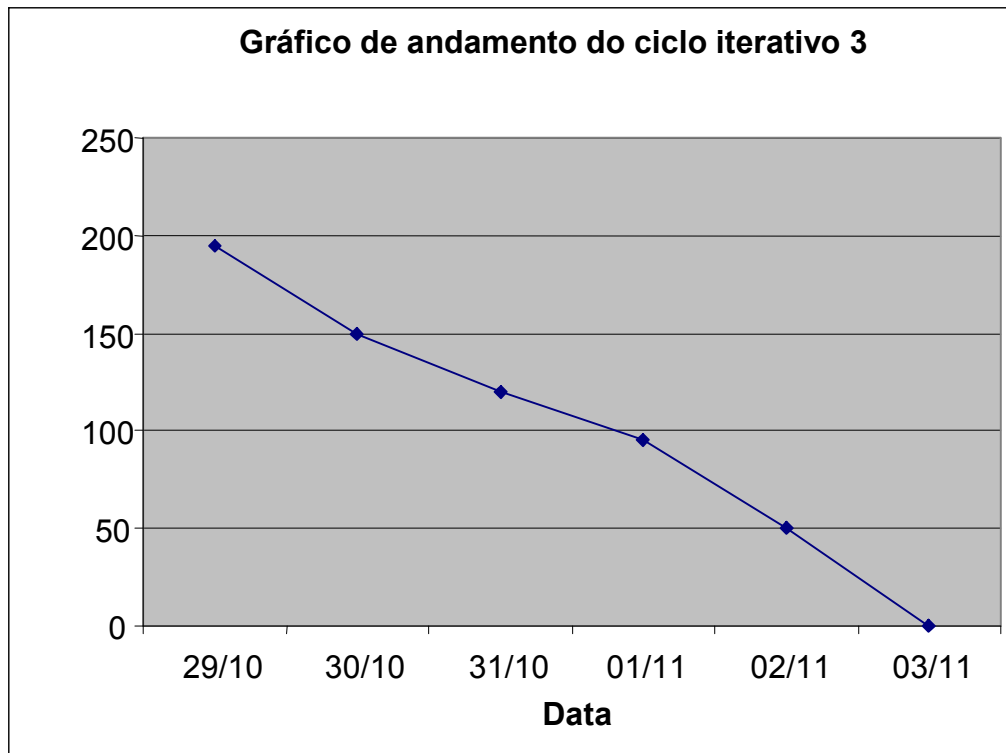


Ilustração 13 – Gráfico de andamento do ciclo iterativo 3

4.4.4 Ciclo iterativo 4

O ciclo iterativo 4 foi realizado no período de 05 a 10 de novembro. E teve um total de duzentas e quarenta horas de tarefas retiradas da lista de funcionalidades do produto.

Na reunião de planejamento do ciclo iterativo tive duas tarefas anexadas a lista de funcionalidades do produto. As tarefas adicionadas foram:

- a) Testes *Alpha* e levantamento final;
- b) Efeitos gráficos (Bola de fogo).

A tarefa “Testes *Alpha* e levantamento final” foi criada para estipular erros pequenos e funcionalidades que ainda não foram desenvolvidas, para serem ajustados no ciclo iterativo final. A tarefa “Efeitos gráficos (Bola de fogo)” foi uma exigência da Dalponte Calçados.

Ao fim do ciclo iterativo todas as tarefas foram concluídas, algumas funcionalidades e detalhes de algumas tarefas foram abortados, entretanto as tarefas foram consideradas prontas.

Na Tabela 5 pode-se visualizar a lista de funcionalidades final do ciclo iterativo 4 e na Ilustração 14 o gráfico de andamento do quarto ciclo iterativo.

Tabela 5 – Lista de funcionalidades do ciclo iterativo 4

Título da tarefa	Dono da tarefa	Horas
Posicionamento dos jogadores	Fernando	10
Troca de personagens	Fernando	10
Movimentação de personagens	Fernando	15
Inteligência Artificial	Carlos	40
Implementar passe	Jeferson	10
Implementar chute a gol	Jeferson	10
Implementar corrida	Jeferson	10
Posse da bola	Fernando	15
Animações de game	Tiago	50
Programação dos sons	Jeferson	20
Roubo de bola	Fernando	5
Testes <i>Alpha</i> e levantamento final	Todos	20
Efeitos gráficos (Bola de fogo)	Ricardo / Carlos	15
Barra de força do chute	Ricardo	10

