

## 4 - Experimento prático (C&L)

Este capítulo fala da ferramenta de software livre denominada C&L, no próximo capítulo será apresentada uma proposta de união de cenários com o código em software livres, essa proposta teve como laboratório de estudos esta ferramenta aqui apresentada. Esta ferramenta criada na PUC-Rio foi de inestimável importância para a compreensão deste tipo de desenvolvimento de software e para os testes da proposta. Este capítulo descreverá sua história, suas funcionalidades e principalmente focará no aprendizado que foi desenvolvido em cima dela.

A ferramenta C&L utiliza a MIT License, que é extremamente flexível, e permite que seu código fonte e produtos derivados sejam usados e distribuídos livremente, sejam em projetos comerciais ou software livres sem qualquer custo para o usuário.



### 4.1 - História

O software C&L é resultado da evolução de um software livre desenvolvido durante a disciplina Princípios de Engenharia de Software, oferecida pela Puc-Rio ao curso de Engenharia de Computação, no período de 2002.1. Esta aplicação tem

o objetivo de oferecer aos usuários funcionalidades de edição de Cenários e Léxico.

A ferramenta C&L é um projeto de software livre feito e evoluído usando as linguagens PHP, HTML e JavaScript no laboratório de Engenharia de Software (LES) da PUC-Rio. A ferramenta consiste de aproximadamente 50 módulos, que incluem páginas de exibição da ferramenta, arquivos de ajuda e bibliotecas de funções PHP compartilhadas pelo código. Seu código foi feito e testado por alunos de graduação e pós-graduação da Puc-Rio, sendo que o autor dessa dissertação foi gerente durante a primeira evolução da ferramenta, que deu origem ao C&L, sendo responsável por boa parte da codificação e elaboração destes módulos.

A aplicação original foi usada posteriormente como um caso de estudo em uma outra disciplina também ministrada na Puc-Rio pelo professor Julio Cesar Sampaio do Prado Leite para alunos da Engenharia de Computação e pós-graduação em Engenharia de Software. A disciplina “Evolução de Software” tinha como proposta apresentar os assuntos intrínsecos à evolução de software através de um trabalho prático que possibilitasse aos alunos discutir e relacionar trabalhos científicos a trabalhos práticos na atividade realizada. O trabalho prático em questão consistia em evoluir a ferramenta livre de edição de cenários e léxicos tendo como foco o desenvolvimento de software livre.

A equipe era composta de treze alunos da pós-graduação da Puc-Rio, sendo nove alunos de mestrado e quatro de doutorado. A equipe do projeto foi dividido em grupos onde os integrantes se organizaram voluntariamente segundo seus interesses de estudo e conhecimentos prévios. Dessa forma, todos os integrantes acabaram participando de mais de um grupo. Os seguintes grupos foram formados: desenvolvimento, qualidade, controle de versão, documentação, banco de dados e gerência e administração.

Foi criada uma estrutura hierárquica para centralizar a coordenação do projeto. O autor dessa dissertação ficou como gerente do projeto, ficando

encarregado de acompanhar o progresso dos trabalhos e, na medida do possível, resolver os conflitos e demandas dos diversos grupos. Além disso, foi feito o uso intenso de sistemas de Groupware (principalmente ferramentas de correio eletrônico e uma lista de discussão) de modo que todos os integrantes pudessem apresentar os resultados obtidos, expor dúvidas e trocar experiências com os demais.

O software original foi analisado e a partir desta análise foram feitas melhorias, tais como a troca do banco de dados de PostGre para MySQL, substituição da versão da linguagem PHP3 para PHP4, implementação de novas funcionalidades e melhoria das existentes. Também foi utilizada engenharia reversa para melhorar a documentação existente.

A esta nova ferramenta foi dado o nome de C&L, e todo seu desenvolvimento foi realizado utilizando-se software livre. São eles: PHP [PHP03] versão 4, como linguagem de implementação, o banco de dados MySQL [MySQL03], o servidor Web Apache HTTP Server [Apache03] e a ferramenta de controle de versão e gerenciamento dos códigos-fonte CVS [CVS03].

