

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 10.07.2003

Assinatura:

*Ana Paula Jampais Ferreira*

## Evolução de uma agenda para grupos na *web*

*Tyciano Maia Ribeiro*

**Orientador:** *Profa. Dra. Renata Pontin de Mattos Fortes*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP – São Carlos  
Julho/2003

# Agradecimentos

Agradeço primeiramente a Deus pela saúde e disposição dadas para que este trabalho fosse desenvolvido.

Agradeço também a Prof. Dra. Renata Pontin de Mattos Fortes pela coordenação e extrema paciência no ensino e revisão de todo o conteúdo desta dissertação.

Agradeço a meus pais por sempre terem compreendido a minha ausência quando tinha que ficar semanas em São Carlos para trabalhar no meu projeto e também pelo apoio que sempre deram aos meus projetos pessoais.

Agradeço especialmente ao meu irmão Giovanni que, mesmo estando muito longe, me deu uma força enorme pra não me deixar desistir.

Agradeço finalmente a meus parentes e amigos pelos incentivos dados durante o tempo de mestrado e que foram de fundamental importância para o sucesso do mesmo.

## Índice

<b>RESUMO .....</b>	<b>5</b>
<b>INTRODUÇÃO .....</b>	<b>7</b>
1.1 – OBJETIVOS .....	8
1.2 – ORGANIZAÇÃO DA DISSERTAÇÃO .....	8
<b>2 - TRABALHO COOPERATIVO AUXILIADO POR COMPUTADOR .....</b>	<b>11</b>
2.1 – INTRODUÇÃO.....	11
2.2 - HISTÓRICO.....	13
2.3 – <i>GROUPWARE</i> .....	14
2.3.1 – <i>Definição</i> .....	15
2.3.2 – <i>Características</i> .....	15
2.3.3 – <i>Tipos de Groupware</i> .....	16
2.3.4 – <i>Classificação de Groupwares</i> .....	18
2.4 – FERRAMENTAS DE <i>GROUPWARE</i> .....	20
2.4.1 – <i>CoWeb-ICMC</i> .....	20
2.4.2 – <i>Agenda No Risk Planning</i> .....	22
2.4.2.1 – <i>Usuário: Administrador</i> .....	24
2.4.2.2 – <i>Usuário Convencional</i> .....	26
2.5 – CONTEXTO ( <i>AWARENESS</i> ) NA <i>AGENDA NO RISK PLANNING</i> .....	33
2.6 – CONSIDERAÇÕES FINAIS.....	37
<b>PROCESSO DE DESENVOLVIMENTO DA <i>NO RISK PLANNING</i> .....</b>	<b>39</b>
3.1 - <i>ENGENHARIA DE WEB</i> .....	39
3.1.1 – <i>Introdução</i> .....	39
3.1.2 – <i>Histórico</i> .....	40
3.1.3 – <i>Características de Aplicativos da Web</i> .....	41
3.1.4 – <i>Ciclo de Vida em Engenharia de Web</i> .....	45
3.1.4.1 – <i>Formulação</i> .....	46
3.1.4.2 – <i>Planejamento</i> .....	46
3.1.4.3 – <i>Análise</i> .....	47
3.1.4.4 – <i>Engenharia</i> .....	47
3.1.4.5 – <i>Geração e Testes de Página</i> .....	49
3.1.4.6 – <i>Avaliação do Cliente</i> .....	50
3.2 - <i>PROGRAMAÇÃO EXTREMA NA AGENDA NO RISK PLANNING</i> .....	50
3.2.1 – <i>Definição</i> .....	51
3.2.2 – <i>Histórico</i> .....	51
3.2.3 – <i>Características de XP</i> .....	52
3.2.4 – <i>Valores de XP</i> .....	53
3.2.5 – <i>Práticas de XP x Desenvolvimento da Agenda No Risk Planning</i> .....	54
3.3 – CONSIDERAÇÕES FINAIS.....	59
<b>4 - EVOLUÇÃO DA <i>AGENDA NO RISK PLANNING</i> .....</b>	<b>61</b>
4.1 – <i>AGENDA NO RISK PLANNING - VERSÃO 1.0</i> .....	61
4.2 – <i>AGENDA NO RISK PLANNING - VERSÃO 2.0</i> .....	63
4.2.1 – <i>Relatório Técnico</i> .....	63

4.2.2 – Alterações na Agenda.....	64
4.3 – AGENDA <i>NO RISK PLANNING</i> – VERSÃO 3.0.....	66
4.3.1 – Usuário: Administrador de Disciplinas.....	67
4.3.2 – Usuário: Disciplina.....	68
4.3.3 – Usuário: Guest.....	71
4.3.4 – Outras Alterações na Agenda .....	73
4.4 – IMPLEMENTAÇÕES FUTURAS – AGENDA <i>NO RISK PLANNING</i> – VERSÃO 4.0 .....	77
4.5 – CONSIDERAÇÕES FINAIS.....	78
<b>5 - CONCLUSÕES.....</b>	<b>81</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>83</b>

## Resumo

Este trabalho de mestrado está inserido no contexto de pesquisas sobre as características de desenvolvimento e aperfeiçoamento de uma aplicação para uso na Internet. A aplicação consiste de uma agenda eletrônica, chamada *No Risk Planning*, na forma de um *groupware*, que oferece um suporte ao trabalho colaborativo em grupo. Por ser uma aplicação desenvolvida com base na utilização da Internet, o objetivo foi o de observar quais as características de evolução neste tipo de software foram solicitadas e como um processo de desenvolvimento foi utilizado para atendê-las. Técnicas de Engenharia de *Web* (*web engineering*) foram adotadas, mas devido à rapidez com que as solicitações deveriam ser atendidas, técnicas de *Programação Extrema* (XP) foram acopladas para que o processo se desenvolvesse satisfatoriamente. Assim, a evolução da agenda para grupos na *Web* foi realizada como um estudo de caso, no qual a atualização de documentação, e o acompanhamento das etapas de evolução e dos requisitos foram essenciais para que se tivesse um produto com boa aceitação para uso no meio acadêmico.



## Abstract

*This dissertation is included in the context of researches on characteristics of development and improvement of an Internet-based application. The application consists of an electronic agenda (calendar), called No Risk Planning, that is a groupware which assists the collaborative work in groups. Since the application has been developed regarding the use of Internet, our objective was to observe which evolving characteristics have been requested and how a software process was assumed to implement them. Web engineering techniques were applied, and to attend the fast attendances of requirements, EXtreme Programming (XP) techniques were also used. Thus, the agenda's evolution was taken place as a case study, in which was possible to update the documentation, and the evolution steps and requirements could be traced. Finally, the product should have a good acceptance in the academic environment.*

# Capítulo 1

---

## Introdução

Existem situações do cotidiano em que devemos guardar certa informação tal qual ela nos foi transmitida para que possamos utilizá-la adequadamente no futuro. Assim, foram criados artifícios para armazenar estas informações, tais como: utilização de papel e caneta para fazer anotações importantes e lembretes de compromissos; e uso de agendas eletrônicas de mão como forma para registrar dados.

Essas informações isoladamente, no entanto, não oferecem recursos suficientes para se tirar máximo proveito do que foi registrado. É necessário inter-relacionar o conteúdo do que foi registrado para se obter um conjunto integrado de informações disponível para consultas futuras. Segundo Lee [LEE, 2001] há três tipos de agenda:

- Agenda de papel para uso pessoal;
- Agenda eletrônica somente para uso pessoal;
- Agenda eletrônica *online* para uso pessoal ou em grupo;

Todos esses três tipos de agenda possibilitam que o uso futuro das informações registradas seja aproveitado. Em especial, por terem sido alvo de pesquisa neste trabalho de mestrado, as agendas eletrônicas *online* para uso pessoal e em grupo foram estudadas com mais detalhes. Tais agendas são também denominadas Sistemas de Calendário em Grupo (em inglês, *Groupware Calendar Systems* (GCS's)) e consistem de sistemas que visam suprir a necessidade de integração de informações para uso pessoal ou para uso de um determinado grupo. Os GCS's são calendários que tipicamente podem ser compartilhados em uma rede como a Internet, por exemplo. Um dos benefícios de utilização de GCS's é o fato de que como diversos usuários acessam os GCS's, é possível que encontrem horários em comum para marcar reuniões ou qualquer outro tipo de compromisso entre eles, além de possibilitar que a divulgação de suas agendas individuais facilite se ter conhecimento de quais são suas atividades em andamento.

A agenda eletrônica *No Risk Planning* [MOURA 2001] é um exemplo de GCS e está disponível na Internet para ser utilizada por meio de *browsers*. Ela foi inicialmente desenvolvida para atender a comunidade acadêmica da universidade, mas foi rapidamente disseminada e



passou a ter novos requisitos. Este trabalho trata da evolução da agenda eletrônica *No Risk Planning*, incluindo nesta evolução o seu aperfeiçoamento e a inserção de novas funcionalidades. Em especial, o trabalho de manutenção realizado foi orientado a fazer com que a agenda pudesse oferecer novos tipos de serviços e atendesse, com isso, novos tipos de usuários.

### **1.1 – Objetivos**

O objetivo deste trabalho foi o estudo da evolução da agenda eletrônica *No Risk Planning* através do uso de Engenharia de *Web* e algumas práticas de Programação Extrema. Inserido no contexto de pesquisa sobre as características de desenvolvimento e aperfeiçoamento de uma aplicação para uso na Internet, foram observadas quais as características de evolução se apresentaram mais solicitadas e como o processo de desenvolvimento foi utilizado para atendê-las.

O processo de evolução da agenda contou com os conceitos de Engenharia de *Web*, e devido à rapidez com que as solicitações deveriam ser atendidas, técnicas de Programação Extrema (XP) foram acopladas para que o processo se desenvolvesse satisfatoriamente. Assim, a evolução da agenda *No Risk Planning* foi realizada como um estudo de caso, no qual a atualização de documentação, e o acompanhamento das etapas de evolução e dos requisitos foram essenciais para que se tivesse um produto cada vez mais aceito e utilizado no meio acadêmico, na forma de um *groupware* para uso por professores, funcionários e alunos de uma universidade.

### **1.2 – Organização da Dissertação**

Esta dissertação apresenta os estudos realizados sobre os fundamentos necessários à realização do trabalho de investigação da evolução da agenda para grupos na *Web*, e a seguir apresenta o processo de evolução propriamente. Assim, a dissertação está organizada conforme a seguir.

No Capítulo 2 – Trabalho Cooperativo Auxiliado por Computador (em inglês, *Computer Supported Cooperative Work – CSCW*) – há uma descrição desta tecnologia, que foi utilizada na implementação da agenda eletrônica. Seu histórico, uma descrição de contexto (*awareness*) e *groupware* e análise de algumas ferramentas que utilizam estas tecnologias. Além desta revisão da literatura, no Capítulo 3 – Processo de Desenvolvimento da Agenda *No Risk Planning* – é descrito como foi o processo de desenvolvimento da agenda através do uso de Engenharia de *Web* combinada com técnicas de *Programação Extrema*. Há uma descrição destes dois modelos

de processo, suas características, conforme discutidas na literatura e, como parte do trabalho realizado, é apresentado como foi a aplicação do processo estudado na evolução da agenda *No Risk Planning*. O Capítulo 4 – Evolução da Agenda *No Risk Planning* – descreve como ocorreu o desenvolvimento das versões da agenda *No Risk Planning*, quais foram os passos seguidos neste processo e em qual etapa se encontra atualmente. O Capítulo 5 – Conclusões – apresenta as conclusões deste trabalho, mostrando os resultados obtidos e possibilidades de pesquisas futuras.



# Capítulo 2

---

## Trabalho Cooperativo Auxiliado por Computador

Neste capítulo são descritos os estudos realizados sobre os fundamentos, as origens e algumas ferramentas que utilizam a tecnologia de Trabalho Cooperativo Auxiliado por Computador (CSCW - *Computer Supported Cooperative Work*). Com as noções desta tecnologia foi possível o entendimento dos recursos que uma agenda em grupos deve oferecer como suporte computacional adequado.

### 2.1 – Introdução

Embora muitas vezes as pessoas não percebam, o desenvolvimento de trabalhos em grupo é uma tarefa do cotidiano. Todo trabalho em que mais de uma pessoa participa, seja dando opiniões, tomando decisões ou apenas atendendo ordens, pode ser considerado como um trabalho em grupo. Entretanto, um trabalho em grupo só pode ser considerado um trabalho cooperativo se a relação entre os membros do grupo não for “vertical”, ou seja, quando somente alguns decidem e outros executam o que foi decidido [PORTO et al, 1994]. No trabalho cooperativo, a relação entre os membros do grupo deve ser “horizontal”, isto é, as decisões são tomadas e os problemas são resolvidos pelo grupo, depois de uma discussão feita entre os seus membros. Muitas vezes, um ambiente de trabalho cooperativo também considera a existência de uma hierarquia, mas ela deve ser flexível [PORTO et al, 1994].

Três formas de cooperação humana são descritas por Schmidt [SCHMIDT, 1999]:

- Cooperação *aumentativa*: caracteriza-se por agrupar pessoas que combinam suas habilidades pessoais para superar as capacidades físicas e intelectuais limitadas de um único indivíduo.
- Cooperação de *integração*: processo por meio do qual as pessoas focalizam suas habilidades específicas no esforço para alcançar um resultado que nenhum deles poderia alcançar isoladamente.

- Cooperação por *debate*: processo aplicado por um grupo com o objetivo de tomar decisões por ponderação das opiniões pessoais de cada membro ou por meio da definição de metas específicas em função de metas gerais.

Seja qual for dessas três formas de cooperação, pode-se observar que o trabalho cooperativo visa solucionar problemas por meio da divisão dos mesmos em partes que possam ser resolvidas pelo grupo, e um objetivo subjacente é diminuir o tempo e os custos envolvidos neste processo colaborativo.

A utilização de ferramentas computacionais por equipes de trabalho cooperativo é fundamental para que haja uma efetiva troca de idéias entre os componentes destas equipes, fazendo com que o trabalho seja mais produtivo, e ainda possibilitando a redução do tempo e dos custos envolvidos no processo. O computador, neste caso, se torna uma forma prática e indispensável de aumentar o desempenho dos indivíduos envolvidos nestas tarefas [SCHMIDT, 1999].

O trabalho cooperativo em grupo está sendo cada vez mais utilizado devido a muitos fatores, dentre os quais pode-se destacar: os problemas estão cada vez mais complexos de serem resolvidos, necessitando troca de idéias e de informações para solução dos mesmos; a dimensão destes problemas também tem crescido; tem sido exigida maior especialização dos profissionais envolvidos para solução dos problemas; e as soluções têm requerido diferentes competências.

Como consequência, uma mudança de atitude por parte dos profissionais pode ser notada. Atualmente, os profissionais têm que trabalhar como membros de equipes dentro das empresas, onde ocorre troca de informações e idéias com outros profissionais experientes em outras áreas, ou seja, onde ocorre o trabalho cooperativo em grupo [GOUVEIA, 2002].

O trabalho cooperativo apoiado por computador (CSCW) é uma área de pesquisa preocupada com o *design* de sistemas baseados em computador para o suporte ao trabalho cooperativo que visa melhorar o trabalho realizado por grupo de usuários, tendo estes, tarefas e objetivos em comum. Outra preocupação desta área é o entendimento dos efeitos causados pelo uso dos referidos sistemas [FLUCKIGER, 1995].

A área de CSCW considera também a variedade de formas de se trabalhar em conjunto e, para as quais dois conjuntos de usuários foram identificados [HAWRYSZKIEWYCZ, 1998]:

- Grupos de usuários com, no máximo, dez pessoas trabalhando no desenvolvimento de algum artefato, utilizando um aplicativo síncrono (todos utilizando o aplicativo no mesmo instante):

- Grandes grupos constituídos de dezenas de pessoas de diferentes comunidades com diferentes objetivos, cujo trabalho deve ser coordenado em direção às metas de uma empresa. Este trabalho é feito com um aplicativo de uso assíncrono (todos não precisam estar utilizando o sistema no mesmo instante);

Em muitos casos, os sistemas têm que suportar uso síncrono e assíncrono como é o caso da agenda *No Risk Planning* (detalhes no capítulo 3), em que muitas vezes não há a necessidade de todos estarem simultaneamente acessando-a, a não ser no caso de uso do bate-papo disponível.

Um dos grandes desafios do trabalho cooperativo apoiado por computador (CSCW) é habilitar pessoas a trabalharem efetivamente em equipes que são mediadas através dos computadores [MILLS, 1999]. Muitas vezes, este trabalho é dificultado pela falta de intimidade no uso de computadores por certas pessoas.

Assim, os desafios que a área de CSCW deve enfrentar no desenvolvimento dos seus sistemas estão relacionados com o fornecimento de mecanismos para: comunicação, coordenação e colaboração [CHANG, ZHANG & JIANG, 2001]. A agenda *No Risk Planning*, por exemplo, oferece comunicação por meio do calendário disponibilizado aos grupos, e de certa forma, a coordenação que é favorecida por meio da existência de usuários que criam os grupos e assim controlam sua composição e duração.

## **2.2 - Histórico**

Para a Ciência da Computação, o suporte ao trabalho cooperativo é recente. Em meados dos anos 70, a crescente preocupação em aumentar a produtividade das organizações, onde a maior parte do trabalho era feita em grupos, deu origem a uma área de pesquisa chamada Automação de Escritório (OA - "*Office Automation*") [NITZKE, CARNEIRO & GELLER, 1999].

Os primeiros esforços nessa área buscavam integrar e transformar aplicações monousuário como processadores de texto e planilhas eletrônicas, de forma a permitirem o acesso simultâneo de um grupo de usuários. A partir daí, atenção cada vez maior foi dada ao desenvolvimento de aplicações de uso cooperativo. A evolução das redes de computadores, oferecendo atraentes possibilidades de comunicação e interação entre as máquinas e conseqüentemente entre as pessoas, tem gradativamente redirecionado a visão do uso dos

computadores em geral [MOREIRA, 1998]. O uso de computadores para promover o trabalho colaborativo entre as pessoas tem aumentado significativamente nos últimos tempos.

Em 1986, a sigla CSCW foi publicamente lançada como título de uma conferência organizada pela *Association Computing Machinery* (ACM). A maioria dos trabalhos científicos na área de CSCW surgiu a partir desta primeira conferência. A terceira edição, em 1989, foi a primeira conferência européia sobre o tema. Outras conferências de periodicidade irregular, bem como diversas conferências que versam primordialmente sobre outros temas, têm dedicado um crescente espaço à área de CSCW [NITZKE, CARNEIRO & GELLER, 1999].

CSCW é uma área de pesquisa que continua evoluindo rapidamente. As características desta evolução ao longo do tempo, destacando suas diferenças, são apresentadas na Tabela 2.1.

**Tabela 2.1:** Mudanças de paradigma em pesquisas sobre CSCW [CHANG, ZHANG & JIANG, 2001]

<b>Característica</b>	<b>Passado</b>	<b>Presente</b>	<b>Perspectivas de Futuro</b>
Estilo de Aplicativo	Usuário único/ ambiente assíncrono	Múltiplos usuários/ ambiente síncrono	Baseado em comunidade/ubíquo
Contexto	Falta de contexto	Forçado e customizável	Baseado em demanda
Ambiente	Contexto único	Contexto múltiplo	Ubíquo
Comunidade	Isolada	Cooperativo	Evolucionária
Multimídia/Hipermídia	Textos e gráficos	Multimídia com áudio e vídeo	Realidade Virtual
Linguagem	Centrada no sistema	Centrada no Conhecimento	Linguagem Livre

Pode-se notar que a comunidade de usuários que é alvo das pesquisas em CSCW se mostra incredivelmente cooperativa. Além disso, a ubiqüidade do sistema (o fato dos usuários usarem sistemas CSCW sem que estes percebam a tecnologia que existe por trás destes sistemas) é a grande aposta de Chang, Zhang & Jiang [CHANG, ZHANG & JIANG, 2001] para a criação dos futuros ambientes CSCW.

### **2.3 – Groupware**

*Groupware* é a ferramenta, principalmente de software, que dá suporte ao trabalho em grupos de pessoas que estejam trabalhando com algum alvo comum. Com o auxílio de *groupware*, o trabalho em grupo possibilita a interação entre diferentes pessoas de diferentes organizações visando os seguintes resultados: eliminar as distâncias, aproximar as pessoas, reduzir o tempo gasto, aumentar a produtividade e diminuir custos.

### 2.3.1 – Definição

O termo *groupware* costuma ser usado quase como sinônimo de CSCW, porém alguns autores identificam uma tendência a diferenciar o emprego destes termos. Enquanto CSCW é usado para designar a pesquisa na área do trabalho em grupo e como os computadores podem apoiá-lo, *Groupware* tem sido usado para designar sistemas baseados em computador que dão o suporte a grupos de pessoas que trabalham na mesma tarefa (ou objetivo) e provê uma interface para um ambiente compartilhado. A noção de tarefa comum e ambiente compartilhado é de fundamental importância para a definição de *groupware* [ELLIS et al, 1991].

Assim, sistemas de correio eletrônico, de teleconferências, de suporte a decisão, de suporte a reuniões e editores de texto colaborativos são exemplos de *groupware*, na medida em que promovem a comunicação entre os membros de um grupo de trabalho, e que contribuem com isso para que o resultado seja maior que a soma das contribuições individuais de cada membro do grupo [MOREIRA, 1998].

As aplicações para o suporte ao trabalho cooperativo incluem mecanismos de comunicação que permitam às pessoas ver, ouvir e enviar mensagens umas às outras; mecanismos de compartilhamento da área de trabalho que permitam às pessoas trabalharem no mesmo espaço de trabalho ao mesmo tempo ou em momentos diferentes; e mecanismos de compartilhamento de informações que permitam o trabalho de várias pessoas sobre a mesma base de informações.

### 2.3.2 – Características

Segundo Passos & Bernardo [PASSOS & BERNARDO, 2001], as principais características de um *groupware* são as seguintes:

- *Comunicação*: que possibilita o envolvimento entre as pessoas que se relacionam entre si e pessoas que discutem e produzem novas idéias. A comunicação ajuda as pessoas a compartilhar suas informações pessoais;
- *Coordenação*: que envolve a equipe de projeto, o planejamento e agendamento de compromissos de grupos. A coordenação ajuda as pessoas a conhecerem os seus papéis dentro dos grupos e haver interação com os papéis dos demais membros;



- *Colaboração*: que envolve atividades eletrônicas entre pessoas que estão no mesmo ambiente e que estão em ambientes distintos; também possibilita envolvimento do trabalho de pessoas ao mesmo tempo e em tempos distintos. A colaboração ajuda e estimula as pessoas a trabalharem juntas sobre um mesmo objetivo;
- *Memória Corporativa*: é a chave do sucesso para uma economia baseada em conhecimento. A memória corporativa ocorre por ser importante se conhecer o que foi feito anteriormente para que se possa prosseguir adiante.

Existem diversos tipos de *groupwares* (Seção 2.3.3), e uma agenda para grupos, como é o caso de *No Risk Planning* (detalhes no capítulo 3), possui suas funcionalidades orientadas a suprir as características acima. Conforme cada uma dessas características é priorizada nas funcionalidades oferecidas pelos *groupwares*, surgem diversas razões para o crescente uso e atenção voltados a esse tipo de *software* (*groupware*). Mas as principais razões são provenientes das vantagens em relação aos sistemas de trabalho individuais [MOURA, 2001], pois quando algum assunto ou problema é discutido por um grupo de pessoas, a chance de se chegar a algo produtivo é maior do que se fosse trabalho envolvendo somente uma pessoa.