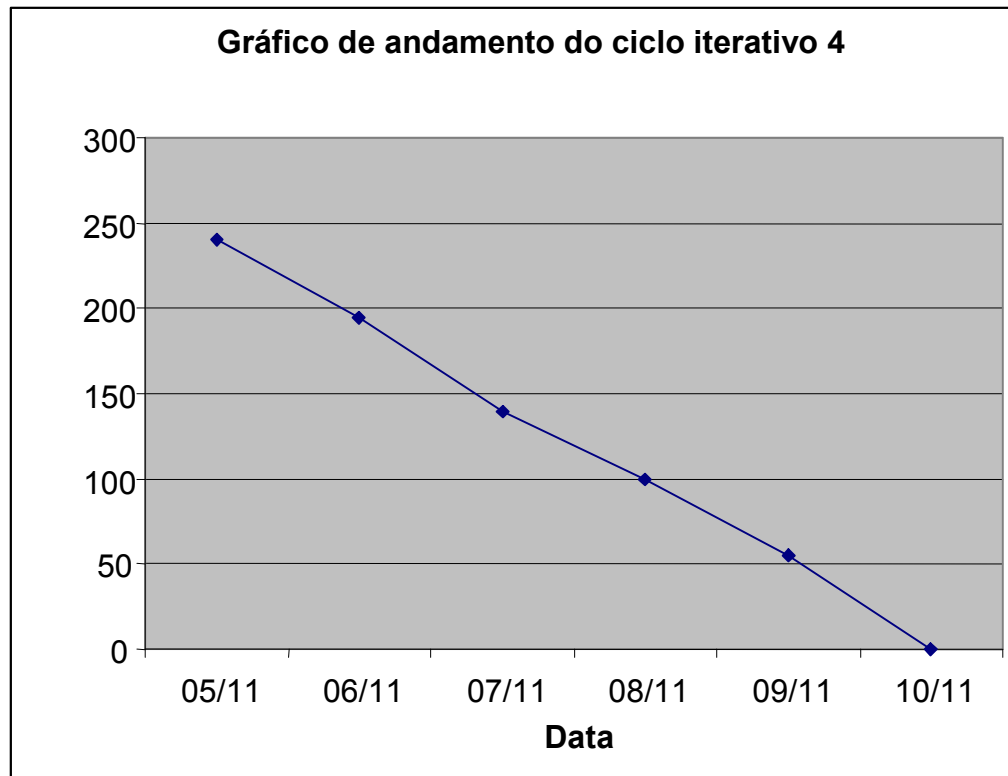


Ilustração 14 – Gráfico de andamento do ciclo iterativo 4

4.4.5 Ciclo iterativo 5

O quinto, e último, ciclo iterativo foi realizado no período de 12 à 21 de novembro. O último ciclo iterativo teve seu período alterado para nove dias e teve um total de duzentas e trinta horas de tarefas retiradas da lista de funcionalidades do produto.

Na reunião de planejamento do ciclo iterativo três tarefas foram anexadas a lista de funcionalidades do produto e três tarefas canceladas e retiradas da lista de funcionalidades do produto inicial.

As tarefas canceladas foram:

- a) Pixel shaders;
- b) Fôlego de corrida;

c) Cruzamento.

As tarefas adicionadas foram:

- a) Configurador de *Joystick*;
- b) Ajustes na movimentação;
- c) Aplicação para configurações.

Todas as tarefas adicionadas forem decorrentes de falhas ou necessidades encontradas no ciclo iterativo 4. As tarefas canceladas ocorreram por não haver mais tempo para desenvolvimento e foram consideradas de baixa prioridade.

Ao fim do ciclo iterativo todas as tarefas foram concluídas e o produto final foi entregue.

Na Tabela 6 pode-se visualizar a lista de funcionalidades final do ciclo iterativo 5 e na Ilustração 15 o gráfico de andamento do quinto ciclo iterativo.