A versão atual disponibiliza livremente o código fonte do C&L, todos podem baixar, modificar e usar o código como desejarem seja para uso comercial ou livre. Qualquer um pode contribuir com o desenvolvimento da ferramenta, mas o código submetido deverá ser aprovado pelos administradores para fazer parte da versão oficial. Foi feito recentemente um *plug-in* de geração de ontologias para o C&L, e a nova versão da ferramenta já está disponível para *download*.

## 4.2 - A ferramenta

A ferramenta C&L implementa um ambiente colaborativo que auxilia a edição de cenários e léxicos descritos em linguagem natural semi-estruturada.

Para a melhor compreensão da ferramenta como um todo e sua importância apresentaremos os conceitos de cenário e léxicos e como eles são aplicados na ferramenta.

#### **4.2.1 - Léxicos e cenários**

Entendemos por cenários a descrição de situações comuns ao cotidiano [Zorman95].

Os cenários devem levar em conta aspectos de usabilidade e permitir o aprofundamento do conhecimento do problema, a unificação de critérios, a obtenção do compromisso de clientes e/ou usuários, a organização de detalhes e o treinamento de pessoas [Carroll94].

Cada cenário descreve, através de linguagem natural semi-estruturada, uma situação específica da aplicação, focando em seu comportamento. Cenários podem ser detalhados e utilizados como desenho de maneira a auxiliar a programação. Existem várias propostas para a representação de cenários, desde a mais informal, em texto livre [Carroll94] até representações formais [Hsia94].

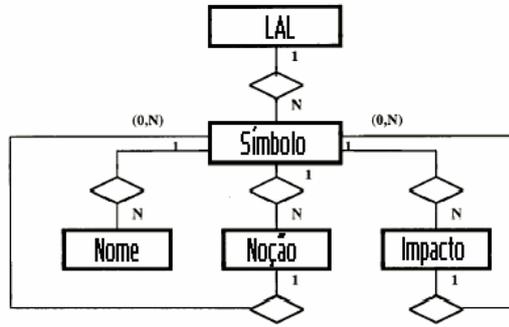
A representação mostrada em [Hsia94] é extremamente formal, onde é dito que todo cenário deve ser elicitado com a construção de uma árvore de cenário (um mecanismo que mostra todos os cenários para uma determinada visão), estabelecendo para tanto, as diversas visões para cada usuário e seus agentes (que podem ser usuários externos, estímulos externos, ou componentes funcionais). Após essas definições, é necessário formalizar cada cenário, transformando cada árvore em uma gramática regular, é necessário então checar por inconsistências e redundâncias para posteriormente usar um processo automático de geração de cenários, gerando um protótipo que possa validá-lo. Esta representação é por demais formal, e tem o inconveniente de não poder representar sistemas que possuam mais de uma resposta por estímulo, não podendo representar sistemas com concorrência.

Já a representação mostrada em [Carroll94] apresenta o outro extremo, sendo mostrada uma representação totalmente em texto livre, sem preocupação com qualquer estrutura de representação.

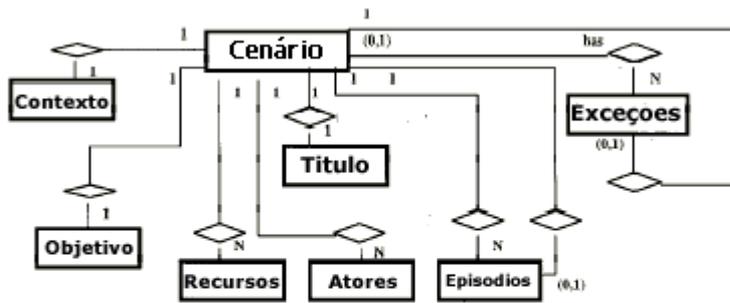
A ferramenta C&L utiliza uma representação intermediária que, ao mesmo tempo em que facilita a compreensão através da utilização de linguagem natural, força a organização da informação através de uma estrutura bem definida. Esta estrutura é composta por elementos descritivos que expressam: o objetivo, o contexto, os recursos, os atores, episódios (ações) e as exceções que ocorrem nessas atividades. O conjunto dessas características representa uma situação. Além dessas características a representação também reserva um atributo, restrições, que pode ser usado em contexto, recurso e episódios para refletir aspectos não-funcionais.