Em comparação com reuniões presenciais, o uso de um software como a agenda eletrônica para grupos, na tomada de decisões, traz as seguintes vantagens: melhor performance e resultados melhores na aprendizagem e discussão do assunto em questão. Ocorre também menor desperdício de idéias geradas nas discussões porque geralmente as pessoas refletem mais sobre os assuntos que irão discutir utilizando a agenda de grupo [BENBUNAN et al, 2002].

Como se pode notar, há várias razões para se utilizar sistemas de *groupware* seja em uma grande empresa, seja dentro de universidades. A agenda para grupos *No Risk Planning* foi desenvolvida como um *groupware* prioritariamente para auxiliar o trabalho em grupos de uma universidade. A seguir, são descritos outros tipos de *groupwares*.

### 2.3.3 – Tipos de *Groupware*

Brink [BRINK, 1998] identifica os seguintes exemplos típicos de aplicações de *groupware*, que em geral são bastante utilizados:

- ***E-mail***: O *e-mail* é a aplicação mais comum do *groupware*. Através dele, as pessoas podem trocar idéias e informações entre si e, com isso, visam adquirir mais conhecimentos sobre os assuntos abordados. Os sistemas básicos de *e-mail* incluem as

funcionalidades características de: repassar mensagens, arquivar mensagens, criar grupos de usuários e anexar arquivos a um *e-mail*.

- **Newsgroups e Listas de Usuários:** Os *newsgroups* e as listas são similares aos sistemas do *e-mail* exceto que utilizam mensagens entre grupos grandes de pessoas em vez de uma comunicação somente pessoa-a-pessoa.
- **Workflow:** Os sistemas *workflow* permitem que documentos sejam distribuídos através das organizações. Um exemplo simples de uma aplicação *workflow* é um relatório de despesas em uma organização: um empregado escreve um relatório de despesas e submete-o a seu chefe. Uma cópia de relatório é arquivada e então é enviada ao gerente para aprovação. O gerente recebe o original, aprova-o eletronicamente, a despesa é registrada no grupo e enviada ao departamento de contabilidade para o pagamento. Os sistemas do *workflow* podem fornecer características tais como a distribuição, o desenvolvimento dos formulários e a sustentação para diferentes papéis e privilégios dos seus usuários.
- **Edição Colaborativa:** Edição colaborativa, como o próprio nome diz, é a edição de documentos por mais de uma pessoa. Os sistemas colaborativos de escrita em geral suportam trabalhos que acontecem no mesmo momento e trabalhos que acontecem em momentos diferentes.
- **Lousas Compartilhadas:** As lousas compartilhadas permitem que duas ou mais pessoas visualizem e desenhem em uma mesma tela mesmo estando em locais diferentes. A maioria das lousas compartilhadas é projetada para a conversação informal, mas podem também servir a comunicações estruturadas ou a tarefas extraíndo informações mais sofisticadas, tais como o projeto gráfico colaborativo, publicações ou aplicações de engenharia.
- **Videoconferência:** A videoconferência é essencialmente um sistema de telefone com um componente visual adicional. O vídeo é vantajoso quando a informação visual está sendo discutida, mas não pode fornecer o benefício substancial na maioria de casos onde os telefones convencionais são utilizados. O vídeo suporta a conversação entre os membros de um *groupware* e oferece uma visão remota das atividades que estão sendo desenvolvidas pelo grupo.

- **Bate-Papo:** Os sistemas de bate-papo permitem a muitas pessoas escreverem mensagens em tempo real em um espaço público. Muitas salas permitem acesso controlado ou com moderadores para conduzir as discussões.
- **Sistemas de Suporte a Decisão:** Os sistemas de suporte a decisão são projetados para facilitar a tomada de decisões por grupos. Fornecem ferramentas para *brainstorming*, críticas a idéias, colocação de pesos e probabilidades sobre eventos e alternativas, e suporte a votação. Tais sistemas permitem decisões mais racionais porque fazem com que os membros do grupo expressem mais seu raciocínio, e portanto organizem seu pensamento, sobre o assunto em questão.
- **Jogos multi-usuário:** Os jogos multi-usuário estão tornando-se muito comuns na Internet. Os jogos são o exemplo das situações multi-usuário não cooperativo, embora mesmo os jogos de competição exijam cooperação dos jogadores para seguirem as regras do jogo. Os jogos podem ser enriquecidos por outros meios de comunicação, tais como o bate-papo ou os sistemas do vídeo.
- **Agendas em Grupo:** As agendas em grupo permitem marcar, gerenciar projetos e coordenar ações e compromissos entre muitas pessoas através da visualização do seu calendário de atividades pessoais. Uma característica importante é encontrar horários de conflito para evitar marcar reuniões para membros de grupos que tenham outro compromisso em determinado horário. Os calendários do grupo ajudam também a encontrar pessoas.

#### 2.3.4 – Classificação de *Groupwares*

Os sistemas de *groupware* podem ser classificados de acordo com a localidade de seus membros e com o tempo de uso. Na Tabela 2.2, são apresentados alguns exemplos de cada categoria [Usability First, 2002].

Observando os exemplos dados nas categorias, vale ressaltar que muitas vezes, no entanto, as aplicações de *groupware* atendem não exclusivamente às características para as quais foi classificada. Por exemplo, a agenda *No Risk Planning* se encaixa também na categoria de *groupware* que provê acesso ao mesmo tema (simultâneo), uma vez que uma de suas funcionalidades é possibilitar bate-papo entre os membros de um grupo de usuários (detalhes no capítulo 3).

**Tabela 2.2:** Tipos de *groupware* [Usability First, 2002]

	<b>Ao Mesmo Tempo "Simultâneo"</b>	<b>Em Tempos Diferentes "Não-simultâneo"</b>
<b>No Mesmo Local "Próximo"</b>	Jogos Multi-usuário Sistemas de Suporte a Decisão	<i>Workflow</i> <i>NewsGroups</i>
<b>Em Local Diferente "Distante"</b>	Lousas Compartilhadas Bate-papo	Agendas de Grupos Edição Colaborativa

Segundo Passos & Bernardo [PASSOS & BERNARDO, 2001], os sistemas de *Groupware* podem ainda ser classificados levando-se em conta a previsibilidade ou não de suas tarefas, como mostra a Tabela 2.3. Deve-se notar que existe ainda um consenso não totalmente estabelecido na área, uma vez que na segunda classificação os Sistemas de Suporte à Decisão são alocados na categoria de *groupware* que ocorre em momentos diferentes, o que equivaleria à categoria "não-simultâneo" da Tabela 2.2.

**Tabela 2.3:** Tipos de *groupware* levando-se em conta sua previsibilidade [PASSOS & BERNARDO, 2001]

	<b>Mesmo Tempo</b>	<b>Momentos diferentes, mas previsíveis</b>	<b>Momentos diferentes e imprevisíveis</b>
<b>Mesmo Local</b>	Jogos Multi-usuários	Videconferência	Sistemas de Apoio a Decisão
<b>Locais diferentes, mas previsíveis</b>	Lousas Compartilhadas	<i>E-mail</i>	<i>Newsletters</i>
<b>Locais diferentes e imprevisíveis</b>	Bate-papo	Edição Colaborativa	<i>Workflow</i> Agenda de Grupo

Nesta segunda classificação (Tabela 2.3), também há exemplos que se encaixam em mais de uma categoria. Jogos multi-usuário podem estar acontecendo no mesmo instante e no mesmo local, mas muitos *groupwares* deste tipo têm sido desenvolvidos para serem acessados em instantes e locais totalmente diferentes e imprevisíveis.

Ainda assim, essas categorizações indicam os principais requisitos de *groupware* que se espera para cada um dos tipos de aplicações. Vale ressaltar que, conforme os *groupwares* atendem a mais características, mais usuários deve conquistar, por atender aos anseios pelo trabalho em equipe. Na próxima seção são descritas duas ferramentas (CoTeia e *No Risk Planning*) que utilizam os conceitos de *groupware* e CSCW, que foram desenvolvidas no ICMC-USP.

## **2.4 – Ferramentas de *groupware***

Considerando-se a importância do tema *groupware* no contexto deste trabalho, foram estudadas as principais características de duas ferramentas com o objetivo de entender melhor como os principais conceitos se aplicam. A segunda ferramenta, a agenda *No Risk Planning* foi alvo de estudos também com relação ao processo de sua evolução.

### **2.4.1 – *CoWeb-ICMC***

A *CoWeb (Collaborative Website)* é uma ferramenta computacional hipermídia, desenvolvida no *Georgia Institute of Technology (GeorgiaTech)* em 1998, que permite a autoria colaborativa de página Web sem que o usuário necessite de conhecimento prévio de quaisquer tecnologias relacionadas, como comunicação cliente-servidor, linguagens de marcação e de scripts [GUZDIAL, 1999]. Fundamenta-se na tecnologia de *wiki*, que é um *website* colaborativo, onde o seu usuário pode editar e criar páginas.

A principal característica da *CoWeb* é a autoria aberta, uma vez que é dada a qualquer usuário a liberdade de criação, modificação e ligação entre as páginas Web existentes. A ferramenta *CoWeb* suporta a construção de um repositório informal de hiperdocumentos sem estrutura pré-definida, uma vez que os usuários podem criar ligações internas e externas de modo a construir e expandir o repositório.

Mesmo possuindo uma infra-estrutura simples, a *CoWeb* fornece ainda as seguintes funcionalidades aos seus usuários: sistema de busca por palavras-chave, histórico de modificações, controle de versão, sistema de submissão de arquivos externos ao ambiente (uploads) e ajuda on-line.

A partir de um levantamento bibliográfico realizado em torno de ambientes de suporte à edição colaborativa, observou-se que o potencial de utilização dessa ferramenta poderia ser expandido se algumas extensões à infra-estrutura atual fossem realizadas, como, por exemplo, controle de acesso e controle de concorrência [MACEDO, BULCÃO NETO & PIMENTEL 2001]. Visando minimizar algumas dessas limitações e incorporar novas funcionalidades, foi desenvolvida uma nova ferramenta tipo *CoWeb*, que vem sendo utilizada no ICMC-USP desde o início de 2001. Essa nova ferramenta foi denominada *CoTeia* [ARRUDA JR & PIMENTEL, 2002].

A *CoTeia* é uma ferramenta com implementação totalmente nova mas que mantém as características de simplicidade e de autoria colaborativa de documentos Web da *CoWeb* original.

Na CoTeia, um conjunto de páginas Web colaborativas é referenciado por uma *Swiki* (um *website* que permite edição aberta pelos usuários por meios de programas em *Squeak* – linguagem similar a *Smaltalk*, mas de distribuição livre).

A CoTeia tem sido muito utilizada no suporte às aulas do ICMC/USP, permitindo que o professor disponibilize *online* todo o material utilizado em sala de aula e os alunos possam interagir de forma direta com o professor e com os seus colegas através de postagem de novos materiais encontrados, listas de exercícios, dentre outros.

Sua interface de uso navegacional (Figura 2.1) possui um rodapé com páginas que possuem *links* para a página atual e possui cabeçalho com as funcionalidades básicas de edição.

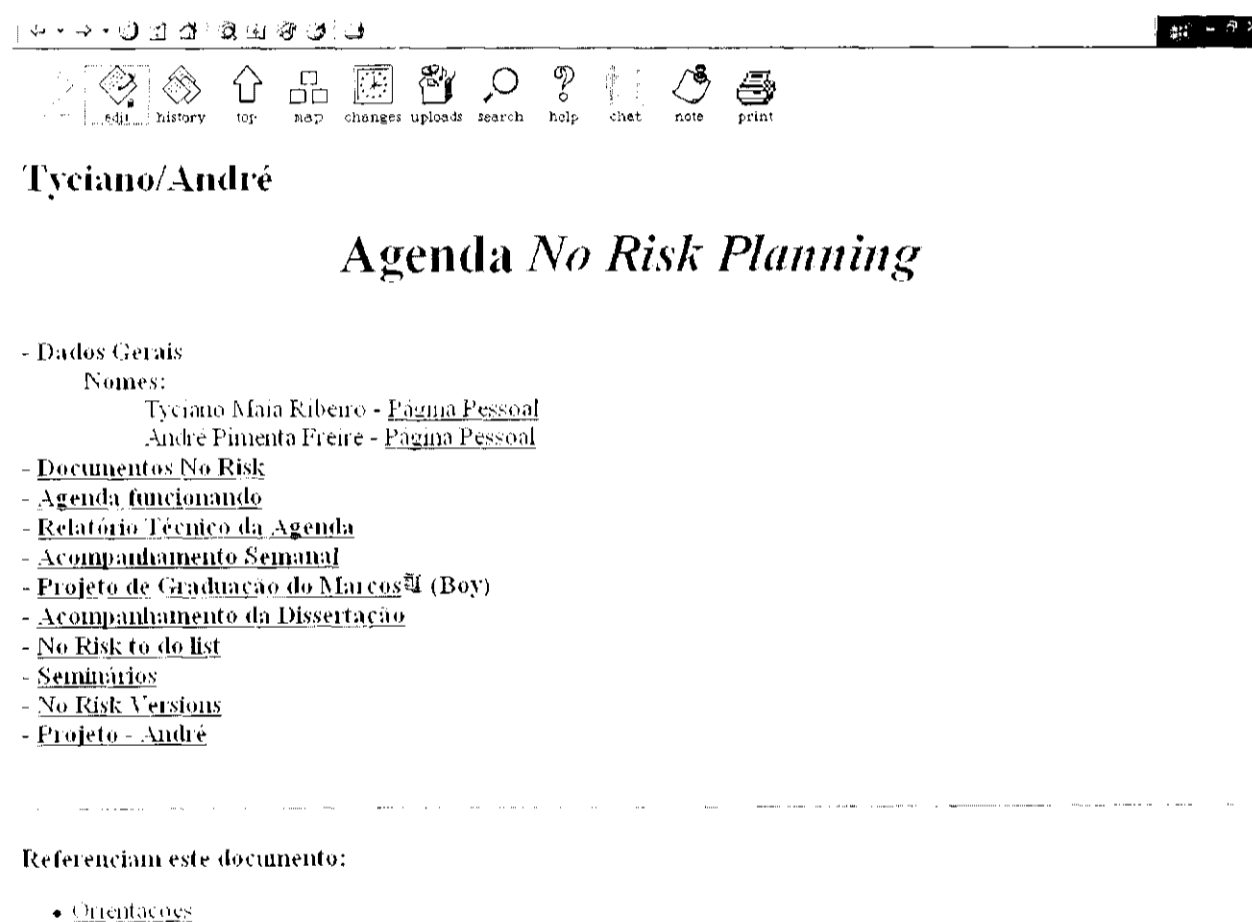


Figura 2.1: Página da *swiki* Tyciano/André na CoTeia

O mecanismo de criação de novos nós ou páginas da CoTeia (*swikis*), é totalmente transparente ao usuário e permite que estas páginas sejam criadas, armazenadas, interpretadas e finalmente visualizadas.



nesta seção é a versão inicial do projeto e os recursos e funções adicionados à *No Risk Planning* serão descritos no Capítulo 4.

Apesar de bastante simples, os serviços disponibilizados pela agenda são dos mais utilizados em trabalho cooperativo. A agenda serve também como um mecanismo de integração do grupo. Cada usuário deve ter a sua agenda pessoal onde faz anotações e marca compromissos. Uma funcionalidade bastante comum é a marcação de uma reunião com todo o grupo de trabalho. Neste caso, a agenda do grupo pode procurar por horários livres em cada agenda pessoal e definir o horário do encontro ou reunião [PORTO et al, 1994].

A agenda *No Risk Planning* consiste em uma agenda compartilhada por um grupo de pessoas de modo que os usuários possam utilizá-la de qualquer lugar que tenha acesso à Internet disponível. Seu projeto inicialmente se apresentava como um estudo dos principais conceitos sobre o desenvolvimento de um *Web Information System* (WIS) para suporte ao trabalho colaborativo em grupo [MOURA, 2001]. Toda a documentação de projeto do desenvolvimento da primeira versão da *No Risk Planning*, desde a idéia inicial, até a modelagem final e implementação da primeira etapa de desenvolvimento, se encontra em Moura [MOURA, 2001].

O objetivo inicial do projeto da agenda foi o de disponibilizar um conjunto de recursos funcionais para o sistema, possibilitando, por meio do seu real uso, o surgimento de outros requisitos que pudessem ser mais específicos às tarefas dos grupos de usuários ou mesmo visualizar outras formas de possíveis interações que fossem mais eficientes para agendamento de atividades em grupo.

A agenda *No Risk Planning* foi escrita em PHP versão 3.0 e utilizando *MySQL*, como Sistema Gerenciador de Base de Dados. A linguagem PHP foi escolhida por ser uma combinação de linguagem de programação e servidor de aplicações. Nela pode-se programar como em qualquer outra linguagem, definindo variáveis, criando funções, realizando *loops* e várias operações através de funções e procedimentos. O que realmente difere o PHP das outras linguagens de programação é a sua capacidade de interagir com a *Web*, transformando páginas estáticas em verdadeiras fontes de informações dinâmicas [MOURA, 2001]. O *MySQL* foi utilizado por ser um dos mais populares Sistemas Gerenciadores de Banco de Dados (SGBD) de código aberto que inclui suporte a *Structured Query Language* (SQL). No caso da agenda *No Risk Planning*, o *MySQL* foi utilizado para armazenar as tabelas de usuários, de grupos de usuários, de dias da semana, de compromissos, dentre outras.

Para que os dados dos usuários, que estejam acessando a agenda *No Risk Planning*, sejam guardados, foram utilizados *cookies* que são mecanismos para armazenar dados no navegador

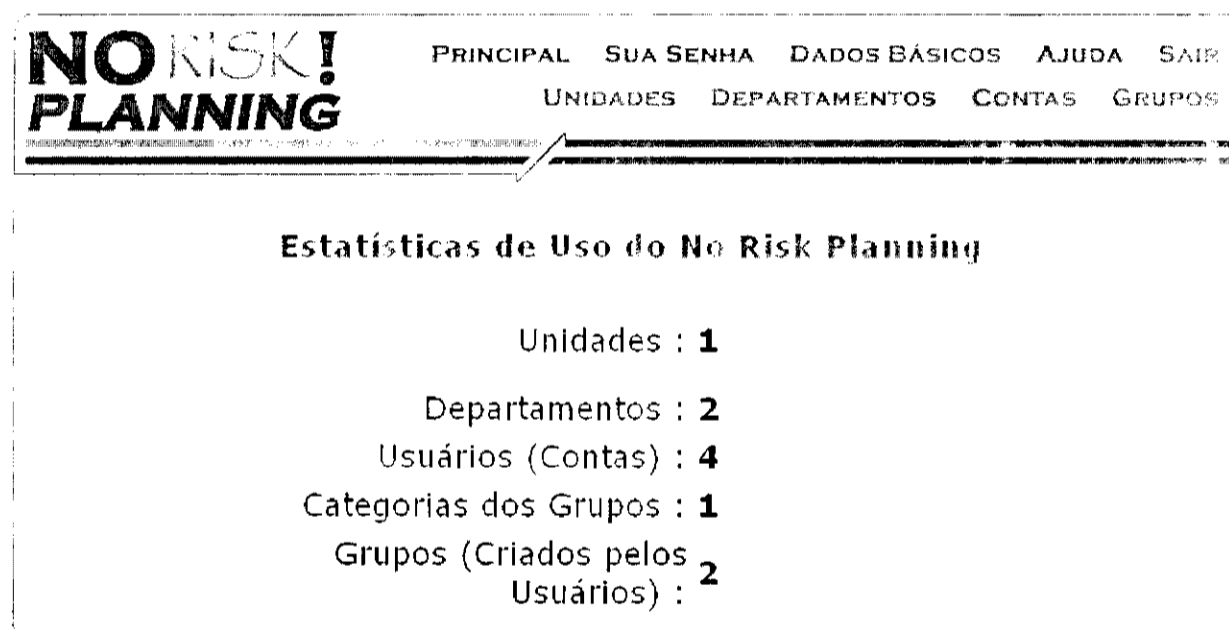


remoto de modo a garantir o acesso aos usuários. Estes dados são guardados em um arquivo-texto no computador do cliente que é acessado toda vez que o usuário entra no sistema.

Nas subseções a seguir são apresentadas as características funcionais da agenda *No Risk Planning*, do ponto de vista de seus dois usuários típicos: o administrador e o usuário convencional.

#### 2.4.2.1 – Usuário: Administrador

O administrador é responsável por definir a configuração do sistema para a instituição para a qual ele está sendo implantado, bem como criar as contas dos usuários e as categorias dos grupos. Ao preencher sua senha, um *cookie* é salvo no computador do administrador, para garantir seu acesso a toda a área administrativa, sem restrições.



**Figura 2.3:** Página de entrada do administrador

Na página de entrada do administrador (Figura 2.3), são apresentados dados estatísticos do sistema que indicam: o número de Unidades, de Departamentos, de Contas de Usuários, de Categorias de Grupos e de Grupos criados e gerenciados pela agenda. Na parte superior desta tela, o administrador possui um menu com as seguintes opções de configuração, além da **Principal** que se refere à apresentação desta mesma página (Figura 2.3):

- **Sua Senha:** para que o administrador modifique a sua senha pessoal.
- **Dados Básicos:** onde são preenchidas várias informações que são utilizadas frequentemente e representam a configuração do sistema, por exemplo: a página inicial

do sistema, o nome da instituição e o tamanho máximo permitido dos documentos do grupo.

- **Ajuda:** que exibe uma breve descrição do sistema com a finalidade de tirar possíveis dúvidas do administrador.
- **Unidades:** proporciona o gerenciamento das Unidades da agenda (Figura 2.4). Primeiramente tem-se um botão para criar nova unidade, que leva a um formulário para preencher as informações sobre a nova unidade. As devidas verificações são feitas para garantir que a unidade seja criada corretamente. Quando ocorre algum problema, são enviadas mensagens descrevendo tal problema ao administrador.

Nome da Unidade	Sigla	Breve Descrição
Instituto de Ciências Matemáticas e Computação	ICMC	-

Figura 2.4: Página para gerenciamento das unidades

A seguir, tem-se as opções de **modificar** ou **remover** uma unidade já existente. Para a opção de remoção e modificação, uma caixa de opções de unidades possibilita se escolher o nome da unidade e clicar no botão correspondente, um pedido de confirmação será solicitado antes da unidade ser removida ou modificada. Ainda durante o gerenciamento das unidades, tem-se a lista de todas as unidades existentes no sistema.

- **Departamentos:** gerenciamento dos departamentos, que segue o mesmo procedimento descrito no gerenciamento das unidades, com opções de criar, modificar ou remover um determinado departamento, além de exibir a lista de todos os departamentos existentes no sistema.
- **Contas:** gerenciamento das contas dos usuários, que também segue o mesmo procedimento descrito no gerenciamento das unidades, com opções de criar, modificar ou

remover uma determinada conta, além da lista de todas as contas existentes no sistema. A única diferença existente é quando se cria uma nova conta de usuário, pois um *e-mail* é enviado automaticamente para o mesmo, com seu identificador e sua senha pessoal, além de outras informações sobre o sistema para que ele possa começar a utilizá-lo.

- **Grupos:** as categorias dos grupos são usadas pelos usuários para definir seus grupos, bem como realizar buscas por usuários no sistema. A página de gerenciamento das categorias dos grupos também segue o mesmo procedimento descrito no gerenciamento das unidades, com opções de criar, modificar ou remover uma categoria, além da lista de todas as categorias existentes no sistema.
- **Sair:** ao clicar em Sair, o *cookie* que ficou salvo no computador é apagado, não permitindo que outros usuários acessem a conta, e assim o usuário se desconecta da agenda.

Com essas opções dispostas na parte superior da página, o administrador realiza facilmente o gerenciamento de todas as informações da agenda *No Risk Planning*.

o

#### 2.4.2.2 – Usuário Convencional

O usuário da *No Risk Planning* é responsável por administrar sua agenda pessoal, pode participar de grupos com outros usuários da agenda e pode ainda compartilhar sua agenda e documentos que julgar conveniente.