Tabela 6 – Lista de funcionalidades do ciclo iterativo 5

Título da tarefa	Dono da tarefa	Horas
Ajustes na movimentação	Carlos	10
Configurador de <i>Joystick</i>	Jeferson	10
Aplicação para configurações	Fernando	10
Animações de game	Tiago	50
Tela de resultados	Jeferson	20
Textos da tela de resultado	Jeferson	5
Pontuação	Jeferson	5
Reposição de bola	Fernando	10
Contagem regressiva no jogo	Ricardo	5
Rodada de testes do jogo	Todos	100
Instalador	Carlos	5

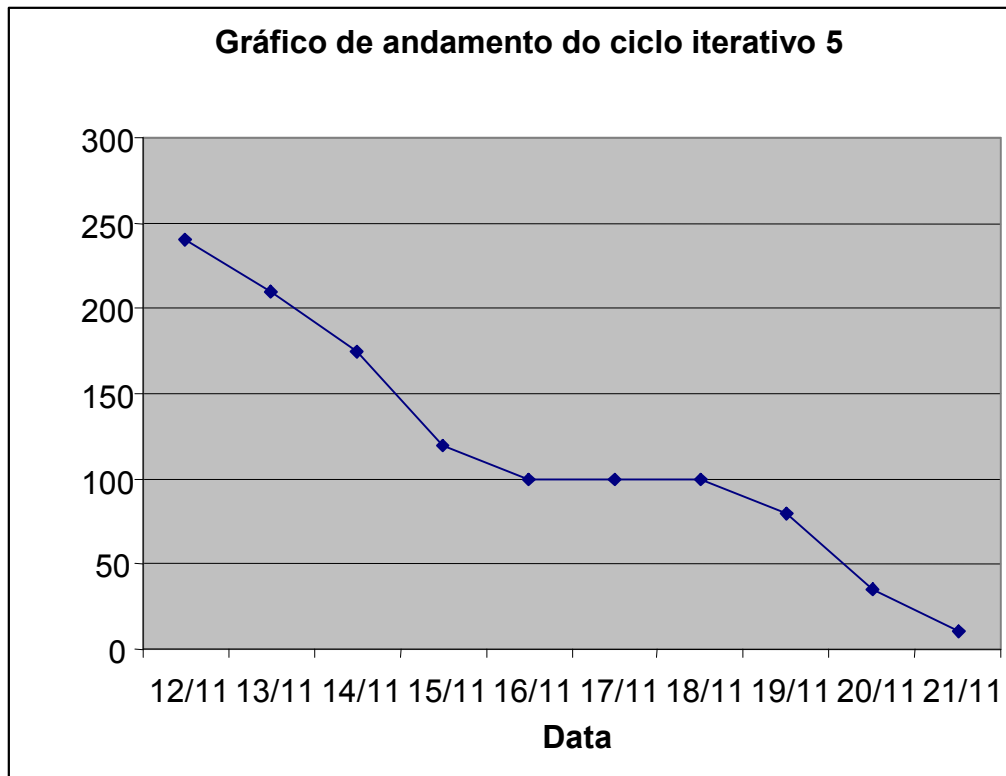


Ilustração 15 – Gráfico de andamento do ciclo iterativo 5

4.5 Análise dos gráficos finais do projeto

Ao final do projeto Arena Dalponte foi realizada uma análise em cima dos dados de gerência, possibilitando a geração de gráficos de desempenho e andamento do projeto.

4.5.1 Gráfico de andamento do projeto

No gráfico de andamento do projeto é possível visualizar uma curva linear de desenvolvimento com uma breve aceleração nos últimos dois ciclos iterativos. Com a

visualização da Ilustração 16, pode-se verificar que o projeto conseguiu manter uma média de tarefas/horas realizadas constante, mantendo a equipe produtiva durante todo o processo.