O processo de construção de cenários está relacionado à existência do Léxico Ampliado da Linguagem (LAL). O LAL é a implementação de conceitos de semiótica num hipergrafo [Leite 93], onde tanto a denotação como a conotação são expressas para cada símbolo do léxico. Estas descrições seguem o princípio da circularidade e o princípio de vocabulário mínimo. O princípio da circularidade faz com que cada descrição de denotação ou conotação faça referência a outros símbolos da linguagem. As partes da descrição que não são símbolos devem ser de um subconjunto reduzido de palavras com significado bem definido (vocabulário mínimo).

Os termos utilizados no LAL são descritos de maneira a retratar dois aspectos: a noção (ou denotação) e o impacto (ou conotação). A noção é o que o termo significa e o impacto representa como o termo exerce influência nesse contexto. O LAL permite também que cada elemento do vocabulário mínimo tenha um ou mais sinônimos. Cada sinônimo teria noção e impactos iguais aos seus termos, mas seriam nomeados de forma diferente.



**Diagrama 4.1** - Diagrama de entidade-relacionamento do LAL segundo Leite[00].



**Diagrama 4.2** - Diagrama de entidade-relacionamento do cenário segundo [Leite00].

### 4.2.2 - Público alvo

O público alvo desta ferramenta é principalmente o engenheiro de software; estudantes e interessados na área de requisitos também terão ganhos de produtividade nas tarefas de edição e visualização da informação. Contudo, mesmo para aqueles que não estejam familiarizados com as técnicas do Léxico Ampliado da Linguagem e Cenários é possível utilizá-la, pois, ela é apresentada através de formulários intuitivos que recebem e estruturam a informação.

### 4.2.3 - Objetivos e funcionalidades da ferramenta

A ferramenta C&L tem como objetivo criar um ambiente colaborativo para a gerência do léxico e cenários do domínio da aplicação. Entendemos por gerenciar, as tarefas de criação, edição, manutenção e evolução do léxico e dos cenários. A informação é disponibilizada em linguagem natural semi-estruturada, e organizada de forma simples e rápida para seu acesso, podendo ser visualizada e editada por um grupo de usuários cadastrados em um mesmo projeto.

A ferramenta fornece suporte à criação de ligações (*links*) entre os artefatos, em particular o léxico, que é naturalmente implementado no formato de hipergrafo. Ligações são criadas automaticamente pela ferramenta quando termos já existentes são citados, facilitando a navegabilidade entre os artefatos e permitindo a compreensão dos relacionamentos entre os conceitos do domínio descrito com poucos cliques do mouse.

É importante lembrar que a ferramenta foi desenvolvida tendo em vista ser um ambiente colaborativo, para tanto a ferramenta implementa dois níveis de acesso para o sistema: usuário e administrador. Quando estiver participando de um projeto, o usuário poderá visualizar, criar, alterar e remover léxicos e cenários deste, mas caso ele não seja administrador do projeto, suas ações terão que ser aprovadas antes de serem efetivamente realizadas.

Cada projeto deve ter um administrador e não há limite para o número de usuários. Entre as funcionalidades exclusivas do administrador do projeto se encontram: aprovar inserções, mudanças e remoções de cenários e termos do léxico feitos por outros usuários, remover o projeto, adicionar ou remover usuários neste projeto, gerar e recuperar XML com o conteúdo do projeto.

A ferramenta C&L apresenta os cenários e termos do léxico utilizando uma interface gráfica que pode ser visualizada através de um navegador padrão. O usuário pode escolher se irá instalar o software em um servidor próprio ou se usará a versão disponível na página oficial localizada em [C&L03].