Na página inicial do sistema, o usuário preenche seu identificador (ID) e sua senha para poder acessar sua conta pessoal. Caso tenha perdido ou esquecido o seu ID ou senha, ele poderá recuperá-los preenchendo com seu *e-mail* no local indicado. Ao fazer isso, o usuário receberá por *e-mail* os dados relativos à sua conta. Ao acessar sua conta, um *cookie* é salvo em seu computador, para garantir seu acesso a todos os recursos de sua conta. A seguir, o usuário visualiza a página contendo o calendário semanal, conforme mostra a Figura 2.5.

A partir da página principal (Figura 2.5), o usuário visualiza sua agenda de horários da semana corrente (com intervalos de meia hora para marcar seus compromissos). Nesta agenda de horários semanal, encontram-se todos os seus compromissos, que podem ser classificados em Fixo, Importante e Normal, e cada um é identificado por uma cor diferente de acordo com seu tipo. Há também os compromissos do(s) grupo(s) do(s) qual(is) ele faz parte, que aparecem

com uma cor definida pelo criador do(s) grupo(s). Entre parênteses, aparece a sigla ou abreviação do nome do grupo, caso ela exista.

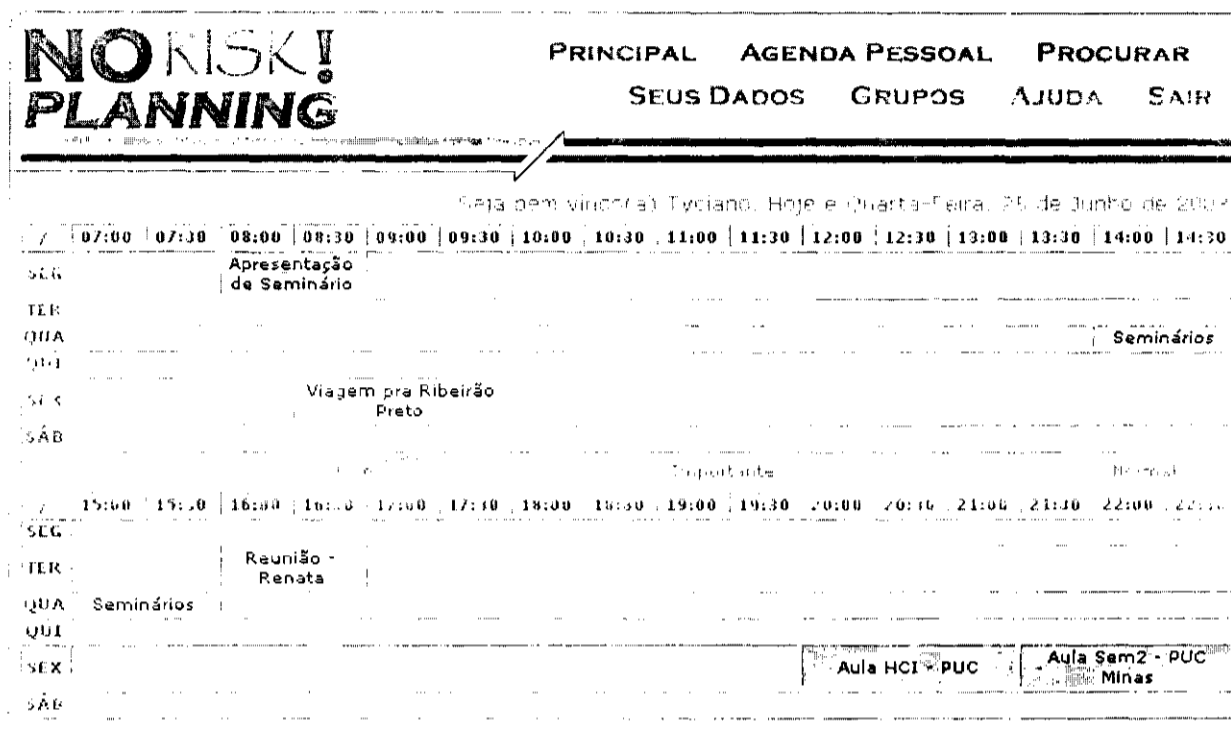


Figura 2.5: Página principal do usuário

Na parte superior da página principal (Figura 2.5), o usuário possui um conjunto de seis opções de operações disponíveis, conforme descritas a seguir.

#### “SEUS DADOS”

Nesta opção o usuário pode alterar seus dados básicos, por exemplo, seu nome, seu *e-mail* pessoal e sua senha. Esses dados, com exceção da senha, serão mostrados quando outros usuários realizarem buscas através da opção de menu Procurar, que será detalhada posteriormente.

#### “AGENDA PESSOAL”

Neste caso, o usuário possui três opções:

- **Inserir Novo Compromisso** - escolhendo esta opção, o usuário pode inserir novos compromissos em sua agenda pessoal. Todas as verificações são feitas antes do compromisso ser agendado. Se já houver compromissos marcados nos horários escolhidos pelo usuário, estes são verificados e removidos em todos os horários para os quais ele existe, a fim de manter-se a consistência da agenda. Também é verificado se há

compromissos de grupos marcados nestes horários. Compromissos de grupo não podem ser sobrepostos, pois são compromissos que fazem parte das agendas de todos os membros do grupo. O único que pode remover este compromisso é o criador do grupo e, fazendo isso, o compromisso é removido da agenda de todos os membros, para manter mais uma vez a consistência das agendas. Para adicionar um novo compromisso pessoal, o usuário escolhe o dia da semana, o horário de início e de fim do compromisso, além do tipo (Fixo, Importante ou Normal). Ao clicar em Alterar, além das verificações descritas anteriormente, também é feita a seguinte análise de consistência: o horário inicial tem que ser menor que o horário final do compromisso.

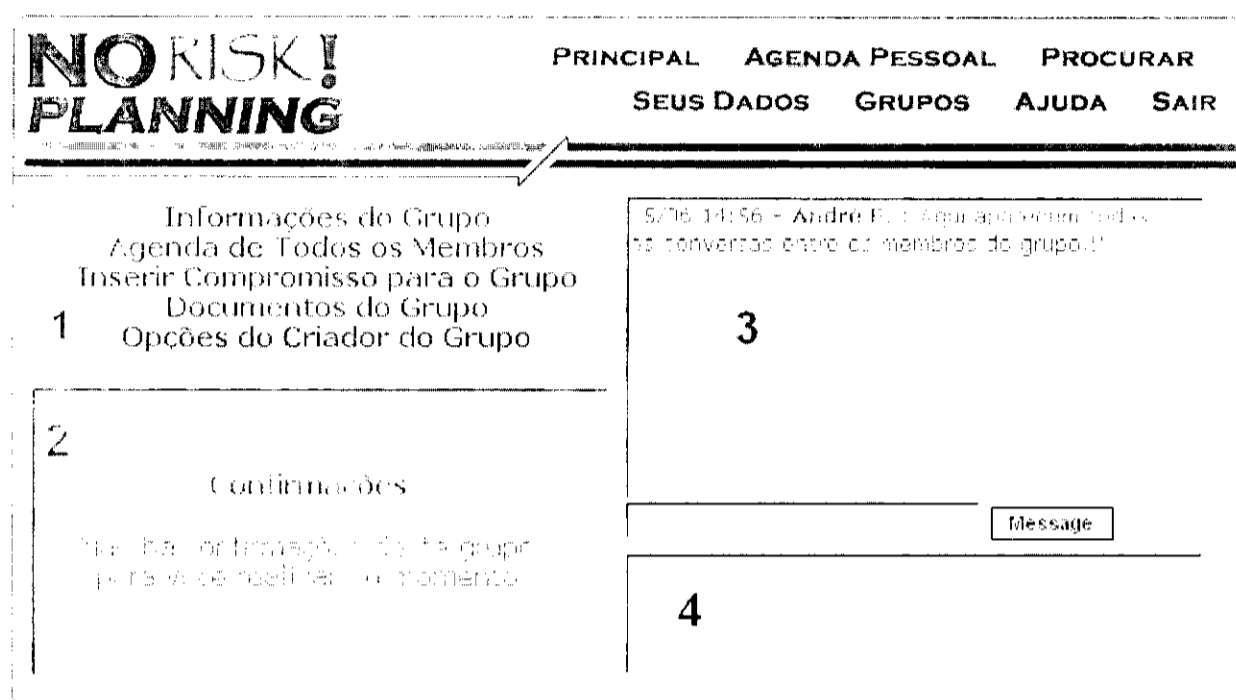
- **Remover Algum Compromisso** - esta opção exibe uma lista de todos os compromissos encontrados na agenda do usuário, e este deverá escolher qual destes compromissos deverá ser removido. Apenas os compromissos pessoais podem ser removidos por esta opção. Os compromissos de grupo só podem ser removidos pelo criador do grupo, como mencionado na opção anterior. Ao clicar no botão de remoção, um pedido de confirmação será solicitado para verificar se o usuário quer realmente remover o compromisso escolhido.
- **Limpar Toda a Agenda** - nesta opção o usuário poderá remover todos os compromissos da sua agenda de uma só vez. Novamente, os compromissos de grupo não podem ser removidos por esta opção. Um pedido de confirmação da remoção também será mostrado para evitar que o usuário limpe toda a sua agenda erroneamente.

#### “GRUPOS”

Esta opção é a que leva aos recursos funcionais mais importantes oferecidos pelo sistema, pois nela estão todas as opções para que os usuários trabalhem em grupo, seguindo alguns dos vários conceitos de *groupware*.

Primeiramente, o usuário encontra a opção de criar um novo grupo, bem como a lista de todos os grupos aos quais ele faz parte. Na opção de **Criar Novo Grupo**, o usuário escolhe os dados deste novo grupo, que são: nome, sigla, cor que os compromissos deste grupo serão representados nas agendas dos membros, categoria, número de membros e uma breve descrição. Ao clicar em **Continuar**, caso os dados tenham sido corretamente preenchidos, a próxima etapa mostra a lista de usuários no sistema, para que sejam escolhidos os membros que irão participar deste grupo. A seguir, aparecerá uma tela com vários campos que dependerá do número de usuários a serem inseridos no grupo. Escolhidos todos os membros, o grupo é criado, porém os

membros escolhidos não são adicionados automaticamente, pois estes podem não querer participar deste grupo. Então, o sistema envia um *e-mail* a cada um deles, pedindo autorização para serem adicionados. Ao aceitar o convite, os usuários passam automaticamente a fazer parte do grupo.



**Figura 2.6:** Página da área de trabalho em um grupo

Voltando à tela inicial da área de grupos encontra-se a lista de todos os grupos aos quais o usuário faz parte, e nela encontram-se as informações básicas destes grupos. Nesta lista também se encontra, ao lado esquerdo, o nome do grupo. Ao clicar no nome do grupo, o usuário tem acesso à área de trabalho deste grupo e obtém os recursos mostrados na Figura 2.6.

As funcionalidades da área de bate-papo da agenda *No Risk Planning*, de acordo com os números que aparecem na Figura 2.6, são as seguintes:

- **Área 1:** Ao clicar em uma destas opções, uma nova janela é aberta, e a tarefa escolhida é realizada. As opções para essas cinco tarefas são:
  - 1. Informações do Grupo:** são exibidas todas as informações do grupo, além dos membros e suas informações pessoais.
  - 2. Agenda de Todos os Membros:** semelhante à forma como são exibidos os horários das agendas dos usuários na seção Principal. Acessando esta opção, serão mostradas as agendas de todos os membros que fazem parte do grupo, possibilitando que todos

possam fazer comparações entre seus horários e os horários dos demais membros. As tabelas são divididas por dia da semana, com o nome do membro e o seu horário correspondente. A Figura 2.7 ilustra os compromissos na quinta-feira de um grupo que possui dois membros.

**3. Inserir Compromisso para o Grupo:** o sistema recupera os compromissos de todos os membros e compara seus horários, mostrando os horários ocupados (onde pelo menos um membro possui compromisso) e os horários livres (onde nenhum dos membros possui compromisso). O membro pode então verificar os horários livres e agendar um compromisso em um desses horários. Ao clicar em Agendar, é novamente feita uma verificação para garantir que o horário escolhido esteja livre para todos os membros e, caso positivo, o compromisso é agendado temporariamente para que todos os membros possam aprová-lo.

**4. Documentos do Grupo:** Nesta tarefa é disponibilizada a opção de **Adicionar Novo Documento**, para a qual o usuário escolhe em seu computador o documento a ser adicionado, bem como escreve uma descrição para o mesmo. O tamanho máximo permitido (em *kilobytes*) para os documentos é pré-definido pelo administrador. O usuário também visualiza todos os documentos do grupo, bem como seu tamanho, e pode fazer o *download* deles clicando em sua descrição.

**5. Opções do Criador do Grupo:** essas opções só aparecem para o membro que criou o grupo, conforme descritas a seguir.

- **Modificar Dados Básicos:** modifica as informações do grupo, que foram preenchidas quando o grupo foi criado.
- **Adicionar Novo Membro:** escolhe-se o usuário a ser adicionado na lista de usuários da agenda. Este recebe um *e-mail* pedindo autorização para ser adicionado. Caso aceite, ele passa a fazer parte do grupo, mas não adquire os compromissos previamente agendados para este grupo à sua agenda, uma vez que ele não participou da aprovação dos compromissos anteriores.
- **Remover Membro:** escolhe-se, entre os membros do grupo, qual será removido. Um pedido de confirmação é mostrado antes de removê-lo.
- **Remover Algum Compromisso do Grupo:** escolhe-se, entre os compromissos existentes para esse grupo, qual será removido. Um pedido de confirmação é mostrado antes de removê-lo. Após isto, o compromisso é removido da agenda de todos os membros.

- **Remover Todos os Compromissos do Grupo:** o criador do grupo pode remover todos os compromissos do grupo de uma só vez. Um pedido de confirmação é mostrado.
  - **Remover Documentos:** escolhe-se, entre os documentos do grupo, qual será removido. Um pedido de confirmação é mostrado antes de removê-lo.
  - **Remover Este Grupo:** um pedido de confirmação é mostrado e, após isto, o grupo e os dados que estiverem relacionados a ele são removidos, por exemplo, os compromissos e os documentos.
- **Área 2:** Quando um dos membros do grupo realiza agendamento de algum compromisso, os demais membros recebem nesse quadro um aviso para confirmar ou não o agendamento. Basta que apenas um dos membros rejeite o agendamento para que este não seja aceito. Caso todos aceitem, o compromisso é agendado. O prazo para que todos aprove o agendamento é de 24 horas. Passado este prazo, o compromisso é automaticamente rejeitado.
  - **Área 3:** Esta região consiste de um bate-papo, com as seguintes diferenças em relação aos bate-papos convencionais: a última mensagem enviada é sempre mostrada na parte superior do quadro, as mensagens ficam sempre armazenadas no banco de dados e o usuário pode sair e voltar na área de grupo, que ele sempre verá as últimas 20 mensagens.
  - **Área 4:** Neste quadro são mostrados avisos a respeito do agendamento de compromissos, se estes foram aceitos ou rejeitados, com a data e o nome do membro (caso tenha sido rejeitado).

Continuando a descrição das opções aos usuários a partir da página principal, temo-se:

#### “PROCURAR”

Diversas opções de buscas podem ser realizadas seguindo a estrutura da instituição, como mostra a Figura 2.7. Feita a escolha, os resultados, ordenados pelo nome do usuário, os dados básicos, como unidade e departamento a que pertence, *e-mail* e comentários são mostrados.



**NO RISK!  
PLANNING**

PRINCIPAL AGENDA PESSOAL PROCURAR  
SEUS DADOS GRUPOS AJUDA SAIR

Selecione no lado esquerdo qual das opções de pesquisa você quer realizar:

Primeiro Nome do Usuário  Sobrenome

Nome de um Grupo  Sigla de um Grupo

Unidade da Instituição

Departamento da Instituição

Categoria dos Grupos

Mostrar por página :  5  10  20  50 resultados

**Figura 2.7:** Opções de buscas de elementos da agenda *No Risk Planning*

#### “AJUDA”

Descreve o modo de funcionamento do sistema, para que o usuário sinta-se familiarizado mais rapidamente com ele. Cada função de menu é descrita para que o usuário saiba exatamente o que estará fazendo caso acesso determinada parte da agenda.

#### “SAIR”

Ao clicar em Sair, o *cookie* que ficou salvo no computador é removido, não permitindo que outros usuários acessem a conta e o usuário sai da agenda.

Todas essas funcionalidades foram oferecidas ao administrador e aos usuários convencionais pela versão 1.0 da agenda eletrônica *No Risk Planning*. O trabalho apresentado nesta dissertação investigou formas de se incrementar a agenda de modo a proporcionar outras funcionalidades e, com isso, aumentar a variedade de usuários atendidos. Ainda com relação a esse incremento de funcionalidades e atendimento a outros usuários, observou-se que a noção de contexto (*awareness*) na agenda deveria ser priorizada, pois as solicitações de seus novos

usuários cresciam a cada dia. Na próxima seção, é apresentado o estudo das informações de contexto que são proporcionadas na agenda *No Risk Planning*.

## **2.5 – Contexto (*Awareness*) na Agenda *No Risk Planning***

A falta de informações de contexto dentro de um grupo pode causar uma série de problemas que afetam a eficiência e a qualidade do trabalho em grupo. Quando os membros não têm conhecimento sobre o que está sendo desenvolvido pelos seus colegas, o trabalho resultante pode ficar sem coesão ou com partes repetidas [PINHEIRO, 2001].

Para tentar solucionar estes problemas, surgiu o conceito de *awareness* (contexto). *Awareness* é o conhecimento sobre o grupo e suas atividades passadas e presentes e constitui ponto vital para o trabalho colaborativo, sem o qual este trabalho fica descoordenado e perde em qualidade e eficiência [PINHEIRO, LIMA & BORGES, 2001]. O contexto também diz respeito à percepção que o usuário tem dos diversos estados em que se encontram os trabalhos que ele vem desenvolvendo no grupo comparado a seus colegas [SANTOS et al, 1998].

O objetivo do conceito de contexto é possibilitar que os membros de um grupo tenham a percepção de onde o seu trabalho pessoal se encaixa no trabalho do grupo como um todo. Os colaboradores devem estar conscientes dos trabalhos que os outros membros do grupo estão desenvolvendo em determinado momento para que eles possam decidir sobre suas próprias ações [CHANG, ZHANG & JIANG, 2001].

No processo de tomada de decisão em grupo, o sistema tem que exibir para os seus usuários, o estado, o comportamento e percepções em um ambiente que viabiliza este processo. O usuário percebe as mudanças de comportamento e participação de cada colega ao longo do processo de tomada de decisão [BANNON & BODKER, 1997]. Na agenda *No Risk Planning*, a principal fonte de contexto é obtida na medida em que os usuários encontram as informações inseridas por seus colegas de grupo no espaço compartilhado.

Em geral, existem mecanismos que indicam ao usuário as atividades dos outros membros do grupo. Tais mecanismos possuem características relacionadas a seis questões importantes (*what, when, where, how, who e how much*) que são descritas a seguir [PINHEIRO, LIMA & BORGES, 2001].

**1. O que (*What*):** refere-se a informações que devem ser oferecidas aos usuários para que eles conheçam qual é o trabalho do grupo como um todo. Estas informações dizem respeito a dois

aspectos principais: as **atividades** relacionadas ao próprio grupo, que são a base do trabalho cooperativo, e os **papéis** de cada membro do grupo, representando a noção de hierarquia dentro do grupo.

Com relação às atividades, devem ser considerados os dois tipos de ambientes de trabalho: no ambiente *síncrono* é mais interessante saber em detalhes quais atividades estão sendo realizadas em determinado momento e no ambiente *assíncrono*, como não há garantias quanto a em que momento determinada tarefa será realizada por um colega, é mais interessante saber quem está responsável por quais atividades e que trecho do trabalho está sob os cuidados de quem. Na agenda *No Risk Planning*, por se tratar de um espaço compartilhado por grupos de pessoas, tem-se características de ambiente *assíncrono* prioritariamente, mas também provê ambiente *síncrono* (bate-papo), conforme descrito na Seção 2.4.2.2, assim os usuários têm tanto a noção da divisão de tarefas dentro do grupo para poder situar seu trabalho individual, por meio de comunicação com os grupos aos quais pertence, quanto têm noção de quais são as atividades dos demais participantes do grupo, pois pode visualizar suas agendas individuais.

Com relação a noção de papéis de cada usuário, a agenda *No Risk Planning* possibilita que os criadores de grupos tenham controle sobre os mesmos; a hierarquia não é registrada.

**2. Quando (*When*):** refere-se a informações de quando ocorrem os **eventos** geradores de contexto e quando se dá a **apresentação** destas informações.

Pode-se dividir a ocorrência de eventos em quatro tempos distintos que são: “passado” para as ações que aconteceram em um intervalo de tempo anterior; “passado contínuo” para eventos que começaram no passado e continuam ativos no presente; “presente” para eventos que estão acontecendo neste exato momento e “futuro” para ações de grupo que acontecerão posteriormente.

Estes tempos são mais ou menos importantes, dependendo do ambiente do sistema. Em ambientes *síncronos*, o foco de visão é no que está acontecendo no presente. Sem esta noção, seus membros podem ficar perdidos e sem saber o que está acontecendo no grupo. Em ambientes *assíncronos*, o foco de visão é no que aconteceu no passado e no passado contínuo, fazendo com que haja a necessidade de se ter uma alta persistência das informações geradas. Nos dois tipos de ambientes, uma visão de futuro serve para manter os membros atentos aos possíveis rumos do trabalho desenvolvido a partir do presente momento.

No caso da agenda *No Risk Planning*, a visão de ocorrências de eventos nos tempos descritos é essencial para que haja bom uso de seus recursos. A fonte geradora desta informação

de contexto (quando) é tipicamente o seu usuário. Quanto mais real for a alimentação desses dados, mais os membros de cada grupo sabem o que ocorreu no passado e passado contínuo para poderem, a partir daí, planejarem o futuro, e também agendarem em suas agendas.

Outra característica importante é quando fazer a apresentação das informações. Em sistemas *síncronos*, como o interesse está nos eventos que estão ocorrendo naquele instante, quanto antes for dada a percepção destes eventos ao grupo, melhor será o desempenho do grupo. No caso do bate-papo provido na agenda *No Risk Planning*, esta percepção é efetiva. Já em sistemas *assíncronos*, há um intervalo de tempo entre a ocorrência do evento e a sua percepção pelos colegas do grupo. Assim, nestes sistemas a informação da percepção é apresentada aos usuários possivelmente em um momento posterior à ocorrência dos eventos, como ocorre na agenda *No Risk Planning* também.

**3. Onde (Where):** refere-se ao lugar onde as informações são geradas e apresentadas a outros membros.

A percepção do **espaço compartilhado** (*workspace awareness*) pelo grupo é muito importante em ambientes *síncronos* para que seus membros tenham a noção de todas as atividades realizadas neste espaço de trabalho compartilhado e, com isso, possam estruturar sua própria participação no grupo. Em ambientes *assíncronos*, são observados os **objetos compartilhados** pelo grupo, já que não há como se certificar da presença dos membros num intervalo de tempo. É através da manipulação destes objetos compartilhados e do histórico desta manipulação, que a comunicação entre os membros ocorre efetivamente.

O compartilhamento de objetos na agenda *No Risk Planning* ocorre através das mensagens de grupos, através dos compromissos de grupos, através dos arquivos armazenados por seus membros e principalmente através de troca de *e-mails* entre seus membros.

A metáfora utilizada pelo tipo de encontro de grupo afeta o modo como as informações podem ser percebidas pelos participantes, havendo a necessidade de enriquecer adequadamente esta metáfora com as informações de contexto. Em reuniões *síncronas*, usa-se a metáfora de um escritório e em reuniões *assíncronas* usa-se a metáfora de uma agenda, já que os membros não necessariamente estariam reunidos de forma *síncrona*.