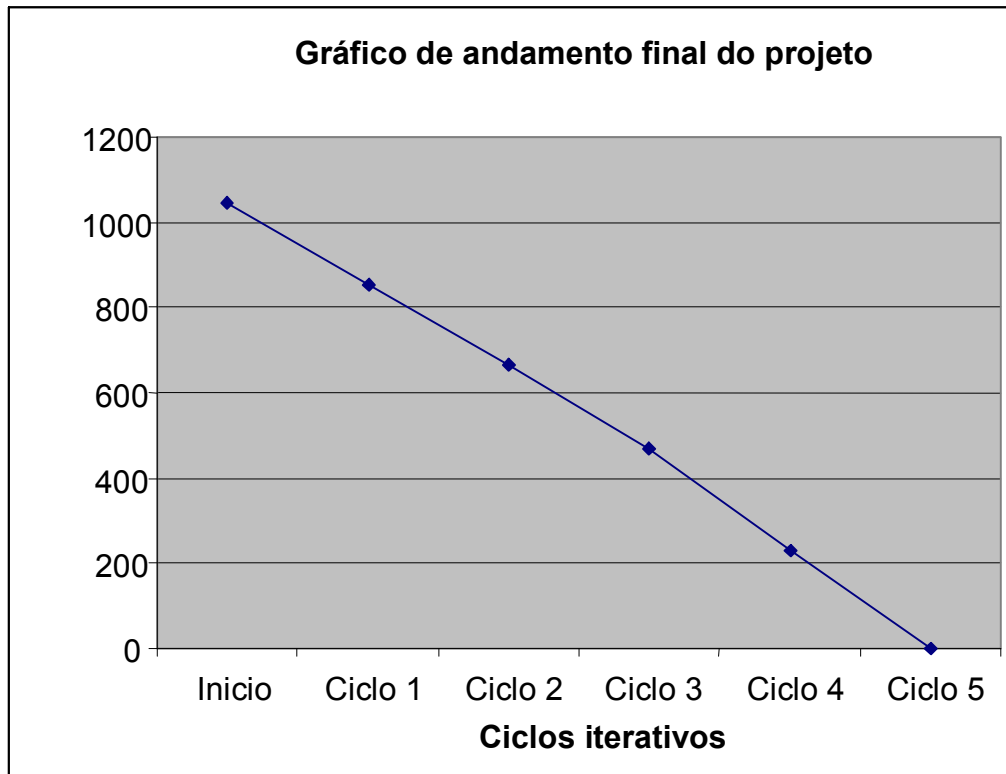


Ilustração 16 – Gráfico de andamento final do projeto

Também é possível visualizar na Ilustração 17 que a estimativa inicial para o tempo de projeto foi bastante próxima do tempo final de horas realizadas, e com um crescimento moderado na quantidade de horas de trabalho do projeto.