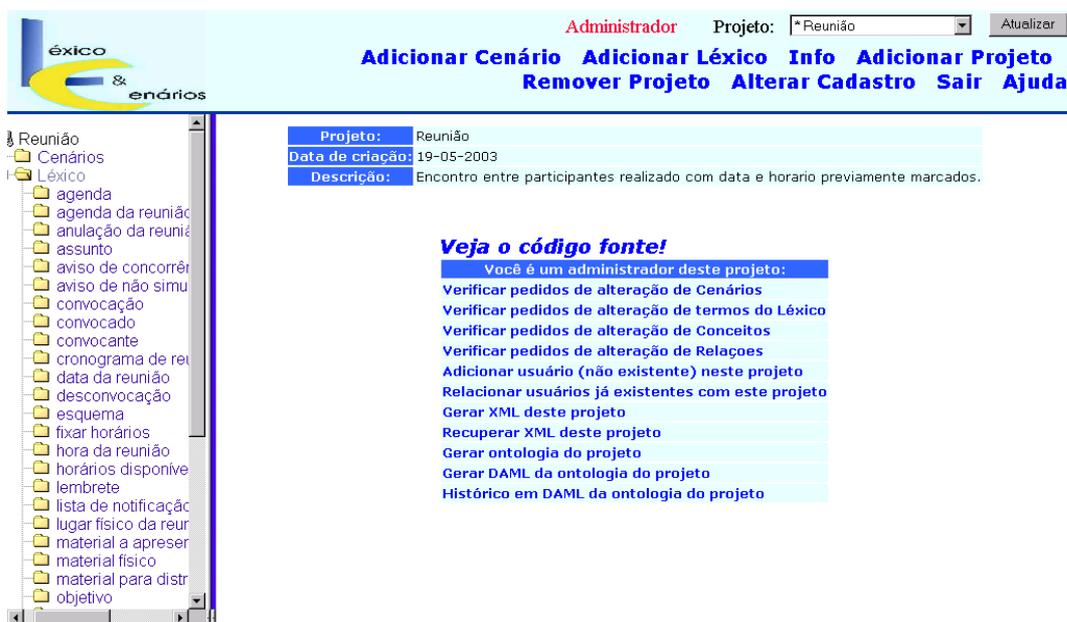
The screenshot displays the C&L web interface. At the top, there is a header with the logo 'léxico & cenários' on the left. The header also shows the user's access level as 'Usuário normal' and the current project as 'CEL Edit'. A navigation menu includes options like 'Adicionar Cenário', 'Adicionar Léxico', 'Info', 'Adicionar Projeto', 'Alterar Cadastro', and 'Sair Ajuda'. The main content area is titled 'Informações sobre o léxico' and contains a form with the following fields:

- Nome:** projeto
- Noção:** É constituído de **Léxico** e **Cenários**. É uma abstração utilizada para denotar o **Léxico** e os **Cenários** de algum sistema que esteja sendo desenvolvido.
- Classificação:** objeto
- Impacto(s):**
- Sinônimo:**

Below the form, there are buttons for 'Alterar Léxico' and 'Remover Léxico'. A section titled 'Cenários e termos do léxico que referenciam este termo' shows a list of terms: 'usuário' and 'administrador'. A note indicates that these are terms of the LAL that reference this term. At the bottom, there is a link 'Veja o código fonte!'.

Figura 4.1 - Edição de um termo do LAL no C&L.

Na versão mais recente da ferramenta, já é possível para o administrador do projeto, gerar e recuperar uma ontologia do projeto e gerar o DAML da ontologia do projeto, além de poder visualizar o histórico em DAML.



**Figura 4.2** - Funcionalidade exclusivas do administrador do projeto.

Não encontramos atualmente no mercado uma ferramenta que trate da edição de cenários e léxicos de acordo as regras estabelecidas em [Leite00]. Assim, o engenheiro de software que deseja utilizar essa notação é obrigado a usar alguma ferramenta mais genérica e não tão indicada para a tarefa, como um editor de textos. O C&L preenche essa lacuna oferecendo um ambiente diferenciado, e com suporte a trabalho colaborativo, além de oferecer diversas funcionalidades extras para o usuário. Estas funcionalidades estão descritas na tabela 1.