Em resumo, a pergunta “onde” refere-se ao local que as informações são geradas e apresentadas a todos os membros do grupo, relacionando-se ao espaço de trabalho e à metáfora utilizada neste espaço dependendo do tipo de grupo.

**4. Como (How):** refere-se a como as informações são apresentadas aos membros de grupo e como a sua interface reage a favor da percepção de usuário.

A interface da aplicação deve transmitir as informações de forma reduzida, evitando sobrecargas e perdas de conteúdo. Para tanto, deve-se utilizar elementos de interface adequados, que apresentem as informações de maneira balanceada, sem que haja excesso e nem falta de informações. Em geral, a estratégia é usar filtragem e agrupamento de informações.

Em sistemas *síncronos*, é importante ter a oportunidade de ver onde o seu colega está trabalhando agora e o que está fazendo em detalhes. Para isso, são usadas as seguintes interfaces: *What You See Is What I See (WYSIWIS)* onde todos os membros têm a mesma imagem de espaço compartilhado e de seus objetos, garantindo o contexto das atividades; *WYSIWIS* “relaxada” que permite maior liberdade de navegação aos usuários e, finalmente, múltiplas visões em que os usuários podem optar em visualizar o espaço compartilhado e seus objetos de várias maneiras.

Em sistemas *assíncronos*, como não há certeza dos membros estarem trabalhando ao mesmo tempo, as interfaces *WYSIWIS* puras não se aplicam. Em geral, utiliza-se: *WYSIWIS* “relaxado”, múltiplas visões e as interfaces desacopladas que são interfaces independentes umas das outras. No caso da agenda *No Risk Planning*, a interface utilizada é a *WYSIWIS* “relaxada” porque seus membros de grupo têm a visão que os outros do grupo disponibilizaram para ele.

**5. Quem (Who):** refere-se a quem está trabalhando e quem está acessando o grupo em determinado momento.

Em ambientes *síncronos* basta “olhar ao redor” para se ter a noção de quem está ou não atento ao que está sendo discutido naquele momento. Neste caso, a percepção de outros membros é necessária para que haja um trabalho em grupo efetivo; esta percepção é obtida das interações e comunicação entre os membros. A autoria das atividades também contribui para a noção do contexto “quem”, e geralmente é identificada em um ambiente síncrono.

Em ambientes *assíncronos* não há a necessidade de se ter uma noção de presença de outros membros. Neste caso, a presença de membros simultâneos é somente uma oportunidade a mais de estar desenvolvendo o trabalho; a presença de um membro no sistema não garante que ele esteja realmente atento. A agenda *No Risk Planning*, como prioritariamente um sistema assíncrono, não exige que todos os membros do grupo estejam *online* em determinado momento para que o trabalho possa ser desenvolvido. A evolução do trabalho se dá conforme são percebidos os objetos compartilhados pelos membros do grupo.

Quando a noção de “quem” envolve pessoas, tem-se que considerar também as formas de comunicação entre estas pessoas. Em sistemas *síncronos*, ocorre comunicação síncrona que inclui ferramentas do tipo conferência e bate-papo e em sistemas *assíncronos*, ocorre o uso de ferramentas assíncronas, como *e-mails*, quadros de aviso e notas (como acontece na agenda *No Risk Planning*).

**6. Quanto (*How much*):** refere-se à quantidade ideal de informações que deve ser apresentada ao usuário para prover uma percepção real sobre o grupo e suas atividades.

Esta questão afeta todas as outras cinco descritas anteriormente (tornando-se uma dimensão paralela), porque a insuficiência de informações pode fazer com que os usuários fiquem sem contexto para as suas atividades e, a sobrecarga faz com que os usuários interrompam seu trabalho para tentar filtrar informações que atendam os seus interesses.

Desta forma, o principal objetivo é não omitir as informações, nem as apresentar todas aos membros do grupo. É necessário buscar um meio termo entre a insuficiência e a sobrecarga de informações. Uma solução para isto é utilizar a idéia de filtragem e agrupamento das informações não somente na interface, mas também na escolha destas informações e na escolha do momento da apresentação. Isso significa permitir aos próprios membros colocarem critérios particulares para a filtragem e agrupamento de informações, com base nos seus interesses e aspirações particulares.

Na agenda *No Risk Planning*, a quantidade de informações necessárias para que os membros dos grupos tenham a exata noção de contexto de seu trabalho, é oferecida pelas mensagens postadas, pela agenda propriamente dita e pelos *e-mails* que circulam entre os membros do grupo. A quantidade de informações disponibilizadas nestes meios possibilitará ao usuário a sua contextualização no grupo ao qual pertence.

## **2.6 – Considerações Finais**

Este capítulo apresentou os princípios de CSCW e os conceitos estudados sobre *groupware*, que foram fundamentais para entendimento da finalidade básica da agenda *No Risk Planning*. Foi feita uma descrição do protótipo da agenda, considerando os parâmetros das áreas de CSCW e *groupware*.

A seguir, foi apresentada uma análise de seis questões de contexto (*awareness*), que constitui um *framework* que resume as características consideradas mais importantes para se

ponderar que tipo de suporte uma ferramenta de *groupware* deve priorizar. Esta análise representou um estudo importante para se justificar quais solicitações dos novos usuários da agenda deveria ser atendida ou não.

Considerando-se como ponto de partida o protótipo da agenda *No Risk Planning* apresentado neste capítulo, e a partir de diversas solicitações de mudanças de visão dos usuários e de inserção de novas funcionalidades na agenda, foi necessária a evolução da agenda. Assim, as mudanças motivaram o emprego de um processo de evolução da agenda, com base na Engenharia de *Web* e algumas técnicas de *Programação Extrema (XP)*, conforme descritos no próximo capítulo.

## Capítulo 3

---

### Processo de Desenvolvimento da *No Risk Planning*

O processo de desenvolvimento da agenda eletrônica *No Risk Planning* cumpriu as etapas do ciclo de vida de Engenharia de *Web* aplicando-se algumas regras de Programação Extrema. Essas atividades foram realizadas com o objetivo de viabilizar a implementação das novas versões da agenda, atendendo aos novos requisitos solicitados e possibilitando que o processo adotado garantisse certas qualidades ao produto e que o processo, ganhando melhorias.

#### 3.1 - Engenharia de *Web*

O conjunto básico de atividades, que na literatura especializada é encontrado sob o nome de Engenharia de *Web*, definiu todo o processo de desenvolvimento de evolução da agenda *No Risk Planning*, e por meio dele todas as mudanças realizadas foram documentadas. Nas próximas subseções, são descritos os conceitos relativos à Engenharia de *Web*, revisados da literatura.

##### 3.1.1 – Introdução

Em um curto período de tempo, a Internet e a *World Wide Web* se tornaram ambientes ubíquos ultrapassando outros produtos tecnológicos desenvolvidos em toda a história da humanidade. As aplicações para Internet também cresceram rapidamente em escopo e extensão de uso afetando, significativamente, todos os aspectos de nossa vida cotidiana [ATHULA & MURUGESAN, 2001].

A Internet, atualmente, se apresenta como ponto-chave para o desenvolvimento de diversas aplicações em diferentes áreas de conhecimento. Nota-se ainda que as formas de criação de aplicações na Internet mudam constantemente, assim como também mudam os processos de qualidade, integridade e manutenibilidade [MURUGESAN & DESHPANDE, 2002]. Observa-se, no entanto, um crescimento caótico, pois a ênfase está na implementação do aplicativo e pouco valor tem sido dado ao processo de desenvolvimento utilizado.



Este crescimento caótico da tecnologia de *Web* nos remete ao passado, a uma outra época: os primeiros dias do *software*. Era uma época de pouca disciplina, mas de muito entusiasmo, em que os programadores freqüentemente integravam sistemas (uns bons, outros ruins). Como foi observado no período da Crise do *Software*, a ausência de um processo disciplinado para o desenvolvimento de sistemas gerou diversos problemas no desenvolvimento, disponibilização, entrega e manutenção desses sistemas. Em relação ao desenvolvimento de aplicativos para Internet, problemas semelhantes serão enfrentados caso não sejam adotados processos sistemáticos que promovam a organização das informações relevantes para o desenvolvimento e manutenção dos aplicativos [PRESSMAN, 2002]. De acordo com Isaak [ISAAK, 2002], a Engenharia de *Web*, aumenta a satisfação do usuário com o sistema e diminui os custos de manutenção dos aplicativos.

Para a construção destes sistemas, os desenvolvedores devem utilizar metodologias e processos disciplinados, boas ferramentas CASE e um conjunto coerente de diretrizes para desenvolvimento de software. A Engenharia de *Web* supre estas necessidades na medida em que tenta minimizar riscos e melhorar a manutenibilidade e qualidade de um aplicativo para Internet [ATHULA & MURUGESAN, 2001].

A Engenharia de *Web* se refere ao estabelecimento e uso de princípios científicos sólidos, de engenharia e de gestão e abordagens disciplinadas e sistemáticas para o desenvolvimento, disponibilização e manutenção de sistemas e aplicações de alta qualidade baseados na *Web*. O processo de Engenharia de *Web* considera três domínios: o conteúdo da informação, a estética e a *performance* [ATHULA & MURUGESAN, 2001].

O principal objetivo da Engenharia de *Web* é gerenciar, com sucesso, a diversidade e a complexidade do processo de desenvolvimento de aplicações de Internet e, com isso, evitar falhas em potencial que possam acarretar várias implicações [GINIGE & MURUGESAN, 2001].

### **3.1.2 – Histórico**

A preocupação com a qualidade de *websites* foi intensificada quando a Internet passou a ser usada comercialmente. Este processo de construção de *websites*, chamado de Engenharia de *Web*, assim como a Internet comercial propriamente dita, é uma área bem recente da computação e surgiu em 1995.

Os primeiros artigos discutindo os problemas e soluções relacionados à Engenharia de *Web* apareceram nas *World Wide Web Conferences* em 1995/1996. Em Paris na França, Bebo

White organizou um tutorial sobre a Documentação da Engenharia de *Web* e Gellersen, Wicke, e Gaedke discutiram o ciclo de vida de um *software* baseado na *Web* na *6th World Wide Web Conference* em Santa Clara, CA, EUA [GAEDKE, 2002]. Na *7º World Wide Web Conference* em Brisbane, ocorrida na Austrália em 1997, San Murugesan organizou o primeiro *workshop* com temas relacionados à Engenharia de *Web*.

Nos anos de 1997 a 2001 diversos outros eventos foram organizados com foco na Engenharia de *Web*. Estes eventos incluem séries de *workshops* nas *World Wide Web Conferences* e a *International Conference on Software Engineering (ICSE)*, e mini-trilhas na *Hawaii'n International Conference on System Science*.

Em 2001, Daniel Schwabe (Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio) organizou o primeiro *workshop* internacional a respeito de Tecnologia de Software Orientada a *Web*. Também em 2001, pela primeira vez, o tema Engenharia de *Web* se tornou parte do programa principal da 11ª Conferência Internacional de *World Wide Web (WWW2002)*. A Trilha de Engenharia de *Web* foi uma trilha muito bem sucedida, coordenada por Yogesh Deshpande, Martin Gaedke, San Murugesan e Daniel Schwabe [GAEDKE, 2002].

Atualmente, o interesse pela Engenharia de *Web* tem aumentado significativamente. Esta nova área tem atraído profissionais envolvidos com engenharia de *software*, sistemas distribuídos, recuperação da informação, dentre outras [ATHULA & MURUGESAN, 2001].

### 3.1.3 – Características de Aplicativos da *Web*

Os sistemas e aplicativos baseados na Internet diferem de *softwares* convencionais ou enfatizam determinados aspectos devido as suas características peculiares.

O escopo e a variedade de aplicações de Internet variam muito de acordo com sua funcionalidade englobando desde aplicativos de pequena escala e poucos serviços até aplicativos distribuídos de empresas disponibilizados através da Internet, intranets e extranets. Algumas das características principais destes aplicativos estão resumidas na Tabela 3.1.

Analisando a Tabela 3.1, é possível concluir que a agenda *No Risk Planning*, objeto de estudo desta dissertação, possui características dos dois tipos de sistemas para Internet (Sistemas de Internet Simples e Sistemas de Internet Complexos), por possuir as seguintes características:

- Páginas complexas que exibem, além de texto, imagens dos menus disponibilizados ao usuário;

- Informação é dinâmica e muda de acordo com o tempo e as necessidades de usuário. As agendas armazenadas mudam de acordo com o usuário e suas tarefas;
- Navegação simples e intuitiva com menus e funções auto-explicativas;
- Integrados a bases de dados onde são armazenados os horários e usuários do sistema;
- Requer alta *performance* e disponibilidade contínua para que os usuários possam confiar que a agenda estará disponível;
- Desenvolvido por uma pessoa ou por uma pequena equipe;
- Usado para disseminação de informações entre pessoas do mesmo grupo;

**Tabela 3.1:** Diferenças entre sistemas Simples e Complexos de Internet [ATHULA & MURUGESAN, 2001]

Sistemas de Internet Simples	Sistemas de Internet Complexos
Páginas de Internet apresentando basicamente texto.	Páginas de Internet complexas.
Conteúdo da informação geralmente não muda (estática).	Informação é dinâmica e muda de acordo com o tempo e as necessidades de usuário.
Navegação simples.	Dificuldade em navegar e encontrar a informação desejada.
Sistemas <i>stand alone</i> .	Integrados a bases de dados e outros sistemas de planejamento e marcação.
Alta <i>performance</i> não é o principal requisito.	Requer alta <i>performance</i> e disponibilidade contínua.
Desenvolvido por somente uma pessoa ou um pequeno time.	Requer um grande time de desenvolvimento com especialistas em diferentes áreas.
Usado para disseminação de informações.	Usado também em aplicações críticas.

Identificar estas características na agenda *No Risk Planning* é importante para utilizar no processo de Engenharia de *Web*. Segundo Pressman [PRESSMAN, 2002], as características de sistemas de *Web* são:

- Uso intensivo de redes: os aplicativos de Internet residem em uma rede e precisam servir às necessidades de uma comunidade diversificada de clientes e possíveis usuários;
- Evolução contínua: as aplicações de Internet evoluem de forma contínua, ou seja, seus requisitos estão em constante mudança devido a alguns fatores: inicialmente não se conhece o universo de possíveis usuários do aplicativo, como também não se conhecem todas as funcionalidades que podem ser oferecidas a estes usuários;
- Direcionada a conteúdo: em muitos casos, a função primordial de um sistema de *Web* é usar a hipermídia para apresentar conteúdo de texto, gráficos, de áudio e vídeo ao usuário

final. Por causa disso, a atenção dispensada ao conteúdo de um aplicativo baseado na Internet é de fundamental importância.

Estas características fazem com que o sistema ou aplicativo baseado na Internet tenha um processo de desenvolvimento diferente dos *softwares* convencionais. Assim, as características importantes no *processo* de desenvolvimento de sistemas de *Web* são [PRESSMAN, 2002]:

- **Imediatismo:** um sistema ou aplicativo baseado na Internet deve ser desenvolvido em curto prazo. É comum que o sistema completo tenha que estar disponível *online* em dias ou até semanas. Para tanto, os desenvolvedores precisam usar métodos de planejamento, análise, projeto, implementação e teste que sejam adaptados ao cronograma “apertado”;
- **Segurança:** como a Internet é aberta a todas as pessoas, é fundamental garantir a segurança das informações contidas no aplicativo e os serviços oferecidos por este sistema. Para que isso ocorra, medidas de segurança precisam ser implementadas em toda a infra-estrutura que apóia este aplicativo de *Web* e no aplicativo propriamente dito;
- **Estética:** a interface é a primeira impressão que o usuário terá do sistema. Portanto, a estética envolvendo a interface deve ser bem desenvolvida de forma que o usuário possa se interessar pelo uso do sistema.

Estas características direcionam o desenvolvimento de sistemas e aplicativos de *Web* e é através destas características que eventuais mudanças de requisitos podem acontecer e, com isso, fazer com que um novo processo seja iniciado.

Segundo McDonald & Welland [MCDONALD & WELLAND, 2001], um processo de Engenharia de *Web* para ser bem sucedido deve ter as seguintes características:

- Pequeno tempo de desenvolvimento do ciclo de vida;
- Entrega de soluções rápidas;
- Equipe de desenvolvedores multidisciplinar;
- Equipes de desenvolvedores pequenas trabalhando em paralelo em diferentes tarefas;
- Foco no processo de Análise e Avaliação;
- Mudança de requisitos e testes;
- Forte manutenção para garantir que o aplicativo terá longevidade e atenderá realmente aos requisitos de usuário;

Estas características evidenciam o surgimento de um novo conceito dentro de Engenharia de *Software*, denominado Engenharia de *Web*, que trata especialmente do desenvolvimento de aplicativos voltados para Internet.

Segundo Deshpande & Hansen [DESHPANDE & HANSEN, 2001], a Engenharia de *Web* abrange diversas áreas de conhecimento. No processo de desenvolvimento da agenda *No Risk Planning* foram as seguintes áreas foram consideradas:

- **Interface de Usuário:** o conceito de usuário de um sistema de Internet é muito abstrato, pois abrange uma grande variedade de pessoas de diferentes formações e níveis intelectuais. Assim, a interface da agenda procurou ser a mais simples possível para que não fosse um empecilho ao seu uso, e conquistasse sempre novos adeptos;
- **Questões de Segurança:** a segurança na agenda advém do fato de cada usuário ter a sua própria senha e, com isso, fica impossibilitado de manipulação e modificação de agendas de outros usuários. Esta senha é criptografada e armazenada no servidor da agenda;
- **Questões Lógicas, Sociais e Éticas:** estas questões dizem respeito a aspectos de discriminação de pessoas, direito de autoria e invasão de privacidade. Na agenda não há qualquer tipo de discriminação dos seus usuários. O direito de autoria está preservado porque o que cada usuário colocar de arquivos ou postar qualquer assunto no bate-papo, seu nome estará associado à informação armazenada. A invasão de privacidade também é respeitada devido ao uso de senhas pessoais por parte dos usuários como visto no item anterior;
- **Gerenciamento de Páginas:** o gerenciamento de páginas abrange o controle de *links* quebrados e a atualização de informações contidas nas páginas. O processo de gerenciamento de páginas é dividido entre os usuários e o administrador do sistema. O administrador cuida dos *links* e informações relacionadas ao sistema, enquanto os usuários cuidam dos *links* e informações referentes a sua área pessoal ou área de trabalho em grupo;
- **Estruturação da Informação:** a informação apresentada de maneira estruturada facilita o entendimento das pessoas que estão acessando a página em questão. Na agenda, a estruturação é realizada principalmente pela organização das opções na página principal e pelo pequeno número de níveis na estrutura do *site*. A estruturação também ajuda os usuários na compreensão do funcionamento global da agenda;
- **Desenvolvimento de Sistemas Evolucionários:** o desenvolvimento de sistemas com constantes solicitações de mudanças, na forma de requisitos de usuários, é um dos

maiores desafios da Engenharia de *Web* e, por isso, é uma questão que deve ser sempre considerada. A agenda, tipicamente, teve seu processo constantemente orientado aos novos requisitos;

***Performance de Download:*** o tempo de carregamento da página pode afetar a qualidade de interação do usuário com o sistema. No caso da agenda, este tempo de carregamento é muito baixo, pois a maior parte do que deve ser carregado é texto e possui somente algumas poucas e pequenas figuras que constituem, em sua maior parte, as opções da página principal, as laterais e parte inferior da interface da agenda;

#### **3.1.4 – Ciclo de Vida em Engenharia de *Web***

O principal objetivo da Engenharia de *Web* é gerenciar, com sucesso, a diversidade e a complexidade do processo de desenvolvimento de aplicações de Internet [GINIGE & MURUGESAN, 2001]. O desenvolvimento de aplicações baseadas em Internet diferente do desenvolvimento de *softwares* tradicionais na natureza, no ciclo de vida, no seu desenvolvimento e na sua manutenção.

Os sistemas baseados em Internet, comumente têm seus requisitos mudando e aumentando rapidamente, além de seu conteúdo e funcionalidades durante o seu ciclo de vida. Assim, o desenvolvimento de sistemas baseados na Internet é uma atividade contínua sem atualizações específicas como ocorrem, geralmente, em *software* convencional [ATHULA & MURUGESAN, 2001].

Nos últimos anos, muitos modelos e metodologias foram propostos para dar suporte ao desenvolvimento de aplicativos de Internet. Estas metodologias devem ajudar os desenvolvedores a gerenciar a complexidade de um sistema de Internet que requer uma variedade de atividades, por exemplo, a sua organização estrutural e escolha de conteúdo e a maneira como este conteúdo deve ser apresentado ao usuário [COSTAGLIOLA, FERRUCCI & FRANCESE, 2002].

Assim, o modelo espiral parece ser o mais adequado, por ser geralmente usado para desenvolver projetos que possuem alto grau de incerteza de seus requisitos e da natureza da tecnologia que muda constantemente. As aplicações na Internet têm estas características. Com base neste modelo, a cada incremento, a aplicação “cresce”, ou seja, aumentam suas funcionalidades, e os requisitos vão sendo melhor identificados, ao passo que os riscos envolvidos tendem a ser reduzidos [COSTAGLIOLA, FERRUCCI & FRANCESE, 2002].

Dessa forma, o modelo espiral, sugerido por Pressman [PRESSMAN, 2002], foi adotado como modelo de ciclo de vida para evolução da agenda. As etapas do modelo espiral são descritas a seguir.

#### *3.1.4.1 – Formulação*

Nesta primeira etapa de desenvolvimento, as metas e objetivos da aplicação a ser construída devem ser identificados. As seguintes tarefas devem ser realizadas nesta etapa de formulação:

- Determinar a função global do sistema;
- Identificar as metas, considerando-se que há dois tipos de metas:
  - o Metas Informativas, que indicam a intenção de fornecer conteúdo específico e/ou informação ao usuário final;
  - o Metas Funcionais, que indicam a capacidade de realizar alguma tarefa dentro do aplicativo;
- Traçar um perfil de usuário, para que o conteúdo/estética seja desenvolvido para este tipo de usuário;
- Definir o escopo.

Nesta primeira etapa, deve ter uma noção global do projeto para verificar se a sua existência é realmente necessária e se há um público-alvo para o tipo de aplicação a ser desenvolvido. Mas, em muitos casos, não é possível especificar exatamente o que um aplicativo de Internet irá fazer ou irá conter no início de seu processo de desenvolvimento porque sua estrutura e funcionalidades evoluem continuamente [GINIGE & MURUGESAN, 2001].

#### *3.1.4.2 – Planejamento*

Nesta etapa do processo, estima-se o custo global do projeto, avalia-se os riscos associados com o esforço do desenvolvimento e se define o cronograma inicial para desenvolvimento da aplicação. Este cronograma deve ser construído com base no número de pessoas trabalhando no projeto, na urgência requerida para se ter o aplicativo rodando na Internet e nos custos envolvidos.

#### 3.1.4.3 – Análise

A etapa de análise estabelece os requisitos técnicos do aplicativo de Internet e identifica os itens de conteúdo que serão incorporados a ele. Quatro tipos de análises ocorrem em um processo de Engenharia de Web:

- Análise de *Conteúdo*: identifica todo o conteúdo que deverá ser apresentado pelo aplicativo de Web ao usuário final e inclui desde texto a gráficos, figuras, vídeo e áudio. Pode-se fazer uma modelagem de dados para identificar e descrever cada um dos objetos de dados a ser utilizado no aplicativo;
- Análise *Funcional*: identifica as funções que o usuário terá à sua disposição, obtidas pelos seus cenários de uso. Todas as operações são descritas em detalhe;
- Análise de *Interação*: identifica o modo como o usuário do aplicativo irá interagir com o mesmo, obtido a partir de casos de uso;
- Análise de *Configuração*: identifica o ambiente em que o aplicativo estará sendo executado e a infra-estrutura disponível para o mesmo. Esta descrição de configuração ocorre de maneira detalhada.