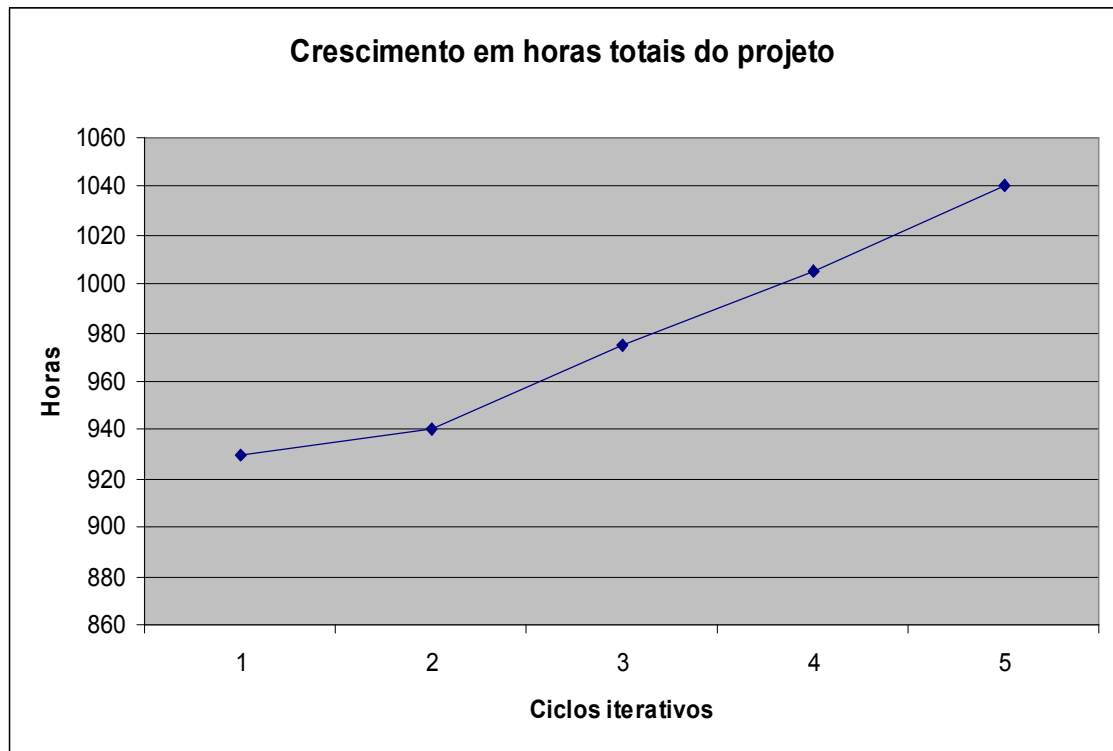


Ilustração 17 – Crescimento em horas totais do projeto

4.5.2 Tarefas adicionadas

No projeto Arena Dalponte houve uma concentração de inserção de tarefas na metade final do projeto, conforme a Ilustração 18. Isso é bastante comum em planejamento baseado em *Scrum*, pois o mesmo prevê que ao decorrer do projeto, novas funcionalidades e erros são descobertos. O projeto teve uma lista de funcionalidades do produto muito bem definido o que resultou em uma baixa porcentagem de inserção de novas tarefas, conforme a Ilustração 19. Do total de tarefas da lista de funcionalidades do produto somente 110 horas foram inseridas no decorrer do projeto.