<b>Geral</b>	<b>Léxico</b>	<b>Cenário</b>	<b>Ontologia</b>	<b>Entrada e saída</b>
<ul style="list-style-type: none"> <li>- Criar projeto e administrador;</li> <li>- Cadastrar usuário no projeto;</li> <li>- Verificar e aprovar ou rejeitar pedidos de alterações nos cenários e léxicos;</li> <li>- Exportar arquivo XML;</li> <li>- Recuperar arquivo XML;</li> <li>- Criação de um banco de dados persistente para armazenar léxicos e cenários.</li> </ul>	<ul style="list-style-type: none"> <li>- Edição: criar, alterar ou remover;</li> <li>- Adição de sinônimos;</li> <li>- Marcação automática dos termos do LAL, seus sinônimos e nomes dos cenários;</li> <li>- Verificação de consistência em consequência da remoção de termos.</li> </ul>	<ul style="list-style-type: none"> <li>- Edição: criar, alterar ou remover;</li> <li>- Marcação automática dos termos do LAL, seus sinônimos e nomes dos cenários;</li> <li>- Verificação de consistência em consequência da remoção de cenários.</li> </ul>	<ul style="list-style-type: none"> <li>- Gerar ontologia</li> <li>- Gerar DAML</li> <li>- Histórico em DAML</li> </ul>	<ul style="list-style-type: none"> <li>- Exportar arquivo XML.</li> </ul>

**Tabela 4.1 - Funcionalidades oferecidas pela ferramenta C&L.**

### **4.3 - Ferramentas utilizadas na criação e evolução do C&L**

Como já foi dito, todo o desenvolvimento da ferramenta C&L foi feito utilizando-se software livre, além de que todas as ferramentas necessárias para a instalação do C&L em um servidor próprio possuem versão gratuita. As ferramentas necessárias em questão são um banco de dados e um servidor HTTP, o C&L se utiliza do MySQL [MySQL03] e do Apache HTTP Server [Apache03] que são excelentes opções gratuitas. A linguagem escolhida é a também gratuita PHP, e a ferramenta de controle de versão e gerenciamento do código-fonte é o CVS [CVS03].

Dessa forma foi possível desenvolver um sistema de qualidade com custo reduzido, já que os programas foram obtidos gratuitamente e sem nenhuma restrição de uso, os desenvolvedores eram todos voluntários e o único quesito que poderia implicar em custo direto, que é o hardware necessário para a instalação do servidor e a hospedagem deste, foi doado pelo Departamento de Informática da Puc-Rio. O hardware consistia de uma máquina de configuração modesta e a

hospedagem é feita em um dos laboratórios do departamento (Laboratório de Engenharia de Software [LES03]).

Segue uma pequena descrição de cada uma das ferramentas usadas no C&L.

#### **4.3.1 – MySQL**

A companhia responsável pelo desenvolvimento, suporte e distribuição do MySQL é a MySQL AB, que distribui o banco de dados e o código fonte segundo a licença GPL. Também é possível comprar o produto com uma licença comercial para o caso de seu usuário não desejar as restrições impostas pela GPL. De posse de uma licença comercial do MySQL é possível comercializar um produto que seja distribuído com o banco de dados. O retorno financeiro da MySQL AB vem da venda destas licenças comerciais e do suporte pago que oferece a todos os usuários.

Hoje em dia, o MySQL é um dos bancos de dados de código aberto mais populares do mundo, com mais de 4 milhões cópias instaladas, sendo usadas em aplicações como páginas na Internet, sistemas de negócios, sistemas de busca e outros. Entre os grandes clientes, destacam-se o Yahoo!, MP3.com, Motorola, NASA e Silicon Graphics que fazem uso do banco de dados para aplicações críticas [MySQL03].



#### **4.3.2 – Apache HTTP Server**

A história deste servidor HTTP e suas características foram detalhada no capítulo 1, não sendo portanto necessário repeti-las. A escolha de qual servidor

HTTP usar foi bem fácil, já que o Apache HTTP Server possui mais de 60% do mercado mundial, é gratuito, sua integração com PHP é excelente, é um software leve, com uma ótima performance e suporte da comunidade, a escolha deste servidor foi portanto uma unanimidade entre os desenvolvedores do projeto.