A fase de análise da agenda eletrônica *No Risk Planning* foi realizada e reportada em [RIBEIRO, FORTES & FREIRE, 2002], como parte da documentação da versão 1.0 da agenda *No Risk Planning*, utilizando-se a linguagem *Unified Modeling Language* (UML).

#### 3.1.4.4 – Engenharia

A etapa de Engenharia define como serão criadas as páginas propriamente ditas e possui dois processos que são executados em paralelo por equipes diferentes:

- Processo 1: Projeto de Conteúdo e Produção;
- Processo 2: Projeto Arquitetural, Projeto de Navegação e Projeto de Interface.

O Processo 1 se refere à produção e/ou aquisição de todo o conteúdo que será disponibilizado no aplicativo. Este processo envolve desde a geração de figuras e gráficos a serem apresentados até dados a serem inseridos em banco de dados. Neste processo também são criados o *layout* detalhado do conteúdo da informação e sua estrutura global.

O Processo 2 se refere a concepção de três projetos distintos, a saber:

- Projeto *Arquitetural* - define a estrutura global de hipermídia do aplicativo de Internet, bem como a utilização de padrões de projeto e estruturas pré-definidas que possam



permitir sua reutilização. Na agenda *No Risk Planning* o projeto arquitetural foi bem definido, possibilitando o reuso de partes do código em seu crescimento, conforme será descrito adiante (Capítulo 4).

- Projeto *Navegacional* - define os caminhos de navegação que irão permitir ao usuário ter acesso à arquitetura proposta no Projeto Arquitetural. Para isso, o projetista deve identificar a semântica de navegação para diferentes usuários no aplicativo e definir a sintaxe para realizar esta navegação. Na agenda *No Risk Planning* foram criados somente dois modelos navegacionais: o modelo do administrador de usuários e o modelo de usuário propriamente dito. Com o processo de crescimento, novos modelos navegacionais foram incluídos, como o modelo de professores, o modelo de disciplinas e o modelo de administrador de disciplinas que serão detalhados mais adiante (Capítulo 4). Para determinado modelo navegacional, deve-se definir os diferentes modelos de interface, de acordo com o dispositivo de interface que o usuário utiliza, isto é, seu navegador (*browser*) [SCHWABE et al, 2001]. A estrutura navegacional deve possibilitar que diferentes maneiras de organização sejam exibidas, por exemplo, linear, em rede, hierárquica, matriz e/ou uma combinação entre estes tipos, dependendo da natureza da informação e do tipo de usuário [COSTAGLIOLA, FERRUCCI & FRANCESE, 2002].
- Projeto de *Interface* - após a concepção dos dois projetos anteriores, o último projeto a ser elaborado na etapa de Engenharia cuida do *layout* de tudo o que foi criado até então. A interface deve ser muito bem definida, com critérios de usabilidade, para que ela não seja empecilho a uma boa interação do usuário com o aplicativo. Assim, deve-se entender as tarefas do usuário antes de se decidir quais facilidades de navegação serão incluídas. A interface deve ajudar o usuário a navegar pela informação dando a ele sinais e retorno de suas ações e apresentando a informação de modo claro e significativo [SCHWABE et al, 2001]. Algumas orientações, de Nielsen & Wagner [NIELSEN & WAGNER, 1996], que foram seguidas durante o processo da agenda, para a criação de boa interface foram:
  - Evitar, ao máximo, erros de servidor;
  - Evitar grandes quantidades de textos;
  - Evite sinais de “Em Construção” para não criar expectativas nos usuários;
  - Imagens devem caber na janela do usuário sem precisar de rolagem;
  - Menus de navegação e barras de título devem estar sempre disponíveis;
  - A estética não deve se sobrepor à funcionalidade do aplicativo;
  - Usar opções de navegação óbvias para evitar confusão para os usuários;

Os dois processos da etapa de Engenharia, executados em paralelo, constituem a principal demanda de esforços na construção de um aplicativo de Internet e, geralmente é onde se aloca a maior parte do tempo do cronograma previsto na etapa de planejamento.

#### 3.1.4.5 – Geração e Testes de Página

Após a definição dos projetos e do conteúdo, deve-se gerar as páginas, com o suporte de ferramentas existentes no mercado.

E construído o aplicativo, testes devem ser realizados para verificação de possíveis erros de implementação e validação de requisitos. Os testes, em geral, incluem os seguintes tipos:

- Teste de Conteúdo: semelhante a uma revisão de um texto escrito, o teste de conteúdo é utilizado para corrigir eventuais erros de grafia, erros gramaticais, erros de consistência, erros de representação gráfica e erros de referência cruzada;
- Teste de Projeto Arquitetural e Navegacional: o uso de casos de uso é utilizado para encontrar falhas no projeto arquitetural e no projeto navegacional, pois simulam a navegação de um usuário comum do aplicativo de Internet;
- Testes de Unidade: cada página do aplicativo engloba conteúdo, ligações de navegação (*links*) e elementos de processamento que devem estar funcionando de maneira adequada. O teste de unidade é realizado para garantir que esses elementos da página estejam funcionando da maneira como foram projetados;
- Testes de Integração: utilizado para verificar se a integração entre as partes de um aplicativo de Internet está funcionando de modo coerente. Dependendo da estrutura do aplicativo, estes testes podem ser semelhantes a testes de integração de *softwares* convencionais ou a testes de aplicações de sistemas orientados a objetos;
- Testes de Funcionalidade e Disponibilidade de Conteúdo: testes para verificar se todas as funcionalidades desejadas estão funcionando corretamente. Casos de uso auxiliam na descoberta de eventuais erros de funcionalidade e disponibilidade de conteúdo;
- Testes de Configuração: testa o funcionamento do aplicativo em diferentes *browsers*, plataformas de *hardware* e protocolos de aplicação;
- Teste por um grupo de usuários: o teste final engloba um grupo de usuários potenciais do aplicativo de Internet em questão. Este grupo deve ser monitorado para que se faça uma verificação no processo de interação.

A fase de testes, embora muitas vezes não seja disciplina em processos de desenvolvimento de aplicações *Web*, deve ser executada para que o aplicativo possua o mínimo de erros possíveis. Diversas vezes, uma boa fase de testes pode evitar que ocorra um novo ciclo de vida do aplicativo.

#### 3.1.4.6 – Avaliação do Cliente

Após o aplicativo ter sido testado, o cliente fará a sua verificação. Se os requisitos foram atendidos, o processo de Engenharia de *Web* termina uma iteração do modelo espiral. No caso da agenda *No Risk Planning*, objeto de estudo desta dissertação, este processo de desenvolvimento aconteceu de forma contínua e atendendo os requisitos à medida que eles eram mudados e/ou acrescentados.

Mas a evolução se tornava cada vez mais rápida, exigindo uma reação mais pró-ativa diante das solicitações e do surgimento de novos tipos de usuários. Assim, a utilização de algumas regras de Programação Extrema (em inglês, *Extreme Programming – XP*) acopladas à Engenharia de *Web* foram estudadas de maneira a contribuir para melhoria no processo rápido da agenda. A utilização de Programação Extrema foi escolhida porque a equipe dos dois desenvolvedores (este aluno de mestrado e um aluno de graduação), envolvida com a evolução da agenda, dedicava grande parte do tempo com o entendimento do código dos programas e com acompanhamento das demais tarefas para garantia da qualidade do código gerado, características essas que são relevantes à prática da Programação Extrema, conforme foi estudada e que é descrita a seguir.

### **3.2 - Programação Extrema na Agenda *No Risk Planning***

O uso de algumas regras de Programação Extrema no desenvolvimento da agenda *No Risk Planning* foi ocasionado pelo fato deste desenvolvimento acontecer de maneira muito rápida e contínua. Nas próximas subseções, são descritos os conceitos relativos à Programação Extrema, revisados da literatura.

### 3.2.1 – Definição

Programação Extrema é definida como uma metodologia “leve” (no sentido de não ser rígida ou que sobrecarregue os seus adeptos), criada para equipes pequenas e médias de desenvolvedores de *software*. É chamada de extrema porque utiliza níveis extremos de boas práticas de programação. Seu foco está nas atividades de codificação, testes, obtenção de conhecimento do usuário e *design* [SMITH & STOECKLIN, 2001].

Também conhecida pela sigla XP (de *eXtreme Programming*), a metodologia enfatiza o esforço de desenvolvimento em atividades para entrega com qualidade das funcionalidades ao usuário, o mais rápido possível. Assim, freqüentemente, a documentação é restrita às histórias de clientes contadas em cartões (*story cards*), códigos-fonte e códigos de teste [MAURER, 2002]. XP foi criada para uso por equipes pequenas e médias, localizadas num mesmo ambiente de desenvolvimento de *softwares*, cujos requisitos não sejam claramente definidos ou que estejam em constante mudança. Além da metodologia dar ênfase ao trabalho em equipe, considera primordial o envolvimento do cliente no processo do *software* [SMITH & STOECKLIN, 2001].

A metodologia XP contraria algumas das práticas de desenvolvimento de *software* tradicionais em alguns aspectos, como: a documentação é quase inexistente, não há especificação formal de *software*, não há distinção entre a fase de desenvolvimento e a fase de testes e existe uma recomendação para que somente a implementação mais simples seja desenvolvida [MÜLLER & TICHY, 2001].

Assim, XP preconiza que os desenvolvedores estejam livres de trabalho desnecessário (documentação extensa, por exemplo) para que possam se concentrar no propósito principal do desenvolvimento de um *software*: escrever código de boa qualidade. Na essência, a idéia é implementar a solução mais simples possível atendendo os atuais requisitos pedidos pelo cliente [SCHNEIDER & JOHNSTON, 2003].

Segundo James Caristi [CARISTI, 2002], XP é uma visão relativamente nova do desenvolvimento de *software* que tem atraído muita atenção de novos profissionais e os educadores na área de Ciência da Computação deveriam conhecer mais sobre as suas práticas.

### 3.2.2 – Histórico

Por muitos anos, desenvolvedores do mundo inteiro usaram um conjunto de práticas de desenvolvimento *ad hoc* que eram interativas, leves e agressivas. Antigamente, avaliações de

comunicação entre pessoas eram realizadas através de documentos, avaliações de bons códigos eram realizadas por meio de diagramas, avaliações de interações rápidas eram realizadas através do retorno do cliente e avaliações do *design* de software eram realizadas para verificar dependências. Obteve-se sucesso através destes processos, mas eles estavam longe de ser uma definição de processo [NEWKIRK & MARTIN, 2000].

Em 1996, Kent Beck, o criador da metodologia XP, introduziu suas práticas enquanto conduzia o C3 (*Chrysler Comprehensive Compensation*) que era um grande projeto para reescrever a folha de pagamento da Daimler-Chrysler. Desde que completou este projeto, Kent Beck escreveu o livro *Extreme Programming Explained: Embrace Change* (Addison-Wesley, 1999) e ganhou o prêmio *Software Development Jolt Product Excellence Award* [ENGLISH, 2002]. O seu trabalho e os resultados obtidos foram estudados cuidadosamente pela comunidade científica internacional. XP confirmou a eficácia de muitas das práticas mencionadas na seção anterior, mas desafiou outras. XP avalia comunicação humana através de papel, avalia bons códigos através de diagramas, avalia interações rápidas através de retorno do cliente, mas não avalia a “aparência” do *software*, através de seu *design*. Ao invés disso, XP estimula um bom *design*, que possa evoluir à medida que os requisitos evoluem [NEWKIRK & MARTIN, 2000].

### 3.2.3 – Características de XP

XP possui algumas características particulares do seu processo de desenvolvimento que o diferencia dos métodos tradicionais de desenvolvimento de *software*, que são:

- Falta de escalabilidade: como a documentação é substituída por comunicação entre as pessoas, um fator limitante para a escalabilidade do tamanho da equipe que utiliza XP como método de desenvolvimento é evidenciado. Se um novo membro é adicionado à equipe, ele terá certa dificuldade inicial para acompanhar o que está acontecendo e entender o que já foi desenvolvido [MAURER, 2002];
- Equipes em um mesmo espaço físico: XP requer que todos os membros da equipe estejam dividindo o mesmo espaço de trabalho. Esta medida melhora a comunicação face-a-face e facilita a resolução de eventuais dúvidas [MAURER, 2002];
- XP dá ênfase ao desenvolvimento de produtos em detrimento de documentos e revisões formais, que são frequentemente utilizados para demonstrar progresso em desenvolvimento de *softwares* convencionais [REIFER, 2002];

- XP dá menos ênfase em requisitos e arquitetura do que os métodos clássicos de desenvolvimento [REIFER, 2002];
- XP é caracterizada por pequenos ciclos, planejamento incremental (ou seja, que vai evoluindo continuamente), *design* evolucionário e pela habilidade de corresponder às mudanças de planos de negócios [DEURSEN, 2001].

Existem ainda outras características de Programação Extrema, que foram estudadas, e serão descritas quando as suas “doze práticas” (ou regras) forem detalhadas mais adiante (Seção 3.2.5) nesta dissertação.

### 3.2.4 – Valores de XP

Para servir de referência sobre o progresso de um processo de desenvolvimento de *software* utilizando XP, são usados os valores de XP como parâmetros. São quatro valores que identificam todas as necessidades que envolvem a prática de uma boa programação e são descritos na Tabela 3.2 a seguir [NEWKIRK, 2002], juntamente com uma análise de uso de cada valor, durante o processo de evolução da agenda. O uso dos valores de XP no desenvolvimento de um aplicativo propicia o devido suporte para a aplicação das “doze regras” de XP, descritas na próxima seção.

**Tabela 3.2:** Valores utilizados em Programação Extrema

Valores	Definição	Uso no desenvolvimento da agenda
Comunicação	Envolve a comunicação entre todos os membros da equipe, incluindo o cliente. É a base para aquisição de novos conhecimentos e da percepção do projeto como um todo.	Valor utilizado no desenvolvimento da agenda <i>No Risk Planning</i> , pois a equipe estava sempre em contato no dia a dia da universidade.
Simplicidade	Implica em fazer o que foi pedido da maneira mais simples possível e sem tentar antecipar a realização de novas tarefas.	Valor utilizado no desenvolvimento da agenda porque as mudanças eram realizadas na medida em que eram solicitadas.
<i>Feedback</i>	O uso de pequenas versões, integração contínua e testes provê um retorno claro do processo de desenvolvimento do <i>software</i> .	Apesar de fazermos os testes somente depois da implementação do código, o retorno do desenvolvimento também se deu de forma clara.
Coragem	Coragem para implementar somente o que foi pedido sem tentar antecipar novas tarefas.	Valor aplicado ao desenvolvimento da agenda, devido ao tempo restrito.

### 3.2.5 – Práticas de XP x Desenvolvimento da Agenda *No Risk Planning*

A metodologia XP provê meios para efetuar o desenvolvimento de software de forma rápida e dinâmica e consiste na aplicação de doze práticas. Em XP, o suporte entre essas doze práticas é de que a fragilidade de determinada prática é compensada pela força de outra [BECK, 2000]

Os fundamentos da metodologia se baseiam nas seguintes “doze práticas” [WILLIAMS & UPCHURCH, 2001]:

- 1. Jogo do Planejamento (*Planning Game*)** – O Planejamento do desenvolvimento é feito com base em “cartões de histórias” do cliente, onde se descreve que tipo de operações o cliente quer que o sistema seja capaz de realizar. Os cartões de histórias definem o que será implementado, ajudam a medir a velocidade de desenvolvimento de um projeto e constitui um ponto-chave na comunicação entre o cliente e os desenvolvedores [RAMACHANDRAN & SHUKLA, 2002]. Os clientes devem ter certeza de que as histórias fornecidas aos desenvolvedores sejam significativas ao projeto [NEWKIRK & MARTIN, 2000]. Se uma história está muito complexa, ela é dividida em outras histórias menores, que juntas realizam a mesma função da história inicial [DEURSEN, 2001]. Quando os cartões de história estão completos com os requisitos, prioridades e estimativas, o jogo do planejamento se inicia. Os cartões são colocados em uma mesa em que estão sentados o cliente e os desenvolvedores. O cliente diz qual a seqüência de cartões que deve ser implementada, de acordo com suas prioridades [WILLIAMS, 2003].
- 2. Pequenas Atualizações** – XP acelera o ritmo do desenvolvimento espiral tendo pequenas novas versões em um período de 3-4 semanas. As pequenas atualizações permitem um retorno mais rápido no investimento dado pelo cliente e dá aos mesmos uma melhor visão do desenvolvimento do projeto [BECK, 2000].
- 3. Metáfora** – A integridade conceitual do sistema deve se basear em metáforas simples, que explicam a essência do funcionamento do sistema. Com a metáfora, o sistema e sua arquitetura podem ser compreendidas, utilizando um conjunto de nomes simples e comuns para cada parte do sistema [RAMACHANDRAN & SHUKLA, 2002].
- 4. Design Simples** – XP afirma que o design do *software* deve ser o mais simples possível, e que não se deve planejá-lo de acordo com o que deverá ser utilizado no futuro. *Design* simples deve ser entendido como tendo as seguintes características: o código e os testes comunicam tudo o que o desenvolvedor quer comunicar, o sistema

não possui código duplicado, possui o menor número de classes possível e cada classe possui o menor número de métodos possível [BECK, 2000];

5. **Testes de Unidade (*Unit Tests*)** – XP recomenda que se deve escrever unidades de teste *antes* de escrever o código propriamente dito. Na criação das unidades de teste, deve-se usar a mesma linguagem usada no desenvolvimento do programa em si [DEURSEN, 2001]. O uso de unidades de teste direciona todo o desenvolvimento de *software* [SMITH & STOECKLIN, 2001]. As unidades de teste provêem um retorno imediato quando algo não está funcionando de acordo com o que foi programado [RASMUSSEN, 2002]. Os clientes se reúnem com a equipe de desenvolvedores para averiguar se todos os requisitos dos cartões de histórias foram cumpridos [BECK, 2000]. Os programadores escrevem somente o código que deve ser executado corretamente nos testes; eles não desenvolvem uma estrutura maior para garantir algum requisito antecipado, mas ainda não requisitado [WILLIAMS, 2003].
6. **Refatoramento (*Refactoring*)** – É o processo de melhorar a estrutura do código preservando a funcionalidade desse. O refatoramento pode ocorrer em várias etapas durante o processo de desenvolvimento e deve ser aplicado quando algo errado é detectado, como nas seguintes situações: quando se descobre que duas funções estão realizando a mesma tarefa, quando existem duas classes contendo os mesmos métodos e quando o código parece estar fora do lugar [SMITH & STOECKLIN, 2001] [DEURSEN, 2001].
7. **Programação em Pares** – Durante todo o trabalho, dois programadores programam lado-a-lado no mesmo computador, colaborando no mesmo *design*, algoritmo, código ou teste. Geralmente, os pares são divididos da seguinte maneira: enquanto um digita as linhas de código, o outro revisa e prevê os próximos passos [SMITH & STOECKLIN, 2001] [DEURSEN, 2001]. Este tipo de programação aumenta a confiança dos desenvolvedores na qualidade do código desenvolvido e na cobertura de todos os requisitos exigidos pelo cliente [RAMACHANDRAN & SHUKLA, 2002] [SMITH & STOECKLIN, 2001]. Para ser mais produtivo, é interessante que os dois parceiros tenham diferentes experiências anteriores e usem estratégias diferentes para solucionar determinado problema [DEURSEN, 2001];
8. **Propriedade Coletiva do Código** – Nenhum programador é proprietário do código que escreve, toda a equipe tem acesso ao código gerado por cada membro. Como toda a equipe é proprietária do código, então qualquer pessoa da equipe pode modificar



qualquer código para completar seu trabalho, sem ter que esperar pela autorização do programador original daquela parte do código [WILLIAMS, 2003]. Assim, todos se responsabilizam pela qualidade do código desenvolvido [RAMACHANDRAN & SHUKLA, 2002].

9. **Integração Contínua e Freqüente** – O trabalho de codificação é dividido em pequenas partes, com duração de não mais que um dia, e integrado ao código coletivo a cada conclusão de tarefa. Quanto mais integração, menos conflitos ocorrem entre partes do código [RAMACHANDRAN & SHUKLA, 2002].
10. **Ritmo Sustentável** – XP defende que os programadores não se cansem com sobrecarga de trabalho, o que comprovadamente diminui a qualidade dos artefatos produzidos por esses. Esta prática reflete que os desenvolvedores devem trabalhar “duro” e descansar somente quando necessitarem [RAMACHANDRAN & SHUKLA, 2002].
11. **Cliente com os desenvolvedores** – Os clientes estão sempre disponíveis para esclarecer e validar os requisitos durante o processo de implementação. Um verdadeiro cliente deve se sentar com a equipe de desenvolvedores e estar disponível para responder perguntas, resolver questões e estabelecer prioridades [BECK, 2000].
12. **Padrões de Código** – Para facilitar o entendimento do código por todos os membros da equipe, se faz necessário adotar um padrão para escrita de código.

Finalmente, uma última prática importante é: adapte XP quando a metodologia falha. Ou seja, a metodologia deve ser adaptada a cada situação e a cada equipe de desenvolvedores. Às vezes, as equipes que utilizam XP criam uma variante customizada da metodologia por um desses dois motivos: a equipe decide não utilizar todas as práticas de XP e, após esta decisão, a equipe diminui a dependência entre as doze práticas com a inserção de outras que tenham o mesmo efeito [RAMACHANDRAN & SHUKLA, 2002].

Durante o processo de desenvolvimento e evolução da agenda *No Risk Planning* foram utilizadas de forma experimental algumas regras de XP, em paralelo com as técnicas de *Engenharia de Web* e Engenharia de software tradicionais, objetivando analisar que benefícios a utilização de XP trouxe para a melhoria de qualidade do software e do seu processo de desenvolvimento.

No desenvolvimento da agenda *No Risk Planning*, a equipe era formada somente por três pessoas: dois desenvolvedores e um cliente. Com apenas duas pessoas como desenvolvedores,

não ocorre os benefícios que trazem a rotatividade e a troca de pares que ocorre em equipes maiores e também, se houvesse alguma discussão que gerasse impasse, não haveriam outras duplas para ajudar na resolução do mesmo [WOOD & KLEB, 2003]. No caso do desenvolvimento da agenda, quando ocorria algum impasse, uma discussão era conduzida, de forma que a melhor opção fosse argumentada e aceita por ambos.

Com relação às práticas de XP, a Tabela 3.3 mostra quais práticas foram ou não adotadas durante o processo de evolução da agenda, e a respectiva justificativa para cada um dos casos.