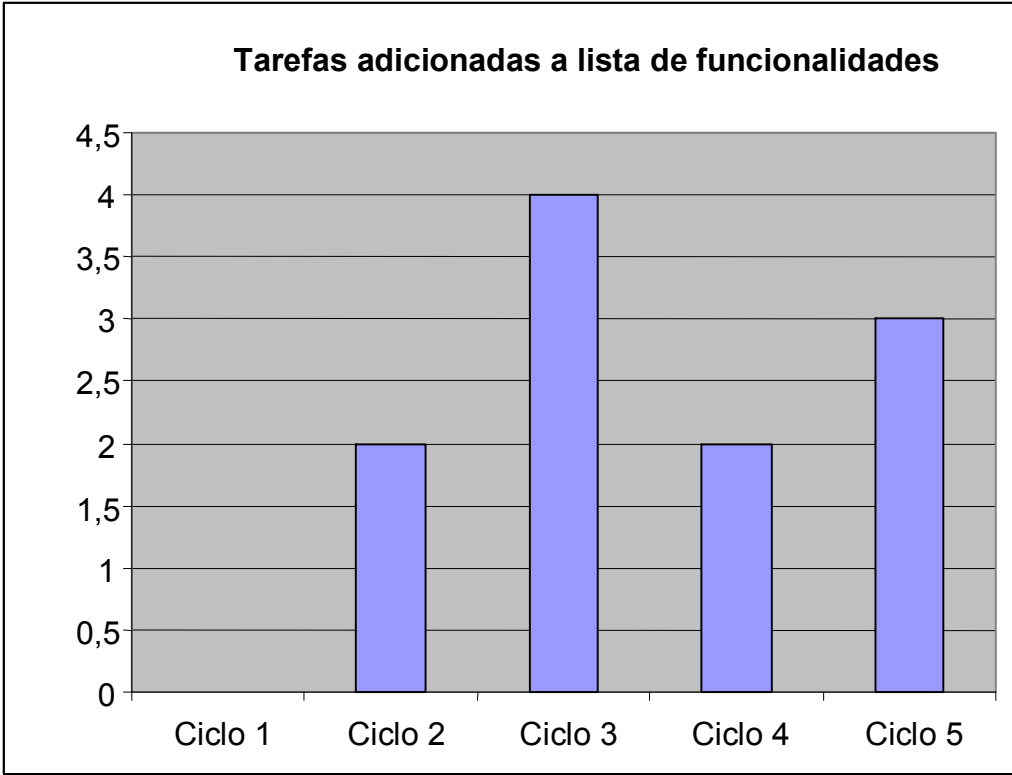


Ilustração 18 – Tarefas adicionadas a lista de funcionalidades

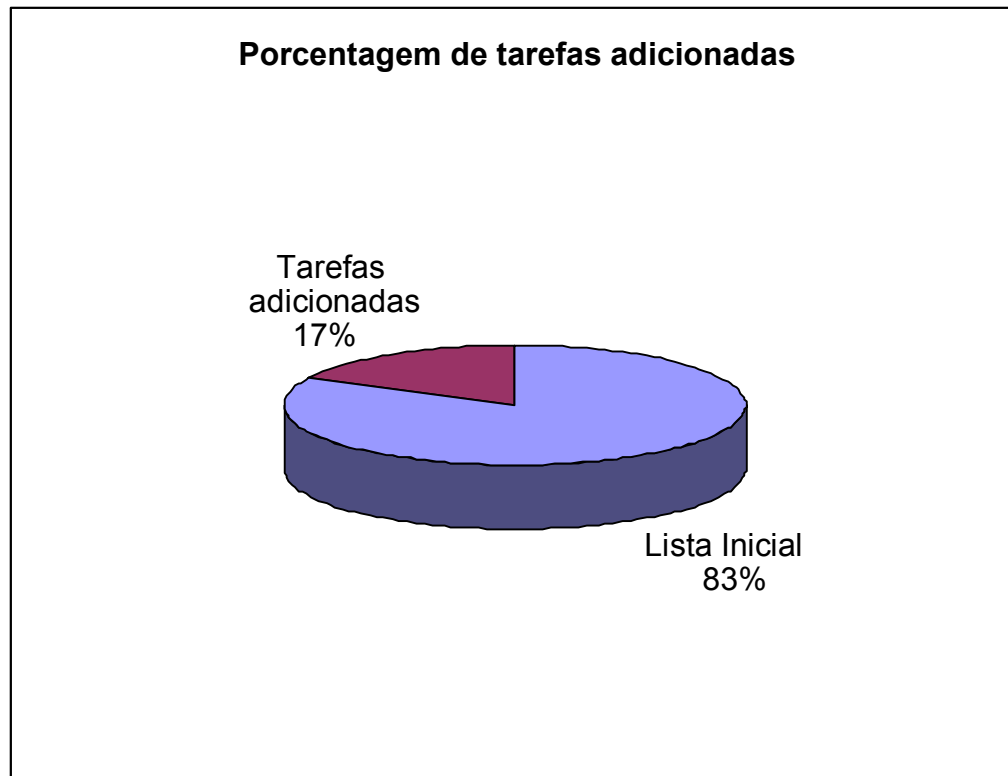


Ilustração 19 – Porcentagem de tarefas adicionadas

4.5.3 Rendimento em horas dos colaboradores

O tempo de trabalho dos membros da equipe por ciclo iterativo, conforme a Ilustração 20, se manteve equilibrado em distribuição de horas, com uma crescente notável na quantidade de horas trabalhadas nos últimos dois ciclos iterativos.