#### 4.3.3 – PHP (PHP: Hypertext Preprocessor )



É uma linguagem de *script* especialmente feita para desenvolvimento Web, podendo ser colocada diretamente no HTML. Hoje em dia, essa é a linguagem de *script* mais usada na Internet, sendo reconhecida mundialmente pela facilidade de uso e velocidade.

PHP começou com algumas modificações feitas na linguagem Perl em 1994 por Rasmus Lerdorf. Nos dois anos seguintes ela evoluiu para uma versão conhecida como PHP/FI 2.0, que conquistou uma boa quantidade de usuários, mas a grande popularização da linguagem começou em 1997 quando Zeev Suraski e Andi Gutmans fizeram algumas modificações que levaram à criação da versão 3.0. Essa versão definiu a semântica e a sintaxe usadas nas versões 3 e mais recentemente na 4.

Muito da sintaxe vem da linguagem C, Java e Perl com algumas características específicas encontradas apenas na linguagem PHP. Esta sintaxe visa o rápido aprendizado da linguagem, já que são padrões conhecidos mundialmente.

A linguagem é totalmente gratuita e o código fonte do interpretador é disponível para todos, existindo inclusive várias listas de discussão com o propósito de discutir erros encontrados e sugerir melhorias.

#### 4.3.4 - CVS (Concurrent Versions System)



É um sistema de controle de versões que permite que se mantenha um histórico dos seus arquivos fonte, sendo possível realizar o armazenamento de várias versões e árvores de desenvolvimento dos mesmos. O armazenamento é feito de forma inteligente, já que as versões são guardadas em um único arquivo contendo as diferenças entre elas, e não os arquivos inteiros, desta forma economizando espaço físico [CVS03].

Algumas funções permitidas pelo CVS são armazenamento do código fonte, criação de árvores de desenvolvimento, fechamento de versões, diferença entre arquivos de versões distintas e recuperação de versões. Também é possível com a edição de arquivos simultaneamente, o que permite que duas ou mais pessoas trabalhem em um mesmo arquivo. Para tanto o CVS cria uma versão exclusiva do arquivo para cada desenvolvedor e depois adiciona as alterações feitas por cada um em um arquivo final.

O CVS começou com alguns simples pedaços de código escritos por Dick Grune no *newsgroup* comp.sources.unix em 1986. Foi baseado neste código que em 1989, Brian Berliner codificou o CVS, sendo ajudado mais tarde por Jeff Polk [Wikipedia03].

CVS é um software livre que possui versões para várias plataformas, além de várias interfaces gráficas disponíveis e também livres. Existem no mercado diversas outras ferramentas de controle de versão, mas o CVS se destaca pelo grande número de usuários e por ser quase uma unanimidade como ferramenta de controle de versões padrão para software livre.

#### **4.4 - Lições com o ambiente de software livre**

O processo de evolução que veio a gerar a ferramenta C&L foi feito com base no processo de desenvolvimento de software livre. É importante lembrar, que nenhum dos treze membros possuía experiência anterior com o desenvolvimento de software livre, o nível de conhecimento da equipe sobre o processo era basicamente o de usuários de sistemas deste tipo e iniciantes no assunto.

O trabalho de desenvolvimento da equipe do projeto foi dividido em grupos onde cada integrante especializou-se naquilo que mais tinha afinidade. O trabalho, na maioria das vezes, foi remoto num ambiente de desenvolvimento colaborativo, baseado no processo de desenvolvimento cooperativo conjunto e compartilhado.

Programadores de software livre são pessoas com conhecimentos distintos que usam da prática de programação do software livre um passatempo, ou seja, não existe um período ou horário pré-estabelecido para o desenvolvimento sendo, sobretudo uma atividade prazerosa. Devido a este fato, não havia um cronograma específico ou alguma exigência de horário de trabalho, mas vale ressaltar que, apesar da flexibilidade no desenvolvimento, existia uma preocupação com o cumprimento de certos prazos pré-estabelecidos, o que nem sempre acontece na maioria dos projetos deste tipo.