**Tabela 3.3:** Tabela de Práticas e seu uso (ou não) no desenvolvimento da Agenda *No Risk Planning*

Prática	Status	Justificativa
Jogo do Planejamento	Adotado	Os cartões de história com os novos requisitos do sistema eram coletados a cada reunião de equipe que acontecia semanalmente com a presença da orientadora (cliente).
Pequenas atualizações	Não adotada	Pequenas atualizações eram adotadas, mas não era atribuído o nome de uma nova versão a cada atualização.
Metáfora	Não adotada	Esta prática não foi adotada por acharmos que não caberia em nosso contexto.
Design Simples	Adotada	O <i>design</i> simples iniciado por Moura [MOURA, 2001], foi mantido nas evoluções posteriores ao seu trabalho.
Testes	Não adotada	Testes foram feitos, mas não como estabelece XP, ou seja, os testes foram realizados após a implementação de cada atualização.
Refatoramento	Adotada parcialmente	O refatoramento era realizado somente quando tínhamos que acoplar um novo módulo ao sistema para fazer sua integração.
Programação em pares	Adotada parcialmente	Esta prática só não era adotada quando aparecia alguma mudança de última hora. Quando isso acontecia, as mudanças eram feitas somente por uma das pessoas, dependendo de quem recebesse a solicitação primeiro.
Código Coletivo	Adotada	Com uma equipe pequena como a nossa, qualquer membro poderia modificar qualquer parte do código a qualquer momento de acordo com as necessidades.
Integração Contínua	Adotada	A integração contínua era feita sempre, porque sempre haviam novos módulos a serem inseridos no sistema e tínhamos que garantir que todo o sistema continuaria a ser executado mesmo após a inserção destes módulos.
Semana de 40 horas	Não adotada	Prática não adotada devido aos outros afazeres na universidade como, provas, estudos, trabalhos e seminários.
Cliente sempre presente	Adotada	O papel de cliente era desempenhado pela orientadora, que estava sempre presente na universidade.
Padrões de Código	Adotada parcialmente	Os padrões de código somente eram adotados quando já existia algum padrão pré-definido. Quando não existia, a programação era feita da maneira mais simples possível.

Quanto às premissas de XP utilizadas e os resultados obtidos através do uso/ não das doze práticas de XP descritas, vale ressaltar as seguintes atividades adotadas durante o processo de evolução da agenda *No Risk Planning*:

- **Desenvolvimento orientado por testes:** Durante todo o processo de implementação da agenda *No Risk Planning* foram realizados testes constantes para assegurar a qualidade do *software* desenvolvido. Testes de funcionalidade e testes com os usuários do sistema foram realizados regularmente. Não pudemos ressaltar grandes vantagens de XP sobre a Engenharia de Software nesse aspecto.
- **Programação am pares:** Toda o processo de evolução da agenda *No Risk Planning* foi realizado utilizando o conceito de programação em pares; dois programadores no mesmo computador na definição do *design*, na codificação e nos testes. Pudemos notar que a utilização desse conceito trouxe grandes benefícios à realização do projeto e ao crescimento do *know-how* da equipe. O número de erros foi reduzido, refletindo na qualidade do *software* resultante, além de que a produtividade (geração de código) aumentou consideravelmente.
- **Cliente com os desenvolvedores:** O papel de “cliente” durante o desenvolvimento do projeto foi desempenhado pela orientadora deste. As diretrizes do projeto eram discutidas semanalmente e a cada reunião eram definidos os objetivos para a semana e obtido o *feedback* em relação ao trabalho realizado, bem como era realizada uma adequação aos requisitos delineados na reunião anterior. Além disso, usuários com os quais foram realizados alguns dos testes, em sua maioria, tinham contato com os desenvolvedores, podendo dar sugestões e críticas de forma a melhorar o sistema. Assim, pôde-se obter uma grande melhora do *software* em relação aos anseios dos usuários.

Através desse experimento, foi possível identificar quais os benefícios trazidos pela utilização de técnicas de XP em conjunto com a utilização de *Engenharia de Web*. As técnicas utilizadas de XP foram aplicadas principalmente durante a implementação e codificação, mas para manter a qualidade do *software*, com o objetivo de evolui-lo, as atividades de planejamento e documentação foram também realizadas.

O uso de parte das práticas de Programação Extrema foi muito importante no processo de desenvolvimento da agenda *No Risk Planning*, pois imprimiu maior agilidade a este processo. Como os requisitos e o público-alvo da aplicação estavam em constante mudança, vimos nas

práticas de XP, uma oportunidade de estar aumentando a velocidade de implementação e, ao mesmo tempo, aprendendo e analisando essa nova forma de produzir *software*.

### **3.3 – Considerações Finais**

Este capítulo apresentou o processo de desenvolvimento da agenda através do uso de Engenharia de *Web* combinada com técnicas de *Programação Extrema*. Uma descrição das duas abordagens foi apresentada, com base nos estudos realizados da literatura, bem como se deu a aplicação do processo estudado na evolução da agenda *No Risk Planning*.

Considerando que o processo da agenda *No Risk Planning* cumpriu as etapas do ciclo de vida de Engenharia de *Web* aplicando-se algumas regras de Programação Extrema, pudemos viabilizar a implementação das novas versões da agenda, atendendo aos novos requisitos solicitados e possibilitando que o processo adotado garantisse qualidades ao produto e que o processo demonstrasse a produtividade da equipe de desenvolvimento.

Dessa forma, a agenda *No Risk Planning* suportada pelo processo adotado, a partir das solicitações de mudanças de visão dos usuários e de inserção de novas funcionalidades à agenda, passou por evoluções, as quais são descritas no próximo capítulo.



# Capítulo 4

---

## **Evolução da Agenda *No Risk Planning***

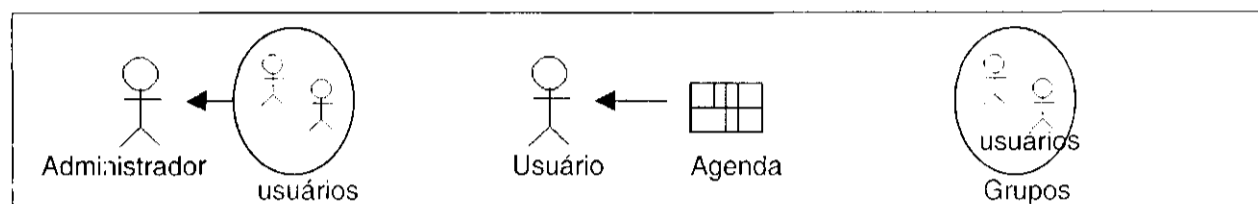
Neste capítulo são descritas as versões da agenda *No Risk Planning* que evoluíram com o emprego do processo de desenvolvimento descrito no Capítulo 3. Considerando-se como ponto de partida a agenda *No Risk Planning* apresentada no Capítulo 2, que havia sido desenvolvida anteriormente, em um projeto de graduação por Moura [MOURA, 2001], diversas solicitações de mudanças de visão dos usuários e de inserção de novas funcionalidades na agenda motivaram a evolução da mesma em mais duas novas versões. Com base nas seis questões de contexto (*awareness*), que resumem as características consideradas mais importantes para se ponderar que tipo de suporte uma ferramenta de *groupware* deve priorizar (Seção 2.5), foi possível se ponderar quais solicitações dos novos usuários da agenda deveriam ser atendidas. E assim, os requisitos que antecederam as mudanças foram estabelecidos em reuniões com a orientadora deste trabalho, que por controlar e centralizar a demanda das solicitações, desempenhou o papel de cliente para o processo de evolução da agenda *No Risk Planning*.

### **4.1 – Agenda *No Risk Planning* – Versão 1.0**

A descrição da agenda *No Risk Planning* com seus usuários e seus modos de interação foi apresentada na Seção 2.4.2 desta dissertação e trata da versão original em português, desenvolvida por Moura [MOURA, 2001]. Esta primeira versão foi desenvolvida com o objetivo de se obter um protótipo funcional da agenda que atendesse aos requisitos básicos de uma agenda compartilhada por grupos de pessoas, baseada na *web*; foi desenvolvida em 3 meses pelo aluno Moura, sob a orientação da mesma orientadora deste trabalho de mestrado.

A versão 1.0 da agenda *No Risk Planning* atende a dois tipos básicos de usuário: 1) o **administrador**, cuja responsabilidade é organizar a estrutura do sistema para a instituição para a qual ele está sendo implantado, bem como criar as contas dos usuários e as categorias dos grupos e 2) o **usuário comum**, que é responsável por administrar sua agenda individualmente, participar de grupos e compartilhar agenda e documentos. Na Figura 4.1 encontram-se esquematizadas as

entidades principais que foram atendidas nesta primeira versão da agenda: o administrador, o usuário, a agenda (espaço compartilhado) e os grupos de usuários.



**Figura 4.1:** Entidades atendidas pela primeira versão da agenda *No Risk Planning*

A implementação da versão 1.0 da agenda *No Risk Planning* contava com 71 arquivos de programa-fonte, 6.615 linhas de código e 41 operações [RIBEIRO, FORTES & FREIRE, 2002]. A base de dados relacional possuía 14 tabelas, conforme esquemas a seguir:

**proj\_admin** (admin\_id, nome\_admin, email\_admin, pagina, caminho, nome\_inst, sigla\_inst, doc, senha)

**proj\_categ\_grupos** (categ\_id, nome)

**proj\_cont\_hora** (conta\_id, horario\_id, descricao, tipo)

**proj\_contas** (conta\_id, departamento\_id, nome, ult\_nome, email, pagina, comentarios, senha)

**proj\_departamentos** (departamento\_id, unidade\_id, nome, sigla, descricao)

**proj\_grupo\_agenda\_temp** (grupo\_id, conta\_id, agendador, dia, inicio, fim, descricao, confirmacao, validade)

**proj\_grupo\_avisos** (grupo\_id, aviso\_id, descricao)

**proj\_grupo\_chat** (grupo\_id, nome, mensagem, hora, tempo)

**proj\_grupo\_doc** (grupo\_id, url, tamanho, descricao)

**proj\_grupo\_membros** (grupo\_id, conta\_id, cargo)

**proj\_grupo\_membros\_temp** (grupo\_id, membro\_id, aleatório)

**proj\_grupos** (grupo\_id, categoria\_id, nome, sigla, cor, descricao)

**proj\_horários** (horario\_id, nome)

**proj\_unidades** (unidade\_id, nome, sigla, descricao)

Na implantação da primeira versão da agenda, foram realizados diversos testes com usuários, que demonstraram interesse a boa aceitação do produto; no entanto, a partir do *feedback* obtido, solicitações de alterações se tornavam muitas e frequentes. Assim, foi adotada

a abordagem evolutiva para o processo de desenvolvimento da agenda, conforme descrita no capítulo anterior, e o desenvolvimento da segunda versão da agenda foi iniciado.

## **4.2 – Agenda *No Risk Planning* – Versão 2.0**

Para o desenvolvimento da segunda versão da agenda *No Risk Planning*, duas etapas foram realizadas paralelamente: a elaboração de um relatório técnico contendo documentação da agenda *No Risk Planning* versão 1.0, e as mudanças para atender às primeiras solicitações estabelecidas. A documentação foi importante para aquisição de conhecimento do estado do código implementado na agenda, em sua primeira versão. Assim, o ciclo de vida de interação da agenda e todos os diagramas UML (*Unified Markup Language*) foram produzidos, refletindo uma engenharia reversa realizada sobre a agenda *No Risk Planning* versão 1.0.

Nas subsecções a seguir, cada uma destas duas etapas é descrita.

### **4.2.1 – Relatório Técnico**

Durante o processo de desenvolvimento da agenda *No Risk Planning* versão 2.0, as seguintes atividades foram realizadas: especificação de requisitos, a análise do sistema, o projeto navegacional do sistema, a realização de testes de unidade e a aceitação. Ainda nesta fase foi realizado um estudo sobre *Engenharia de Web* [ATHULA & MURUGESAN 2001] de forma a auxiliar na aplicação das atividades consideradas importantes para a evolução de uma aplicação *Web* e foi possível se detectar os fatores essenciais ao desenvolvimento de aplicativos *web*: escalabilidade, mecanismos de atualização simples, modelo navegacional consistente, capacidade de integração de componentes e outros.

Como a agenda *No Risk Planning*, apresentada no Capítulo 2, havia sido desenvolvida anteriormente, em um projeto de graduação por Moura [MOURA, 2001], com o objetivo de se obter um protótipo funcional da agenda que atendesse aos requisitos básicos de uma agenda compartilhada por grupos de pessoas, baseada na *web*, somente o código-fonte dos programas e o modelo de dados estavam disponibilizados como documentação da agenda. Assim, foi realizado um estudo do sistema e foi efetuada a engenharia reversa, seguindo o método Fusion-RE/I (*Fusion – Reverse Engineering/ Interface*) [COSTA 1997]. Um relatório técnico de documentação do sistema foi elaborado [RIBEIRO, FORTES & FREIRE, 2002] contendo: quadro de operações, ciclo de vida, uma abstração de um diagrama de classes, diagrama de casos



de uso, diagrama de atividades, diagrama de máquinas de estado, diagrama de colaboração e modelo de dados.

#### 4.2.2 – Alterações na Agenda

O processo de mudanças na agenda teve início com a solicitação de mudança nos horários para se marcar compromissos na agenda. A versão 1.0 da agenda *No Risk Planning* lidava com marcação de meia em meia hora; foi alterado na versão 2.0 para que a agenda trabalhasse com a marcação de compromissos, de *hora em hora*. Para que esta mudança fosse efetivada, foi necessário, além de mudar alguns programas (códigos-fonte) relacionados com a produção das páginas apresentando os calendários semanais, alterar a tabela de horários.

A segunda solicitação de mudança, atendida na versão 2.0 da agenda, foi no sentido de *universalização* da mesma. Assim, a agenda foi traduzida integralmente para o inglês.

Além disso, tornou-se necessária a criação de *novos tipos de compromissos*, além dos três já existentes (Fixo, Importante e Normal). Foram então inseridos os tipos: Lazer e Acadêmico. Como a agenda já estava em inglês, os tipos de compromissos passaram a ser: *Fixed, Important, Common, Leisure e Academic*. Para que os tipos de compromissos se tornassem mais intuitivos, houve também mudança de cores. O esquema anterior de cores não correspondia às funções que os compromissos de usuário queriam realmente dizer, então foram adotadas cores que se relacionavam mais com os tipos de compromissos adotados, como: vermelho para *Important*, verde para *Leisure*, amarelo para *Common*, azul para *Academic* e cinza para *Fixed*.

Depois de traduzida, a agenda sofreu novas modificações nos seus programas para que ela pudesse ser inserida na CoTeia (site <http://coweb.icmc.usp.br/norisk/>). E para que a agenda ficasse realmente *ligada à CoTeia*, o usuário passou a ter a possibilidade de relacionar uma URL a compromissos, permitindo que a agenda seja visualizada como um documento interativo, e informações relevantes sobre um dado compromisso possam ser acessadas com um clique nesse compromisso diretamente da agenda.

Com o direcionamento da agenda para o uso na área acadêmica, foi observado que, se algum professor ou aluno fosse participar de algum congresso ou evento que fugisse da sua rotina, não haveria a possibilidade de se inscrever este (eventual) compromisso na agenda. Para resolver esta situação, foi criado o *conceito de compromisso eventual*, como aquele que acontece esporadicamente e tem tempo definido para iniciar e terminar. Os compromissos eventuais podem se sobrepor a quaisquer outros compromissos previamente agendados, como regulares,

pelo usuário. Na agenda semanal, os compromissos eventuais são mostrados por extenso logo abaixo da agenda propriamente dita. Com a criação deste novo tipo de compromisso, foram inseridas as seguintes operações para a sua manipulação:

- Inserção de um Compromisso Eventual, onde o usuário tem a possibilidade de inserir um novo compromisso e a agenda questiona se ele quer realmente sobrepor o compromisso eventual sobre todos os outros compromissos regulares previamente marcados na agenda;
- Remoção de um Compromisso Eventual, que o usuário pode remover quaisquer compromissos eventuais previamente marcados.

Outra solicitação, sugerida pelos usuários, foi a possibilidade de *visualização mensal* da agenda. Foi desenvolvida uma nova visão da agenda, desta vez mensal, com todos os compromissos eventuais que o usuário tenha marcado. Com esta nova possibilidade, o usuário da agenda pode se programar, de acordo com os seus compromissos eventuais já agendados.

E a última solicitação atendida nesta versão 2.0 da agenda, foi a implementação da *visualização semestral* da agenda. Nesta visão, são apresentadas as seis agendas mensais que compõem um semestre.

Finalmente, realizadas todas estas mudanças na agenda, tornou-se necessária a atualização do *Help* para que realmente pudesse cumprir a sua função.

Estas foram as principais mudanças ocorridas da primeira versão da agenda para a segunda versão. Essas mudanças proporcionaram um crescente interesse por parte dos usuários. As opções oferecidas pela versão 2.0 da agenda *No Risk Planning* se aproximavam mais do que os usuários esperavam encontrar, além de que as mudanças na interface desta segunda versão tornaram visíveis os diferentes modos de visualização, como mostra a Figura 4.2.

Na segunda versão, a implementação contava com 81 arquivos de programas, 8.359 linhas de código e 45 operações. O banco de dados consistia de 15 tabelas, tendo sido criada uma nova tabela para o armazenamento de compromissos *eventuais*, que eram vinculados à data e hora de início e fim, e não ao dia da semana e hora, como os compromissos fixos.

O objetivo principal a ser alcançado com o desenvolvimento da segunda versão foi o de levantar requisitos para melhoria da funcionalidade do sistema, e uma melhor adequação às necessidades dos usuários.

Considerando que o processo da agenda *No Risk Planning* cumpriu as etapas do ciclo de vida de Engenharia de *Web* aplicando-se algumas regras de Programação Extrema, pudemos viabilizar a implementação das novas versões da agenda, atendendo aos novos requisitos solicitados e

possibilitando que o processo adotado garantisse qualidades ao produto e que o processo demonstrasse a produtividade da equipe de desenvolvimento.

Dessa forma, a agenda *No Risk Planning* suportada pelo processo adotado, a partir das solicitações de mudanças de visão dos usuários e de inserção de novas funcionalidades à agenda, passou por evoluções, as quais são descritas no próximo capítulo.

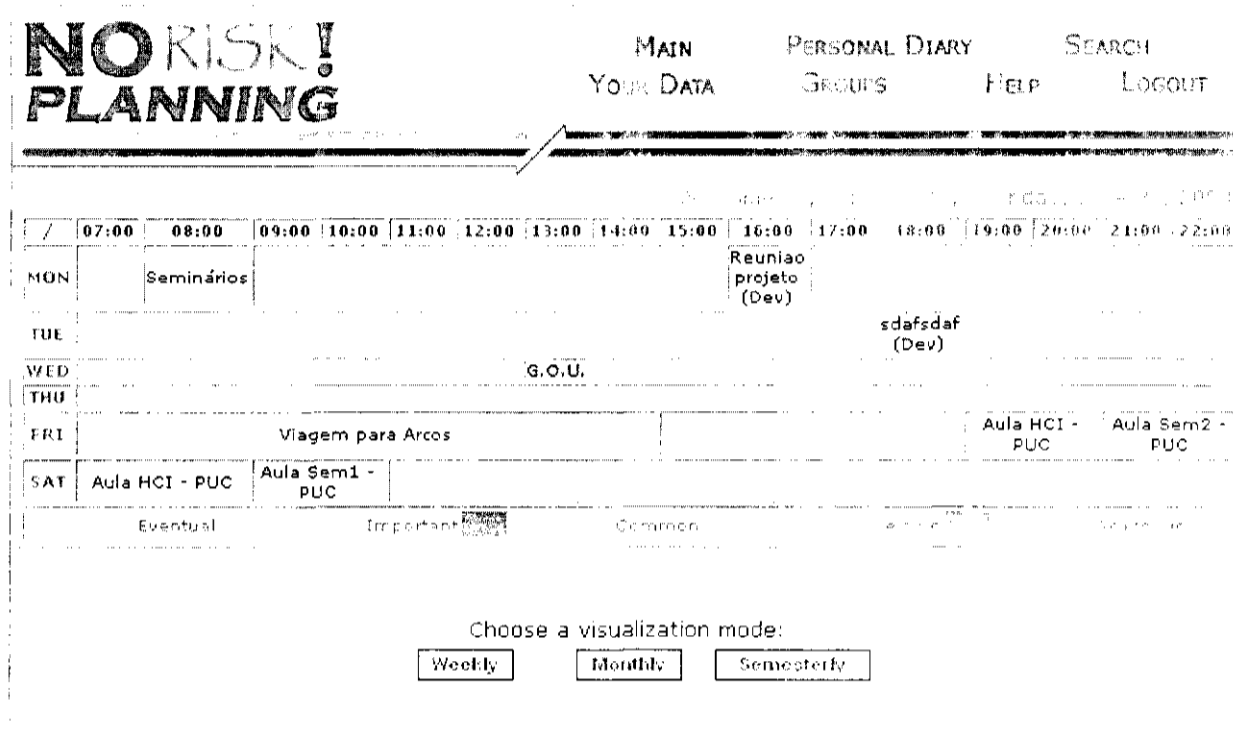


Figura 4.2: Página principal da agenda (versão 2.0) para o usuário convencional

### 4.3 – Agenda *No Risk Planning* – Versão 3.0

Com mais funcionalidades, a agenda *No Risk Planning* – versão 2.0 atendia aos requisitos básicos de um *groupware* do tipo agenda para *web*. A boa aceitação da mesma estimulou estudos sobre como adequar o conceito de agenda implementado a outros domínios de informação, além de dar prosseguimento aos testes com usuários e levantamento de requisitos para sua evolução.

Assim, durante o desenvolvimento da terceira versão da agenda, o destaque foi a criação de três novos tipos de usuário: disciplina, administrador de disciplina e *guest*. Além destes novos usuários, outras mudanças foram também realizadas na estrutura e na interface da agenda *No Risk Planning*.

#### 4.3.1 – Usuário: Administrador de Disciplinas

A idéia de se criar agendas para disciplinas surgiu como forma de estimular o uso da *No Risk Planning* pelos alunos e professores do ICMC-USP.

O administrador de disciplinas é responsável pelo gerenciamento dos usuários do tipo disciplina. Esse administrador de disciplina possui um perfil semelhante ao de um coordenador de cursos, e por isso, diferentemente do administrador de usuários comuns, também tem a permissão de manipular as agendas dos usuários tipo disciplina. As opções disponíveis para o administrador de disciplinas são as seguintes:

- *Main*: é a página inicial do administrador de disciplinas que exibe as estatísticas da agenda com os seguintes dados: número de professores, número de cursos e número de usuários temporários. Usuários temporários são aqueles que estão cadastrados como usuários na agenda, mas não a usam como os usuários convencionais.
- *Courses & Professors*: esta opção é usada para gerenciar os cursos oferecidos e os professores cadastrados. Assim, são disponibilizadas as operações para gerenciamento de contas de professores e de disciplinas, conforme descritas na Tabela 4.1.
- *Password*: opção para trocar a senha do administrador - ele informa a senha atual e depois informa e confirma a nova senha;
- *Help*: exibe uma tela de ajuda para o administrador;
- *Commitment Management*: constitui no gerenciamento de compromissos e funciona da seguinte maneira: o administrador seleciona o código de um determinado curso como mostra a Figura 4.3 e, depois pode escolher entre: inserir um novo compromisso, modificar um compromisso já existente e remover um compromisso da disciplina;
- *Search*: opção de busca de usuários cadastrados na agenda e funciona da mesma maneira que a busca detalhada na seção 2.4.2.2;
- *Logout*: sai do sistema.

**Tabela 4.1:** Funções de gerenciamento de contas de professores e de disciplinas

<b>Função</b>	<b>Descrição</b>
<i>Create New Course</i>	Cria um novo curso e requer os seguintes dados: nome do curso, grupo, professor responsável, descrição, código do curso e comentários a respeito.
<i>Modify Course</i>	Seleciona o <i>login</i> de um determinado curso e modifica seus dados básicos.
<i>Remove Course</i>	Seleciona o <i>login</i> de um determinado curso e apaga seus dados da tabela referente a disciplinas.
<i>Create New Professor</i>	Cria um usuário para um novo professor que tenha sido contratado pelo ICMC e cadastra os seguintes dados: identificação do professor na agenda, seu primeiro e último nome.
<i>Modify Professor</i>	Seleciona o professor pelo nome e modifica seus dados básicos que foram citados na função anterior.
<i>Remove Professor</i>	Seleciona o professor e remove os seus dados da agenda <i>No Risk Planning</i> .