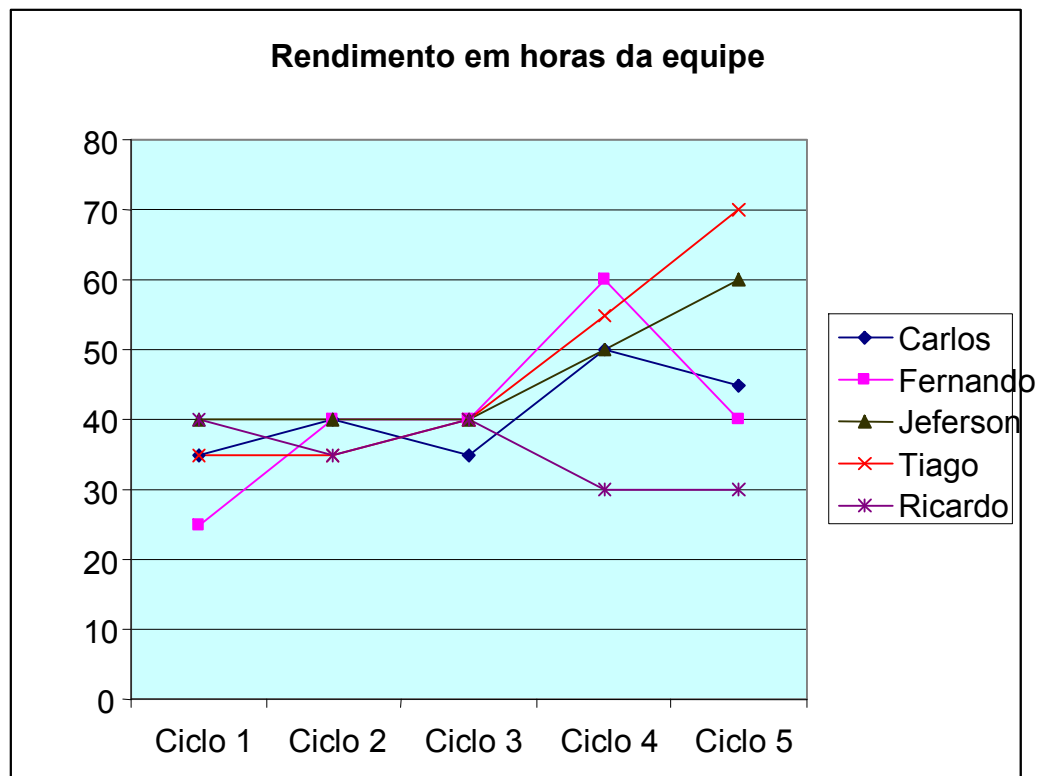


Ilustração 20 – Rendimento em horas da equipe

4.6 Ganhos com *Scrum* no projeto

O grande ganho com a gerência utilizando o método *Scrum* foi o conceito de quebra de um planejamento inicial completo no planejamento de pequenos ciclos iterativos. Dessa forma foi possível focar em pequenas fases de criação, conseguindo desenvolver concorrentemente sem conflitos tarefas das áreas de arte, programação, game design e sons.

Outro ganho perceptível com o uso do *Scrum* no processo de desenvolvimento do projeto Arena Dalponte, foi o envolvimento da equipe durante todo o processo. Por a equipe ter, a cada dia, informações sobre o que cada um estava desenvolvendo, manteve o foco no projeto e nas metas. Os objetivos das tarefas eram claros e as tarefas praticamente modulares, o que facilitava o bom rendimento da equipe.

Também durante o processo teve um retorno importante do andamento do projeto. Apesar de não ter todo o planejamento inicial completo, foi possível saber o que exatamente já foi produzido e quantas horas de trabalho ainda restavam para a finalização do projeto. Essa informação foi essencial para acelerar o desenvolvimento.

O envolvimento direto do cliente no processo criativo também foi fundamental, a empresa teve um retorno de imagem considerável, passando transparência, seriedade e comprometimento. Fator de extrema importância no diferencial que as empresas de desenvolvimento de software buscam agregar à sua imagem.

Por a Decadium Studios não ter adotado anteriormente nenhum tipo de processo de gerência, a aplicação de um método de gerência já foi um ganho de desenvolvimento, particularmente um salto de qualidade e gerência.

5 PRODUTO FINAL E DETALHES TÉCNICOS

Esta seção contém algumas ilustrações do projeto Arena Dalponte em sua versão final e os detalhes técnicos de desenvolvimento do projeto.

Detalhes técnicos:

1. Linguagem utilizada na programação da aplicação, JAVA;
2. Linguagem utilizada na programação do servidor, JAVA;
3. Biblioteca gráfica, OpenGL;
4. Biblioteca de jogo, *jMonkey Engine*;
5. Ambiente de desenvolvimento, *NetBeans*;
6. Aplicação de controle de tarefas, *dotProject*;
7. Ferramenta de modelagem gráfica, *Blender*.

As ilustrações 21 e 22 mostram a aplicação desenvolvida usando *Scrum* para gerenciamento do projeto.



Ilustração 21 – Tela de escolha de jogador



Ilustração 22 – Tela de jogo

6 CONCLUSÃO

Este trabalho apresentou um estudo de caso de um projeto de jogo 3D utilizando o método ágil de gerência *Scrum*. Esse estudo de caso foi realizado sobre o desenvolvimento de um jogo digital desenvolvido na empresa Decadium Studios com o nome de Arena Dalponte. Este trabalho também apresentou o método *Scrum* e os requisitos do projeto Arena Dalponte, e em seu corpo principal, relatou e apresentou os dados gerados durante o processo de gerenciamento *Scrum* no projeto Arena Dalponte.

O principal ganho desse projeto foi o de ter conseguido construir um produto em um curto espaço de tempo e de forma satisfatória utilizando um novo método de processo de software para uma empresa de software. O *Scrum* conseguiu manter a equipe empenhada e ativa durante todo o processo, conseguindo extrair o melhor de cada membro da equipe.

A aplicação do processo *Scrum* no projeto Dalponte foi um sucesso, porém não é possível generalizar afirmando que o método *Scrum* é ideal para qualquer projeto ou que o sucesso do projeto Arena Dalponte se deve ao método *Scrum*. O que é possível afirmar é que o *Scrum* auxiliou de diversas maneiras o desenvolvimento do projeto Arena Dalponte e que ocasionou um projeto tranquilo e linear.