Para que o desenvolvimento do projeto atingisse suas metas, criou-se uma estrutura hierárquica em que existiam pessoas com determinadas responsabilidades e deveres, e o próprio desenvolvimento também caminhou com

regras específicas para cada grupo. Essas regras eram propostas e debatidas entre os participantes, de forma que todos pudessem participar de tarefas que considerassem interessantes. Como a divisão dos grupos e tarefas partiu dos próprios integrantes, formou-se um grupo altamente motivado e interativo.

A comunicação entre os membros era feita com ferramentas de Groupware de uso comum na Internet, sendo que a principal delas foi sem dúvida uma lista de discussão criada exclusivamente para o projeto. Nesta lista, várias mensagens foram trocadas diariamente, novidades foram informadas, votações realizadas, além de gerenciamento através de cobrança de prazos, divisão de tarefas, criação e divulgação de documentações.

#### **4.4.1 – Período de evolução**

O tempo estimado para a evolução da ferramenta pelo grupo era um curto período de quatro meses.

O primeiro desafio encontrado foi o de separar a equipe de treze alunos em grupos responsáveis por tarefas distintas. Foram criados seis grupos, e os membros de cada grupo foram definidos voluntariamente, ou seja, cada integrante se colocou a disposição para contribuir com as áreas do projeto com as quais possuía maior interesse ou afinidade. Todos os integrantes foram voluntários em pelo menos duas áreas, sendo que estas áreas eram:

- Desenvolvimento - responsável pela codificação da ferramenta em si, sendo um das áreas com o maior número de integrantes e tarefas.
- Qualidade - responsável pelos aspectos relativos ao controle da qualidade dos artefatos de software produzidos no processo de evolução.
- Controle de versão - responsável pelo controle do versionamento do sistema.

- Documentação - responsável pela evolução das documentações já existentes do sistema e criação das novas.

- Banco de dados - responsável pelo gerenciamento e modelagem do banco de dados da ferramenta.

- Gerência e administração - responsável pelo gerenciamento do projeto como um todo.

Todos estes grupos, à exceção de um, contavam com pelo menos três membros. O grupo de gerência e administração contava com apenas um participante que ficou responsável pelo estabelecimento de metas, gerência dos prazos estabelecidos e resolução de conflitos. Vale a pena lembrar que a pessoa responsável pela administração também fazia parte da equipe de desenvolvimento e que todas as decisões tomadas foram levadas para o grupo para discussão e troca de idéias.

A primeira tarefa realizada foi a instalação do ambiente necessário para o desenvolvimento do projeto, foi feita então a instalação do servidor junto com as ferramentas citadas anteriormente (banco de dados, linguagem de programação, servidor HTTP e ferramenta de controle de versão), estas tarefas foram realizadas pelo grupo de desenvolvimento em conjunto com o de banco de dados e controle de versão. Paralelamente, o grupo de gerência e administração criou o ambiente de Groupware necessário para a equipe, com a criação da lista de discussão e o acesso remoto à máquina servidora.

No início as principais tarefas se basearam na prática de engenharia reversa, em que a partir do código fonte, eram geradas documentações do sistema. Essa tarefa serviu para a equipe adquirir o conhecimento necessário do projeto para que fossem levantadas as necessidades de evolução do sistema.

Juntamente com a engenharia reversa, o início do projeto foi marcado também pelo estudo da teoria e prática de uso de cenários e do LAL, de modo que

fosse possível gerar um sistema que correspondesse a realidade e ajudasse o engenheiro de software da melhor forma possível. Também foi necessário o estudo da linguagem usada por parte da equipe de desenvolvimento, já que apenas dois dos seis integrantes tinham tido contato com a linguagem PHP.

Uma das primeiras surpresas do projeto foi o fato de que a ferramenta não ficou totalmente funcional devido a versão da linguagem PHP instalada ser mais recente que a versão na qual ela tinha sido desenvolvida. Desta forma uma das prioridades do projeto foi a de evoluir a ferramenta para a nova versão da linguagem.

A documentação resultante da engenharia reversa e as propostas para o projeto foram alvos de intensas discussões na lista do grupo, sendo identificadas oportunidades de melhorias na documentação e no código. Após alguns dias chegou-se à lista de tarefas que deveriam ser realizadas no período estipulado de quatro meses do projeto e cada membro se responsabilizou por uma parte destas tarefas a qual tinha interesse.