É importante mencionar de que para que o usuário administrador de disciplinas iniciasse as suas atividades no ambiente acadêmico esperado, foi necessária a inserção de todas as disciplinas oferecidas pela Graduação e Pós-Graduação do ICMC-USP, como carga na base de dados da agenda.

#### **4.3.2 – Usuário: Disciplina**

Nesta terceira versão foi realizada a adaptação para os requisitos de uma agenda para Disciplinas. O objetivo foi fazer com que a agenda atendesse a requisitos tais como agendamento de aulas teóricas, aulas práticas, provas, trabalhos, dentre outros. Tais requisitos se enquadraram facilmente aos conceitos adotados, de compromissos periódicos semanais e compromissos eventuais, e à estrutura de dados utilizada para tais compromissos.

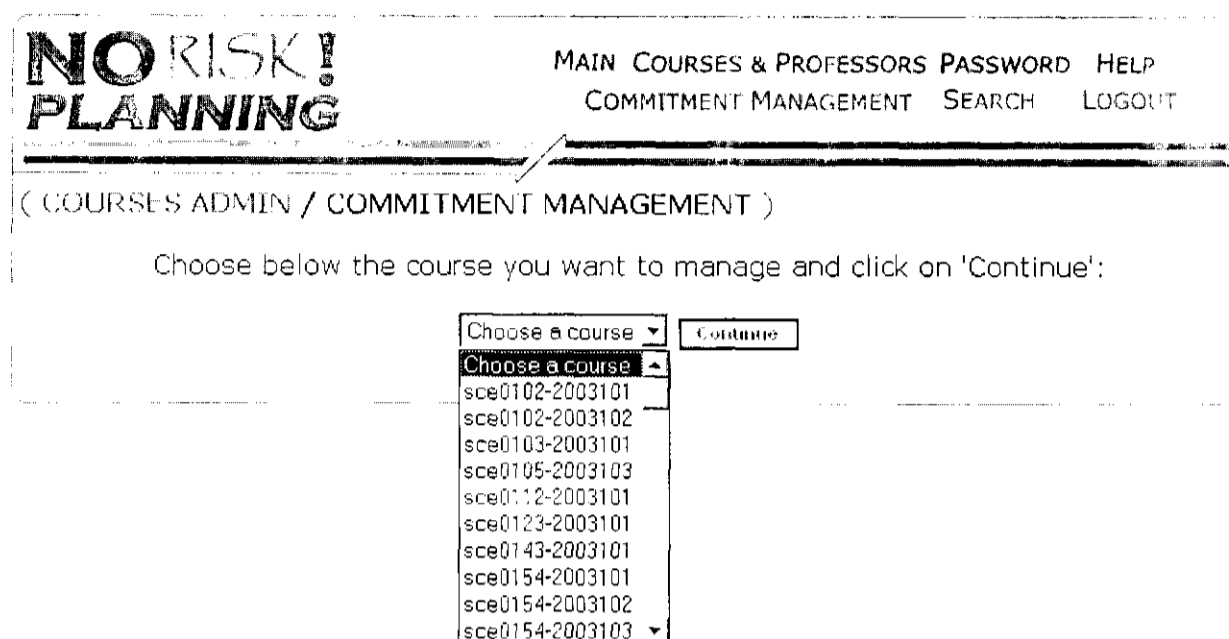


Figura 4.3: Página de gerenciamento de compromissos de disciplinas

A partir daí, foi criado um novo tipo de usuário, denominado “disciplina”, também possuidor de uma agenda; foi desenvolvida uma nova interface para esse tipo de usuário, e realizadas as devidas adaptações para o contexto em que ele se insere.

Para se ter acesso às agendas de compromissos das disciplinas oferecidas, três maneiras podem ser utilizadas: (a) o professor acessa a agenda da disciplina através de sua agenda pessoal ou através do *login* e senha da disciplina, (b) o monitor acessa a agenda com o *login* da disciplina e a senha da disciplina e (c) os alunos acessam a agenda da disciplina através da sua agenda pessoal. Neste último caso, o acesso é somente de leitura, pois os alunos não possuem direito para inserir/remover compromissos referentes às disciplinas que cursam.

O *login* de cada disciplina é dado pela seguinte concatenação das seguintes informações: o código da disciplina no ICMC + ano que foi ou que é ministrada + número da turma da disciplina. Por exemplo, uma disciplina que tenha o código SCE-5787 que foi ministrada em 2002 e tinha duas turmas terá os seguintes *logins*: SCE5787-200301 e SCE5787-200302.

Como as disciplinas eram bem diferentes dos usuários convencionais, houve a necessidade de se criar novos tipos de compromissos na agenda, a saber:

- *Eventual*: indica o compromisso eventual da disciplina como, por exemplo, uma aula de reposição ou um dia de prova;
- *Class*: indica os dias e os horários de aulas da disciplina;
- *Student Assistance*: indica os dias e os horários das monitorias;

- *Laboratory*: indica os dias e os horários com aulas de laboratório, caso a disciplina necessite deste tipo de recurso.

Os compromissos de Aulas, Monitoria e Laboratório poderão ser sobrepostos por compromissos eventuais quando não houver aula, monitoria ou laboratório devido a algum motivo em especial como, a doença de um professor ou um congresso estudantil, quando as aulas são substituídas por palestras e mini-cursos.

As agendas de disciplinas podem ser acessadas por professores, monitores e alunos, mas somente os professores terão opção de modificar dados nas agendas das disciplinas e os monitores e alunos poderão somente visualizar o seu conteúdo. Assim, um usuário tipo “disciplina” tem um proprietário do tipo “professor”, e as agendas desses dois usuários ficam sincronizadas no tocante a compromissos relacionados a atividades docentes, e o restante dos compromissos do professor continuam independentes das agendas das disciplinas pertencentes a ele.

Como as disciplinas não são usuários convencionais, foi criado um novo conjunto de opções (menu) com os seguintes itens, para manuseio das disciplinas:

- *Main*: exibe a agenda semanal da disciplina corrente. Desta agenda, os alunos, monitores e professores poderão visualizar a agenda mensal e a agenda semestral da disciplina atual.
- *Course Diary*: onde os professores poderão inserir, remover e modificar os compromissos fixos e eventuais da agenda da disciplina. Há também a opção de se “limpar” a agenda (remover todos os compromissos marcados) e a opção de salvar a agenda em um arquivo .JPG para disponibilizá-la em algum outro local.
- *Basic Data*: modifica os dados básicos da disciplina; somente os professores têm a opção de realizar esta modificação. Os dados a serem modificados são, como mostra a Figura 4.4: descrição da disciplina, a sigla da disciplina, a página principal do curso na Internet e algum possível comentário a respeito da disciplina ou sua ementa básica.
- *Password*: tela para troca de senha da disciplina onde o professor entra com a senha atual e depois entra e confirma a nova senha.
- *Help*: exibe uma tela de ajuda ao usuário que estiver acessando a agenda da disciplina.
- *Logout*: sai da agenda da disciplina e retorna à agenda pessoal do visitante.

<b>NO RISK! PLANNING</b>	MAIN	COURSE DIARY	BASIC DATA
	PASSWORD	HELP	LOGOUT

---

( SCL0211 / COURSE'S DATA )

Description	Seminários Avançados em Hipermedia
Course's acronym	SemHip
Course's home page	http://coweb.icmc.usp.br/coweb/mostra.php
Comments (Optional) with 200 characters or less	Esta disciplina trata dos temas atuais na área de Hipermedia. <input type="checkbox"/>
	<input type="checkbox"/>

**Figura 4.4:** Página para modificação de dados básicos da disciplina

A agenda de disciplinas do ICMC-USP foi implementada para ajudar os alunos e professores a saberem quais as datas importantes a serem registradas e armazenadas, e quais os dias de monitoria e aulas de laboratório. Os usuários do tipo “disciplina” são armazenados em uma outra tabela separada dos usuários comuns, dadas as particularidades desse tipo de usuário que o usuário comum de *No Risk Planning* não possui.

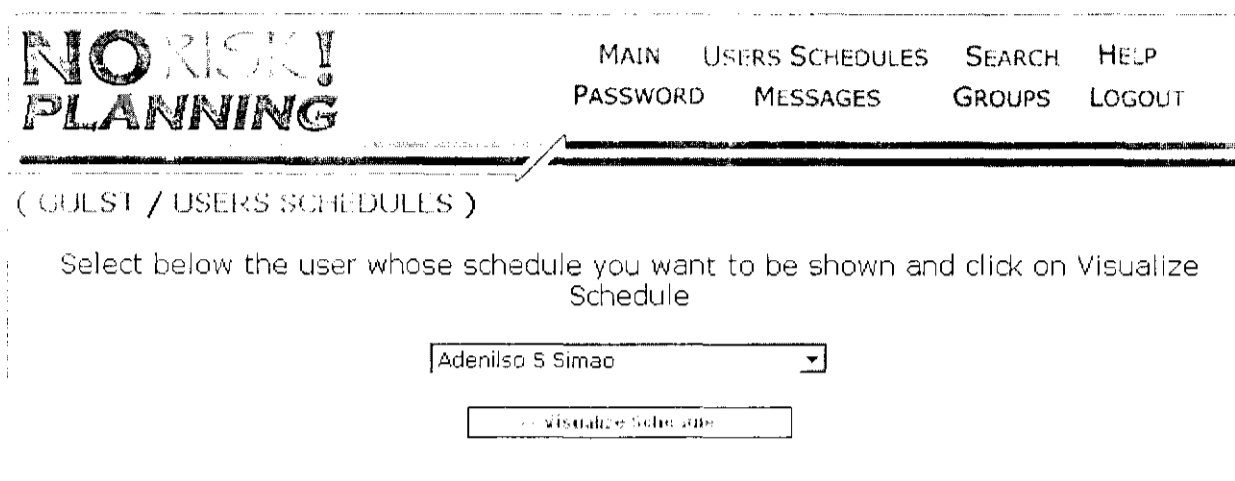
#### 4.3.3 – Usuário: *Guest*

Foi criado um novo tipo de usuário “*Guest*”, que não possui uma agenda particular, mas tem permissão de visualizar as agendas de todos os usuários, assim como criar “grupos virtuais”, podendo visualizar agendas agrupadas, e pode também enviar mensagens aos usuários do sistema. Esse usuário se destina a um usuário com um perfil próximo a de um funcionário de secretaria, que não tem controle sobre as contas, como um administrador, mas pode acessar os dados de todos os usuários.

O usuário *Guest* não possui agenda pessoal, então não há compromissos a serem descritos. O menu de opções ao usuário *Guest* é composto dos seguintes itens:



- *Main*: mostra as estatísticas da agenda *No Risk Planning* para o usuário *Guest* e possui os seguintes dados: número de usuários da agenda, número de tipos de grupos que existem na agenda e número de grupos efetivamente criados pelo *Guest*.
- *Users Schedules*: nesta opção, aparece uma lista com todos os usuários cadastrados na agenda e o *Guest* pode selecionar qualquer usuário desta lista e visualizar a sua agenda pessoal, como mostra a Figura 4.5.
- *Search*: nesta seção, pode-se procurar por qualquer usuário da agenda e funciona exatamente da mesma forma que a opção “Procurar” que foi descrito na seção 2.4.2.2 e a única diferença é que esta opção está toda em inglês.
- *Help*: exibe uma tela de ajuda à pessoa que estiver utilizando a agenda como usuário *Guest*.
- *Password*: opção de troca de senha onde o usuário deve informar a senha atual e depois, informar e confirmar a nova senha.
- *Messages*: nesta seção, o usuário *Guest* pode mandar *e-mails* a quaisquer membros da agenda de modo individual ou então ele pode mandar *e-mail* para um grupo montado por ele, como mostra a Figura 4.6. Esta opção costuma ser usada para marcar e desmarcar reuniões ou convocar alguma pessoa para comparecer à Secretaria.
- *Groups*: seção que cria e gerencia todos os grupos criados pelo *Guest*. Segue a descrição das duas principais atividades dentro desta seção:
  - o Quando a opção de criar um novo grupo é escolhida, uma página é exibida e são pedidos os seguintes dados: nome do grupo, categoria do grupo, número de membros do grupo e uma breve descrição do mesmo. Depois de preenchidos todos os campos, os membros do grupo são escolhidos através da seleção direta na lista de todos os usuários da agenda e a criação do grupo é concluída;
  - o Quando o *Guest* acessa os dados referentes a determinado grupo, uma página é exibida, que possui as opções descritas na Tabela 4.2.
- *Logout*: sai da agenda *No Risk Planning*.



**Figura 4.5:** Página para seleção de usuário para visualização de sua agenda pessoal

**Tabela 4.2:** Opções de manipulação e gerenciamento de grupos.

Opção	Função
<i>Group's Information</i>	Exibe todos os dados pessoais cadastrados por todos os membros do grupo corrente.
<i>Diary of All Members</i>	Exibe a agenda pessoal de todos os membros do grupo corrente.
<i>Modify Basic Data</i>	Modifica os dados básicos (que são: o nome, a descrição e o tipo de grupo) referentes ao grupo atual.
<i>Insert a New Member</i>	Insere um novo membro no grupo corrente.
<i>Remove a Member</i>	Remove algum membro do grupo corrente.
<i>Remove this Group</i>	Apaga todo o grupo atual.

Com a criação do usuário *Guest* foi necessário criar uma nova tabela para armazenar os grupos desse usuário de forma independente dos grupos criados por usuários comuns com o objetivo de compartilhar agendas e documentos e marcar compromissos comuns, uma vez que o objetivo dos grupos criados pelo *guest* é somente visualizar agendas agrupadas.

#### 4.3.4 – Outras Alterações na Agenda

Além da inserção dos três novos tipos de usuário citados, várias outras mudanças foram realizadas na agenda e todas elas foram incorporadas por todos os outros tipos de usuários da agenda.

NO RISK!  
PLANNING

[MAIN](#)   [USERS](#)   [SCHEDULES](#)   [SEARCH](#)   [HELP](#)  
[PASSWORD](#)   [MESSAGES](#)   [GROUPS](#)   [LOGOUT](#)

---

( GUEST / MESSAGES )

At this section you can send messages to one or more users.  
 To send a message to one user, mark the option "user" and choose the user you want to receive the message.  
 To send a message to a group of users, mark the option "group" and choose the group of users that will receive the message. After the option is chosen, fill in all of the fields and click on "Send message".

To:  User:    
        Group:

Subject:

Message:

**Figura 4.6:** Página onde o *Guest* pode mandar *e-mails* para quaisquer membros da agenda

Com a criação do usuário disciplina, houve a necessidade de se disponibilizar aos outros usuários, o acesso para visualização das agendas das disciplinas que ele cursa, se for estudante, ou que ele ministra, se for professor.

Para poder se diferenciar um usuário comum de um usuário professor foi inserido novo atributo na tabela de usuários com a definição do tipo de usuário que está acessando o sistema. Esta diferenciação de usuários também será importante para implementações futuras de novas funcionalidades da agenda.

Como sugestão dos usuários cadastrados na agenda, foi adicionada à mesma uma barra de localização superior onde é mostrado em qual página da aplicação o usuário está navegando. Esta barra ajuda o usuário a se localizar nas opções da agenda e fornece o contexto em que ele está inserido. Com a possibilidade de o usuário acessar outra agenda (da disciplina) a partir de sua agenda pessoal, foi possível que alguns usuários se sentissem perdidos, sem saber onde estão, e a *barra de localização* ajudou de forma decisiva neste quesito.

Com a criação do usuário disciplinas, outra mudança que teve que ser implementada foi a *permissão* para que mais de uma pessoa pudesse modificar determinada agenda. Isso se tornou necessário porque a agenda de uma disciplina pode ser modificada tanto por um professor, quando este acessa à partir de sua agenda, quanto por um monitor, que acessará a disciplina através do *login* e senha específicos.

Outra possibilidade inserida na agenda para o usuário foi a *visualização de uma imagem* de sua agenda pessoal semanal. Neste caso, o usuário tem a opção de escolher quais tipos de compromissos estarão visíveis a outras pessoas e quais não estarão. Caso o usuário queira disponibilizar a imagem da agenda para outras pessoas, há ainda a opção de gerar um arquivo da agenda com a extensão .JPG. Para que ele possa salvar esta imagem, antes ele terá que configurar quais tipos de compromissos ele deixará disponível para visualização e salvar estas configurações.

Na mesma página em que o usuário tem as opções de trabalhar com imagens de sua agenda pessoal, há uma caixa de texto com o código-fonte em *Hypertext Markup Language* (HTML) caso o usuário queira criar uma página de Internet com a imagem de sua agenda pessoal. Para que pudéssemos transformar a agenda em um arquivo, foi estudada a biblioteca GD do PHP. Além disso, teve que ser instalada, acarretando a reinstalação completa do PHP.

Atendendo a mais sugestões de usuários, foi inserida a *possibilidade de se modificar* dados de um compromisso já marcado na agenda. Antes desta mudança, o usuário tinha que desmarcar o compromisso e depois inseri-lo novamente com os novos dados. Para evitar este tipo de transtorno, uma opção de modificação de compromissos fixos e eventuais foi adicionada com opção ao usuário.

Com o intuito de se criar um histórico dos compromissos de usuário, foi inserida uma *barra de navegação semanal* onde o usuário pode conferir seus compromissos nas semanas anteriores e posteriores à semana atual. Esta barra de navegação é composta por três botões que realizam as seguintes funções: '<<' acessa a semana anterior à que está sendo exibida no momento; '*Current Week*' que volta à semana atual independente de qual semana o usuário está visualizando no momento e '>>' para que o usuário acesse a semana posterior à semana que está sendo exibida no momento. Esta barra de navegação semanal, bem como outras mudanças que foram inseridas na agenda, estão mostradas na Figura 4.7.

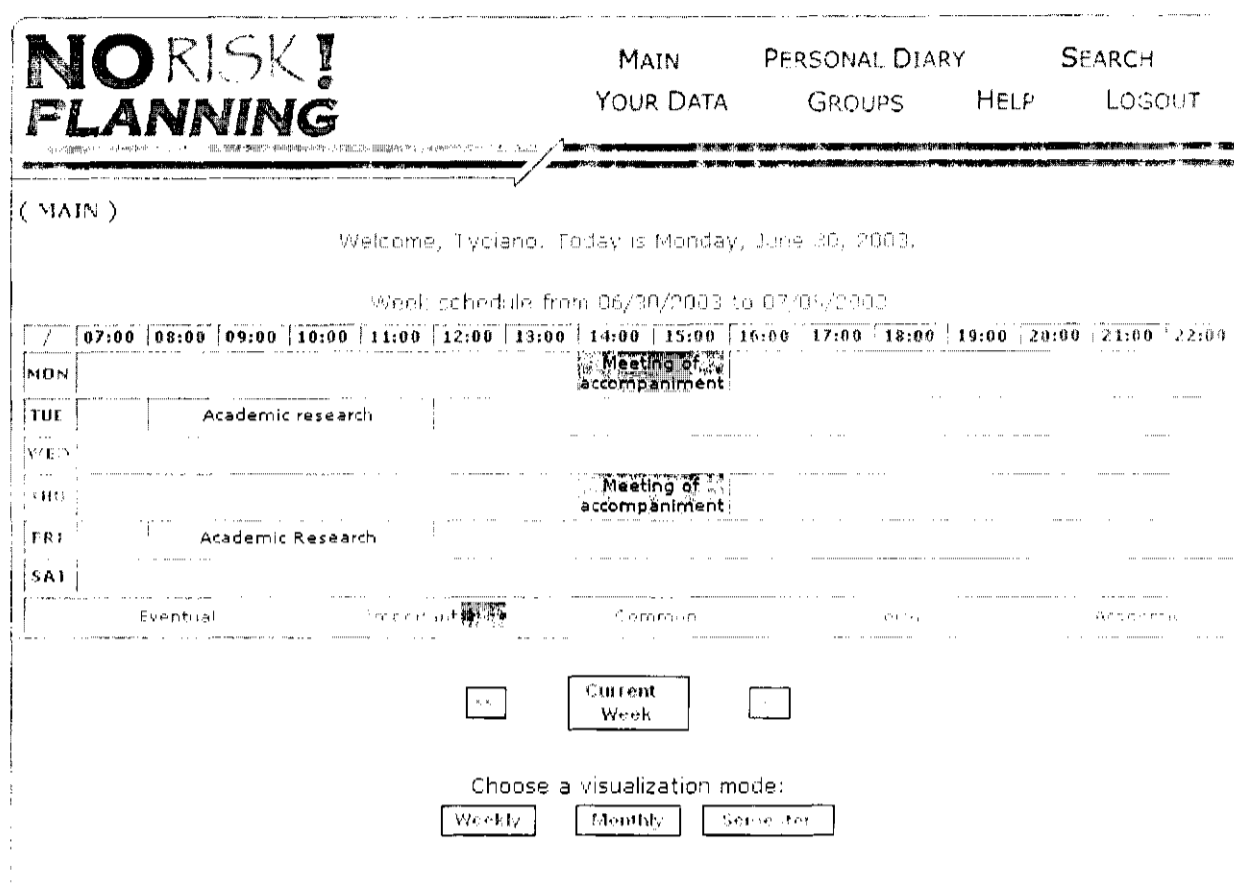


Figura 4.7: Página principal da versão 3.0 da agenda *No Risk Planning*

Com a possibilidade de navegação através das semanas anteriores e posteriores à atual, foi necessário acrescentar na página principal da agenda, a data inicial e a data final da semana corrente para que a chance do usuário se confundir nesta navegação fosse reduzida.

O acesso do usuário à agenda era realizado, até então, através do reconhecimento simples da senha cadastrada por ele. Por se mostrar um procedimento tanto quanto inseguro, passamos a controlar a entrada de usuários no sistema através do uso de *senhas criptografadas*.

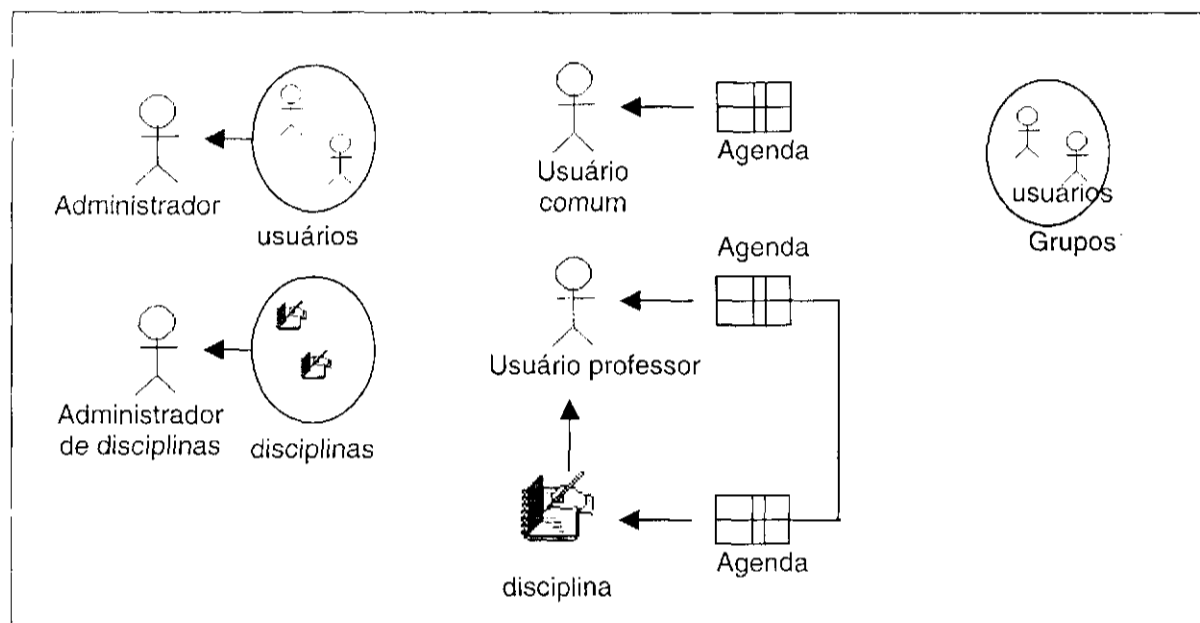
Os compromissos eventuais também passaram a poder receber ligações (*links*) para endereços de Internet, assim como os compromissos fixos.