Esse trabalho também contribui com o relato da aplicação real do método *Scrum* em um processo de desenvolvimento de *software* focado em jogos digitais. Também de forma secundária esse trabalho contribuiu com o processo de desenvolvimento de *software* da Decadium Studios, que agregou à sua empresa um processo de desenvolvimento ágil.

A Decadium Studios pretende continuar utilizando o método para projetos futuros. Também pretendo, de forma pessoal e acadêmica, utilizar dados deste trabalho para desenvolver estudos, procurando analisar e estender o método *Scrum* especificamente para o processo de desenvolvimento de jogos digitais.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABRAGAMES. **Plano Diretor da Promoção da Indústria de Desenvolvimento de Jogos Eletrônicos no Brasil.** São Paulo, 21 dez. 2004. Disponível em: <http://www.abragames.com/docs/pd_diretrizesbasicas.pdf>. Acesso em: 12 nov. 2007.

AGILE MANIFESTO. **Manifesto for Agile Software Development,** Utah, 13 fev. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 15 jan. 2008.

ASSOCIAÇÃO BRASILEIRA DE RUGBY. **Rugby no mundo.** Disponível em: <<http://www.brasilrugby.com.br/Rbrasil.htm>>. Acesso em: 13 jan. 2008.

BABYLON. **Babylon.** [S.l.], 1997. Versão 7.0.1 (r4). Sistema operacional Windows.

BARROS, Raphael Lima Belém. **Análise de Metodologias de Desenvolvimento de Software aplicadas ao Desenvolvimento de Jogos Eletrônicos,** no período de ago./97. 1997. 75f. Trabalho de graduação (Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2007.

JEVEAUX, Paulo César Machado. **SCRUM.** Disponível em: <<http://www.jeveaux.com/blog/2007/scrum/>>. Acesso em: 25 nov. 2007.

LLOPIS, Noel. GDC 2004: Software Engineering Roundtable Summary - Session 2. **Games from Within,** [S.l.], 30 mar. 2004. Disponível em: <<http://www.gamesfromwithin.com/articles/0403/000016.html>>. Acesso em: 11 jan. 2008.

MICROSOFT DEVELOPER NETWORK. **Process Templates.** Disponível em: <<http://msdn2.microsoft.com/en-us/teamsystem/aa718801.aspx>>. Acesso em: 19 nov. 2007

MIKE, Harland (Ed.). **Collins Gem Dicionário:** ingles/português, português/inglês. Co-editor Euzi Moraes. São Paulo, 1992. Edição Revisada.

OPENUP. **Introduction to OpenUP.** Disponível em: <<http://epf.eclipse.org/wikis/openup/index.htm>>. Acesso em: 19 nov. 2007.

PRESSMAN, Roger. **Software Engineering: A Practitioner's Approach, 6nd ed., preface.** In: The McGraw-Hill Companies. Columbus: McGraw-Hill Higher Education, 2005. Disponível em: <<http://highered.mcgraw-hill.com/sites/dl/free/0072853182/127056/pressmanPreface.pdf>>. Acesso em 08 jan. 2008.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004. 192 p.

TELES, Vinícius. **IMPROVE IT - SCRUM**. Disponível em: <<http://www.improveit.com.br/scrum>>. Acesso em: 20 nov. 2007.

WILSON, Kyle. GDC 2007. **Game Architect**, [S.l], 13 mar. 2007. Disponível em: <<http://gamearchitect.net/Articles/GDC2007.html>>. Acesso em: 11 jan. 2008.

8 OUTRAS REFERÊNCIAS BIBLIOGRÁFICAS

ARMOUR, Philip G.. **The Laws of Software Process – A New model for the production and management of software**. Florida: Auerbach Publications - CRC Press LLC - USA 2004. 300 p.

COHN, Mike; FORD, Doris. **Introducing an Agile Process to an Organization**. In: IEEE Computer Society, Digital Library, [S.l.], 2003. Disponível em: < <http://www.computer.org>>. Acesso em 20 set. 2007.

LIMPERT, Martin. et al. **Developing Complex Projects Using XP with Extensions**. In: IEEE Computer Society, Digital Library, [S.l.], 2003. Disponível em: < <http://www.computer.org>>. Acesso em 20 set. 2007.

SCRUM AT TOI. **Scrum bei T-Online International**. Disponível em: < <http://wiki.ipponsoft.de/ewiki/AgileWiki.php?page=ScrumAtTOI>>. Acesso em: 20 nov. 2007.