No início do projeto, existiu um período de aproximadamente um mês no qual pouco foi feito em relação ao código; este período de tempo foi necessário para que o grupo como um todo se acostumassem à ferramenta, à linguagem de programação e ao método de desenvolvimento usado em software livres. Após esse mês, assim que as metas do projeto foram estabelecidas, o desenvolvimento foi bem acelerado, e a lista de discussão permaneceu bem movimentada durante todo esse período, o que é um indicador claro de interesse por parte do grupo, que buscava na lista solução para suas dúvidas e expunha suas novas idéias para a ferramenta.

Durante os três meses seguintes, todos não apenas trabalharam em suas respectivas áreas, como também interagiram e usaram a ferramenta constantemente, desta forma todos os erros encontrados eram enviados para a lista para discussão imediata. Infelizmente o projeto não contou com uma base de dados para o catálogo de erros encontrados, desta forma todas as descrições destes

se encontram apenas nos arquivos de mensagens antigas da lista de discussão.

Após um primeiro mês de projeto de poucos resultados práticos, os três restantes foram bem dinâmicos e muito trabalho foi realizado neste período, principalmente se levarmos em conta que todos trabalhavam na ferramenta apenas no seu tempo livre. Todos os seis grupos interagiram bastante durante o projeto e após os quatro meses estipulados, a ferramenta que agora se chamava C&L, atingiu os requisitos definidos inicialmente.

O código da ferramenta C&L é aberto e está disponível na página [C&L03], juntamente com o código pode ser obtido um exemplo de base de dados preenchida no formato MySQL para uso imediato da ferramenta. O usuário pode fazer uso do código da ferramenta sem qualquer restrição, o código pode ser usado livremente em projetos comerciais ou não.

#### **4.4.2 – Aprendizado**

Este projeto deu ao grupo envolvido uma visão prática da evolução de um software e do método de desenvolvimento que é usado para se gerar um software livre. A prática comprovou o que a teoria dizia sobre os cuidados necessários na fase de evolução de software.

Notou-se a importância de que a documentação gerada refletisse a realidade do código atual, a importância de uma ferramenta de controle de versões para gerenciar e controlar as versões que forem geradas, e a utilidade de inspeções para um sistema em evolução, já que estas ajudam a verificar possíveis problemas e inconsistências nos artefatos gerados, agregando qualidade ao produto final.

No caso da evolução realizada no sistema do projeto, os seguintes tipos de manutenções foram realizados:

- Corretiva: foram realizadas mudanças para corrigir erros do sistema;

- Adaptativa: foram realizadas mudanças de forma a incorporar as necessidades relativas à adequação no ambiente de processamento. Como exemplo, tivemos a substituição do banco de dados, que foi alterado para o MySQL, e também a utilização da ferramenta de controle de versões CVS;
- Evolutiva: foram realizadas mudanças para aumentar a funcionalidade do software, já que alguns acréscimos de funcionalidade foram introduzidos;
- Preventiva: realizou-se mudanças visando a aumentar a manutenibilidade do sistema, de forma a torná-lo mais fácil de ser compreendido e alterado quando necessário. Por exemplo, os cenários foram documentados no próprio código fonte.

A evolução do C&L ocorreu usando-se o método de desenvolvimento de software livre, e esse fato colaborou muito para a melhor compreensão deste método por parte dos envolvidos, já que nenhum destes possuía conhecimento prático sobre o assunto. Durante o projeto todos os integrantes estavam altamente motivados e como já era esperado, o desenvolvimento ocorreu de forma bem dinâmica e prazerosa, mas durante o curso deste, aprendemos que apesar de não existir um processo rígido e controlado, é importante haver um controle de qualidade para projetos deste tipo que estão em constante evolução. Este controle deve ser feito em parte por uma documentação bem atualizada, o que nem sempre é realizado, pois muitos dos desenvolvedores de software livre olham apenas para o código e não se preocupam em atualizar os documentos pertinentes [Robotton03]. O próximo capítulo trará uma proposta para tentar mudar esse quadro, proposta esta que foi testada no desenvolvimento da ferramenta C&L.