Nesta terceira versão da agenda, a implementação contava com 191 arquivos de programas, 23.247 linhas de código e 144 operações. O banco de dados consistia de 18 tabelas.

A tabela **proj\_cont\_hora** foi substituída pela tabela **proj\_compromissos**, que delimitava compromissos por início e fim, e não um registro por horário. Com essa substituição, o número de registros dessa tabela caiu de 3.600 para 180 registros. Também foram acrescentadas as tabelas **proj\_disciplinas** para o armazenamento das disciplinas cadastradas, e as tabelas

**proj\_grupos\_guest** e **proj\_grupo\_membros\_guest**, para o armazenamento dos grupos criados pelo usuário *guest*.

Na Figura 4.8 encontram-se esquematizadas as entidades principais que foram atendidas nesta terceira versão da agenda: o administrador-de-usuários, o usuário-comum, a agenda (espaço compartilhado), os grupos de usuários, o administrador-de-disciplinas, o usuário-professor e a disciplina.



**Figura 4.8:** Entidades atendidas pela terceira versão da agenda *No Risk Planning*

Durante a implementação da terceira versão também começou a ser realizado o controle de versões dos códigos-fonte utilizando a ferramenta *VersionWeb* [SOARES, FORTES & MOREIRA, 2000][SOARES & Fortes, 2001], que é uma ferramenta baseada no CVS para controle de versões na *Web*.

#### **4.4 – Implementações Futuras – Agenda *No Risk Planning* – Versão 4.0**

O processo de desenvolvimento da agenda *No Risk Planning* proporcionou que a mesma evoluísse com evidências de melhorias no processo e no produto. Sobretudo, o processo não se encerrou tendo vários trabalhos sendo realizados, dentre os quais podemos destacar: a criação de

um novo tipo de usuário (usuário sala-de-aula) e a preparação de documentos com base nas informações da agenda para XML (*EXtensible Markup Language*).

A criação do usuário sala-de-aula visa atender a demanda das Secretarias e Seções do ICMC-USP, propiciando a manutenção de uma agenda compartilhada em que todos terão acesso aos horários livres e ocupados de todas as salas-de-aula.

Além disso, a integração da agenda *No Risk Planning* com a Intranet do ICMC-USP também será efetivada. O controle de quem poderá inserir/remover compromissos de determinada sala-de-aula será controlado pelo *login* do usuário na Intranet do ICMC-USP. Diferentes tipos de compromissos podem ser marcados, dependendo de qual sala-de-aula estiver sendo consultada (se uma sala de aula, uma sala de reuniões, auditório, etc.).

Outra etapa no processo para evolução da agenda *No Risk Planning* é a passagem da sua visualização de PHP puro para PHP junto com XML, para facilitar o processo de buscas e integrações futuras com outros aplicativos que utilizam XML como linguagem de visualização.

#### **4.5 – Considerações Finais**

Este capítulo apresentou as versões da agenda *No Risk Planning* que evoluíram com o emprego do processo descrito no Capítulo 3.

A versão 1.0 da agenda foi considerada ponto de partida, e já havia sido desenvolvida anteriormente [MOURA, 2001].

A versão 2.0 atendeu prioritariamente as solicitações de (a) marcação de compromissos, de hora em hora, (b) universalização da agenda, (c) inserção de novos tipos de compromissos, (d) ligação com CoTeia (e) criado o conceito de compromisso eventual, (f) possibilidade de visualização mensal e semestral da agenda e, (g) atualização do Help. Vale ressaltar que a documentação da versão 1.0 foi também produzida.

A versão 3.0 foi desenvolvida para adequar o conceito da agenda para um novo domínio de informação, o de agendamento de compromissos em disciplinas. Assim, destacou-se a criação de três novos tipos de usuário: disciplina, administrador de disciplina e *guest*. Além disso, as seguintes solicitações foram atendidas: (a) acesso para visualização das agendas das disciplinas, (b) a barra de localização, (c) permissão para que mais de uma pessoa pudesse modificar determinada agenda, (d) a visualização de uma imagem da agenda pessoal semanal (e) possibilidade de se modificar dados de um compromisso já marcado na agenda, (f) inserida uma barra de navegação semanal, e (g) uso de senhas criptografadas.

Os requisitos que antecederam as mudanças foram estabelecidos em reuniões com a orientadora deste trabalho, que gerenciava as solicitações, desempenhando o papel de cliente para o processo de evolução da agenda *No Risk Planning*.

Finalmente, observou-se que o processo de desenvolvimento das versões da agenda *No Risk Planning* proporcionou que a mesma evoluísse, demonstrando melhorias no produto e no processo, e permitindo a continuidade do processo, prevendo novas solicitações, como a criação de um novo tipo de usuário “sala-de-aula” e a preparação de informações com base na agenda para intercâmbio em XML (*EXtensible Markup Language*).





## Capítulo 5

---

### Conclusões

O objetivo deste trabalho foi o de observar as características de evolução de uma aplicação desenvolvida com base na utilização da Internet, sendo também um *groupware*, a agenda *No Risk Planning*, e observar como um processo de desenvolvimento foi utilizado para atender às solicitações de mudanças.

Inicialmente, foram estudados os princípios de CSCW e os conceitos sobre *groupware*, fundamentais para entendimento da finalidade básica da agenda *No Risk Planning*. A seguir, foi realizada uma análise das seis questões de contexto (*awareness*), que resumem as características consideradas mais importantes para se ponderar que tipo de suporte uma ferramenta de *groupware* deve priorizar. Esta análise foi relevante, pois proporcionou se justificar quais solicitações dos novos usuários da agenda seriam atendidas.

A partir de diversas solicitações de mudanças dos usuários e de inserção de novas funcionalidades na agenda, foi iniciada a evolução da mesma. Assim, as mudanças solicitadas motivaram o emprego de um processo de evolução, com base na Engenharia de *Web* e algumas técnicas de *Programação Extrema* (XP), conforme descritos no Capítulo 3. Com base na adoção deste processo à agenda *No Risk Planning*, pudemos viabilizar a implementação das novas versões da agenda, atendendo aos novos requisitos solicitados e possibilitando que o processo adotado garantisse qualidades ao produto e que o processo demonstrasse a produtividade da equipe de desenvolvimento.

Foram descritas as três versões que foram observadas detalhadamente e desenvolvidas, demonstrando a evolução da agenda para grupos na *Web*. O processo de evolução foi realizado sobre a agenda como em um estudo de caso, no qual a atualização de documentação, e o acompanhamento das etapas de evolução e dos requisitos foram essenciais para que se tivesse um produto com boa aceitação para uso no meio acadêmico.

Pesquisas futuras apontam para a continuidade na evolução da agenda, visando atender a outros domínios de informação e possibilitar, por meio de XML, intercâmbio de seus dados para outras aplicações na *Web*.

...

...

...

...

## Referências Bibliográficas

- [ARRUDA JR & PIMENTEL, 2001] – ARRUDA JR., C.R.E.; PIMENTEL, M.G.C. – “Projeto e Implementação de um Sistema Colaborativo de Edição.” – *Revista Eletrônica de Iniciação Científica da Sociedade Brasileira de Computação (REIC-SBC)*, Ano 1, 12 pg, Novembro 2001.
- [ATHULA & MURUGESAN, 2001] – ATHULA, G.; MURUGESAN, S. – “Web Engineering: An Introduction.” – *IEEE Multimedia*, Volume 8, Issue 1, p. 14-18. 2001.
- [BANNON & BODKER, 1997] – BANNON, Liam & BØDKER, Susanne. – “Constructing Common Information Spaces.” – *Proceedings of the 5th European CSCW Conference (ECSCW'97)*, p. 81-96, Lancaster – Reino Unido, 1997.
- [BECK, 2000] – BECK, Kent. – “Extreme Programming Explained.” – *Addison-Wesley*, 2000.
- [BENBUNAN et al, 2002] – BENBUNAN, Raquel; Hiltz, Starr Roxanne; Turoff, Murray. – “A Comparative Content Analysis of Face-to-face vs. Asynchronous Group Decision Making.” – *Decision Support Systems* 34, p. 457-469, 2002.
- [BOROVOY, 1998] – BOROVOY, Richard, et. al. – “Meme Tags and Community Mirrors: Moving From Conferences to Collaboration” – *Proceedings of the ACM Conference on Computer Supported Cooperative Work 1998*, p. 159-169, Washington – Estados Unidos, 1998.
- [BRINK, 1998] – BRINK, Tom. – “Usability First: Groupware: Applications.” (*OnLine*). – 1998. Disponível em: <http://www.usabilityfirst.com/groupware/applications.txt>. Acesso em 07/07/2003.
- [CARISTI, 2002] – CARISTI, James. – “Tutorial on Extreme Programming.” – *Proceedings of Consortium for Computing Sciences in Colleges 2002 (CCSC 2002)*, p. 33-34, Reno – Estados Unidos, 2002.
- [CHANG, ZHANG & JIANG, 2001] – CHANG, Carl J.; ZHANG, Jia; JIANG, Tsang Ming. – “Formalization of Computer Supported Cooperative Work Applications.” – *Proceedings of the Eighth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'01)*, p. 185-193, Bolonha – Itália, 2001.
- [COSTA, 1997] COSTA, R.M. – “Fusion-RE/I – Um Método de Engenharia Reversa para Auxiliar a manutenção de Software.” – *Dissertação de Mestrado*, ICMSC-USP, São Carlos – Brasil, 1997.
- [COSTAGLIOLA, FERRUCCI & FRANCESE, 2002] – COSTAGLIOLA, Gennaro; FERRUCCI, Filomena; FRANCESE, Rita. – “Web Engineering: Models and Methodologies for the Design Hypermedia Applications.” – *Handbook of Software Engineering & Knowledge Engineering, Emerging Technologies*, Vol. 2, p. 181-199, 2002.
- [DAVIS, 1998] – DAVIS, Richard C., et. Al. – “A Framework for Sharing Handwritten Notes.” – *Proceedings of User Interface Software and Technology (UIST'98)* – ACM Press, p. 119-120, San Francisco, USA, 1998.

- [DESHPANDE & HANSEN, 2001] – DESHPANDE, Yogesh; HANSEN, Steve. – “Web Engineering: Creating a Discipline among Disciplines.” – *IEEE Multimedia* em Abril/Junho-2001, p. 82-87, 2001.
- [DEURSEN, 2001] – DEURSEN, Arie van. – “Program Comprehension Risks and Opportunities in Extreme Programming.” – *Proceedings of Eighth Working Conference on Reverse Engineering (WCRE'01)*, p. 176-185, Stuttgart – Alemanha, 2001.
- [ELLIS et al, 1991] - ELLIS, C.A.; GIBBS, S.J.; REIN, G.L. - "Groupware: Some Issues and Experiences." - *Communications of the ACM* 34, Vol 1, p. 39-58, 1991.
- [ENGLISH, 2002] – ENGLISH, Arthur. – “Extreme Programming: It’s Worth a Look.” – *Revista It Pro* Maio/Junho-2002, p. 48-50, 2002.
- [FLUCKIGER, 1995] – FLUCKIGER, François. – “Taxonomy of multimedia applications in Understanding networked multimedia: Applications and technology. (Capítulo 6)” – *Prentice-Hall*, p. 109-121, 1995.
- [GAEDKE, 2002] – GAEDKE, Martin. – “The WebEngineering.org Community Site.” (*OnLine*). – 2002. Disponível em: <http://www.webengineering.org/>. Acesso em 07/07/2003.
- [GINIGE & MURUGESAN, 2001] - GINIGE, Athula; MURUGESAN, San. – “The Essence of Web Engineering: Managing the Diversity and Complexity of Web Application Development.” – *IEEE Multimedia* Abril/Junho- 2001, p. 22-25, 2001.
- [GOUVEIA, 2000] – GOUVEIA, Luís Manoel Borges. – “Ambientes Virtuais Colaborativos: a procura de formas alternativas de interação.” *Revista Politécnica* nº2. Dezembro, Edições da Cooperativa de Ensino Politécnico. ISSN 0874-8799, Porto – Portugal, 2000.
- [GUZDIAL, 1999] – Guzdial, M. – “Collaborative website supporting open authoring.” – 1999. *Georgia Institute of Technology (GeorgiaTech)*. Submitted to *Journal of the Learning Sciences*, p. 1-48, 1999.
- [HAWRYSZKIEWYCZ, 1998] – Hawryszkiewicz, Igor. – “An Evolution to a CSCW Design Methodology.” – *Third-First Annual Hawaii International Conference on Systems Sciences*, Vol 1, p. 536-544, Kohala Coast – Hawaii, 1998.
- [ISAAK, 2002] – ISAAK, Jim. – “Web Site Engineering Best Practice Standards.” – *Fourth Workshop on the Web Site Evolution (WSE'02)*, p. 81-84, Montreal – Canadá, 2002.
- [KLAPSING, NEUMANN & CONEN, 2001] – KLAPSING, Reinhold; NEUMANN, Gustaf; CONEN, Wolfram. – “Semantics in Web Engineering: Applying the Resource Description Framework.” – *IEEE Multimedia* Abril/Junho-2001, p. 62-68, 2001.
- [LEE, 2001] – LEE, Heejin. – “Your time and my time: a temporal approach to groupware calendar systems.” *Elsevier Information & Management* em Novembro-2001. p.159-164, 2001.
- [LINHALIS, 2000] – LINHALIS, Flávia. – “Interface Básica para um Servidor Universal.” – *Dissertação de Mestrado* – ICMC-USP, São Carlos – Brasil, 2000.

- [MACEDO, BULCÃO NETO & PIMENTEL 2001] MACEDO, A.A.; BULCÃO NETO, R.F.; PIMENTEL, M.G.C. "Autoria colaborativa na Web: Experiências e Reflexões sobre a CoWeb"; *Revista Brasileira de Informática na Educação (RBIE)*, Setembro 2001.
- [MAURER, 2002] – MAURER, Frank. – "Supporting Distributed Extreme Programming." – *Proceedings on Extreme Programming and Agile Methods – XP/Agile Universe 2002*, p. 13-2, Chicago – Estados Unidos, 2002.
- [MCDONALD & WELLAND, 2001] – MCDONALD, Andrew; WELLAND, Ray. – "Web Engineering in Practice." – *Proceedings of Web Engineering Workshop at WWW10*, p. 21-30, Hong Kong – China, 2001.
- [MILLS, 1999] – MILLS, Kevin L. – "Introduction to the Electronic Symposium on Computer Supported Cooperative Work." *ACM Computing Surveys*, Vol 31, N° 2, June 1999.
- [MOREIRA, 1998] – MOREIRA, Gustavo. – "Sumário Executivo." – Dissertação de Mestrado – Universidade Federal do Rio de Janeiro (UFRJ), 1998.
- [MOURA, 2001] – MOURA, Marcos Vinicius. – "No Risk Planning - um sistema de suporte ao trabalho em grupo: uma proposta para Web." – Projeto de Graduação (monografia) – ICMC-USP, São Carlos – Brasil, 2001.
- [MÜLLER & TICHY, 2001] – MÜLLER, Mathias M.; TICHY, Walter F. – "Case Study: Extreme Programming in a University Environment." – *Proceedings of International Conference on Software Engineering (ICSE'01)*, p. 537-544, Toronto - Canadá, 2001.
- [MURUGESAN & DESHPANDE, 2002] – MURUGESAN, Sam; DESHPANDE, Yogesh. – "Meeting the Challenges of Web Application Development: The Web Engineering Approach." – *Proceedings of International Conference on Software Engineering (ICSE'02)*, p. 687-688, Orlando - Estados Unidos, 2002.
- [NEWKIRK, 2002] – NEWKIRK, James. – "Introduction to Agile Processes and Extreme Programming." – *Proceedings of the 24th International Conference on Software Engineering (ICSE'01)*, p. 695-696, Orlando – Estados Unidos, 2002.
- [NEWKIRK & MARTIN, 2000] – NEWKIRK, James; MARTIN, Robert C. – "Extreme Programming in Practice." – *Proceedings of ACM Conference Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2000)*, ACM Press, p. 25-26, Minneapolis – Estados Unidos, 2000.
- [NIELSEN & WAGNER, 1996] – NIELSEN, Jakob; WAGNER, A. – "User Interface for the World Wide Web." – *Proceedings of Conference on Human Factors in Computing Systems*, ACM Press, Vancouver – Canadá, 1996.
- [NITZKE, CARNEIRO & GELLER, 1999] – NITZKE, Júlio; CARNEIRO, Mara; GELLER, Marliise. – "Avaliando aplicações para a criação de ambientes de aprendizagem colaborativa." – *Anais do Simpósio Brasileiro de Informática e Educação (SBIE'99)*, p. 303-310, Curitiba - Paraná. 1999.
- [PASSOS & BERNARDO, 2001] – PASSOS, Clotilde; BERNARDO, Hélder. – "Trabalho em Grupo e Colaborativo – Levantamento de Informações na Web." – *Dissertação de Mestrado* – Universidade de Aveiro – Portugal, 2001.

- [PINHEIRO, LIMA & BORGES, 2001] – PINHEIRO, Manuelle Kirsch.; LIMA, José Valdeni; BORGES, Marcos R.S. – "Awareness em Sistemas de Groupware." – *Proceedings of International Database Engineering & Applications Symposium (IDEAS'01)*, p. 323-335, Montreal – Canadá, 2001.
- [PINHEIRO, 2001] – PINHEIRO, Manuelle Kirsch. – "Mecanismo de Suporte à Percepção em Ambientes Cooperativos." – *Dissertação de Mestrado*, PPGC/UFRGS, Porto Alegre - Brasil, 2001. Disponível Online em: <http://www-lsr.imag.fr/Les.Personnes/Manuele.Kirsch-Pinheiro/MscKirschPinheiro.pdf>. Acesso em: 24/06/2003.
- [PORTO et al, 1994] – PORTO, Alessandra Monteiro de Castro; OLIVEIRA, João Paulo Souza; OLIVEIRA, Jorge Ricardo Souza; KAIR, Tatiana. – "Trabalho Cooperativo Apoiado por Computador." *Relatório Técnico RT-001/94* – Universidade Federal de Santa Catarina (UFSC), 1994. Disponível online em <http://www.inf.ufsc.br/~pet/projetos/p952/rtescw01/rtescw01.htm>
- [PRESSMAN, 2002] – PRESSMAN, R.S. – "Engenharia de Software 6ª Edição. (Capítulo 29)" – *Makron Books*, 2002.
- [RAMACHANDRAN & SHUKLA, 2002] – RAMACHANDRAN, Vinay; SHUKLA, Anuja. – "Circle of Life, Spiral or Death: Are XP Teams Following the Essential Practices?." – *Proceedings of Extreme Programming and Agile Methods – XP/Agile Universe 2002*, p. 166-173, Chicago – Estados Unidos, 2002.
- [RASMUSSEN, 2002] – RASMUSSEN, Jonathan. – "Strategies for Introducing XP to New Client Sites." – *Proceedings of Extreme Programming and Agile Methods – XP/Agile Universe 2002*, p. 45-51, Chicago – Estados Unidos, 2002.
- [REIFER, 2002] – REIFER, Donald J. – "How to get the Most out of Extreme Programming/Agile Methods." – *Proceedings of Extreme Programming and Agile Methods – XP/Agile Universe 2002*, p. 185-196, Chicago – Estados Unidos, 2002.
- [RIBEIRO, FORTES & FREIRE, 2002] – RIBEIRO, Tyciano M.; FORTES, Renata P.M.; FREIRE, André P. – "Documentação do Software Agenda "No Risk Planning" " – *Relatórios Técnicos do ICMC 182*, ICMC-USP/São Carlos, 2003.
- [SANTOS et al, 1998] – SANTOS, Antônio Carlos; ZANFOLIN, Tédi Itamar; KAYO, Ricardo Muneo; SILVA, Brenner Stefan Gomes. – "Percepções do Usuário em um Ambiente de Trabalho Cooperativo para Apoio na Tomada de Decisão em Grupo." – *1 Workshop sobre Fatores Humanos em Sistemas Computacionais: Compreendendo Usuários, Construindo Interfaces (IHC'98)*, p. 96-105, Rio de Janeiro – Brasil, 1998.
- [SCHMIDT, 1999] – SCHMIDT, Kjeld. – "Cooperative work: a conceptual framework: The Informatics Center of the Danish Trade Union Movement." – *Proceedings of The 6th European Conference on Computer Supported Cooperative Work (ECSCW'99)*, p. 75-109, Copenhagen – Dinamarca, 1999.
- [SCHNEIDER & JOHNSTON, 2003] – SCHNEIDER, Jean-Guy; JOHNSTON, Lorraine. – "eXtreme Programming at Universities – An Educacional Perspective." – *Proceedings of*

*International Conference on Software Engineering (ICSE 2003)*, IEEE Computer Society Press, pp 594-599, Portland – Estados Unidos, 2003.

- [SCHWABE et al, 2001] – SCHWABE, Daniel; Rossi, Gustavo; Esmeraldo, Luíselena; Lyardet, Fernando. – “Engineering Web Applications for Reuse.” – *IEEE Multimedia*. Vol 8, p. 1-12, 2001.
- [SMITH & STOECKLIN, 2001] – SMITH, Suzanne; STOECKLIN, Sara. – “What We Can Learn from Extreme Programming.” – *Proceedings of the Fifteenth Annual CCSC Southeastern Conference (CCSC 2001)*, Volume 17, Number 2, p. 135-142, Greenville – Estados Unidos, 2001.
- [STREITZ, 1994] – STREITZ, Norbert A. – “DOLPHIN: Integrated Meeting Support Across LiveBoards, Local and Remote Desktop Environments.” – *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, p. 345-358, Chapel Hill – Estados Unidos, 1994.
- [SOARES, FORTES & MOREIRA, 2000] - SOARES, M.D.; FORTES, R.P.M.; MOREIRA, D. A. – “VersionWeb: A Tool for Helping Web Pages Version Control.” - *Proceedings of the International Conference on Internet Multimedia. Systems and Applications (IMSA 2000)*, p. 275-280, Las Vegas - USA, 2000.
- [SOARES & FORTES, 2001] - SOARES, M.D.; FORTES, R.P.M. – “Um Suporte ao Controle de Versões na Web.” - *XV Simpósio Brasileiro de Engenharia de Software – Sessão de Ferramentas*, Anais. p.422-427, Rio de Janeiro – Brasil, 2001
- [SUMI & MASE, 2000] – SUMI, Yasuyuki; MASE, Kenji. – “Supporting Awareness of Shared Interests and Experiences in Community.” – *ACM SIGGROUP Bulletin*, Vol.21, No 3, p. 35-42, 2000.
- [Usability First, 2002] – “Usability First: Groupware.” (*OnLine*). – 2002. Disponível em <http://www.usabilityfirst.com/groupware/>. Acesso em 07/07/2003.
- [WILLIAMS, 2003] – WILLIAMS, Laurie. – “The XP Programmer: The Few-Minutes Programmer.” – *IEEE Software* Maio/Junho - 2003, p. 16-20. 2003.
- [WILLIAMS & UPCHURCH, 2001] – WILLIAMS, Laurie; UPCHURCH, Richard. – “Extreme Programming for Software Engineering Education?” – *Session of 31<sup>th</sup> Frontiers in Education Conference*, p. 01-06, Reno – Estados Unidos, 2001.
- [WOOD & KLEB, 2003] – WOOD, William A.; KLEB, William L. – “Exploring XP for Scientific Research.” – *IEEE Software Society* Maio/Junho - 2003, p. 30-36. 2003.