

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**NODIPO: LEVANTAMENTO COLABORATIVO DE REQUISITOS
PARA O DESENVOLVIMENTO DE SOFTWARE LIVRE**

YURI GOMES CARDENAS

Florianópolis/SC

2010/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

NODIPO: LEVANTAMENTO COLABORATIVO DE REQUISITOS PARA O
DESENVOLVIMENTO DE SOFTWARE LIVRE

YURI GOMES CARDENAS

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação.
Professor Orientador: José Eduardo De Lucca.

Florianópolis/SC
2010/1

Dedico este trabalho a Lauro N. T. Cardenas.

AGRADECIMENTOS

Dos que são diretamente relacionados à elaboração deste trabalho, agradeço aos membros da banca e ao meu orientador. As suas considerações, sugestões e correções melhoraram muito a qualidade do trabalho. Fizeram-me refletir sobre cada linha, o que fez da elaboração do mesmo um belo aprendizado. Agradecimento extra ao Wagner e ao De Lucca, por sentarem horas comigo para debatermos o conteúdo e a forma (e pela paciência com as minhas insistências...). Agradeço também à minha querida Cris, que me ajudou muito no desenvolvimento deste trabalho e cuja monografia me salvou várias vezes quando tive dúvidas sobre a formatação.

Às bases que – material, mental e emocionalmente – me possibilitaram cursar a graduação, um agradecimento do tamanho do mundo. Minha sorte é tamanha que, na mesma proporção em que me vieram desafios no caminho, fui amparado por um grupo de protetores e amigos:

Agradeço ao meu avô Vilmar, que me deu apoio significativo nas épocas mais duras, ensinou-me muito sobre qual caminho trilhar nesta vida e me mostrou o valor do trabalho e da fortitude – estes dois, pelo de exemplo de vida.

Agradeço a Nildes e Nilson, pela amizade, por apostarem na minha capacidade e pelos inúmeros apoios que me deram ao longo dos anos.

À Olga, Carlinhos e André Luiz, agradeço pelo grande apoio sem o qual eu não teria tido a oportunidade de realizar este trabalho.

Agradeço ao amigo e orientador De Lucca, com quem tive e tenho o privilégio de trabalhar e conviver. Poucos entendem tão bem quanto ele a relação entre ser professor e a importância de se compartilhar conhecimento.

Agradeço ao meu tio Mauro, pois, definitivamente, sem o grande suporte dele eu certamente não estaria escrevendo essa nota de agradecimento neste trabalho de conclusão do curso de Bacharelado em Sistemas de Informação.

Agradeço à minha amável Cris, pelo carinho, paciência, compreensão, apoio, conforto... teu amor incondicional é admirável e me impressiona.

À minha mãe, Vânia, agradeço pelo suporte que dura simplesmente minha vida toda(!), pelo exemplo de resistência, de doação e de bondade e pela estrutura emocional que construí, com a criação que tive, que me preparou para enfrentar as dificuldades todas.

A todos vocês, muito obrigado.

SUMÁRIO

| | |
|--|-----------|
| LISTA DE ILUSTRAÇÕES..... | 7 |
| RESUMO..... | 8 |
| 1 INTRODUÇÃO..... | 3 |
| 1.1 OBJETIVOS..... | 4 |
| 1.2 ESTRUTURA DO TRABALHO..... | 5 |
| 2 REVISÃO DE LITERATURA..... | 6 |
| 2.1 SOFTWARE LIVRE..... | 6 |
| 2.2 ESCASSEZ DE SOFTWARES LIVRES EM DETERMINADOS SETORES..... | 14 |
| 2.3 ENGENHARIA DE REQUISITOS EM CONTEXTOS DISTRIBUÍDOS..... | 15 |
| 2.4 ENGENHARIA DE REQUISITOS E ENGENHARIA SOCIAL DE REQUISITOS..... | 17 |
| 2.5 A MODELAGEM CONTÍNUA INERENTE ÀS COMUNIDADES DE FLOSS..... | 20 |
| 2.6 MODELAGEM SOCIOTÉCNICA DE SOFTWARE..... | 21 |
| 2.7 WEB 2.0..... | 22 |
| 2.8 FERRAMENTAS RELACIONADAS..... | 23 |
| 3 PROJETO..... | 25 |
| 3.1 HISTÓRICO – O PROJETO VIA DIGITAL..... | 25 |
| 3.2 A PROPOSTA DO NODIPO..... | 26 |
| 3.3 DESCRIÇÃO DO PROCESSO DE LEVANTAMENTO DE REQUISITOS COM O NODIPO..... | 27 |
| 3.4 MODELAGEM..... | 28 |
| 3.5 PADRÃO DE PROJETO DATA ACCESS OBJECT – DAO..... | 31 |
| 4 DESENVOLVIMENTO..... | 32 |
| 4.1 ESPECIFICAÇÕES TÉCNICAS..... | 32 |
| 4.2 DESCRIÇÃO DAS INTERFACES GRÁFICAS..... | 32 |
| 4.3 ESTÁGIO ATUAL DO DESENVOLVIMENTO..... | 40 |
| 4.4 AVALIAÇÃO DA FERRAMENTA..... | 40 |
| 4.5 FUNCIONALIDADES PARA A VERSÃO STANDALONE..... | 43 |
| 5 CONCLUSÃO..... | 44 |
| 6 TRABALHOS FUTUROS..... | 46 |
| 6.1 FUNCIONALIDADES RELEVANTES A IMPLEMENTAR..... | 46 |
| 6.2 DIRECIONAMENTOS TEÓRICOS..... | 47 |

| | |
|--|-----------|
| REFERÊNCIAS..... | 48 |
| APÊNDICE A – DIAGRAMA DE CLASSES..... | 52 |
| APÊNDICE B – CATEGORIAS E SUBCATEGORIAS DE SOFTWARE..... | 53 |
| APÊNDICE C – O PROCESSO COLABORATIVO DE ELICITAÇÃO..... | 54 |
| APÊNDICE D – EXIBIÇÃO DE FUNCIONALIDADE..... | 55 |
| ANEXO A – SOBRE AS FERRAMENTAS SIMILARES..... | 56 |
| A.1 ENGENHARIA DE REQUISITOS BASEADA EM SEMÂNTICA..... | 56 |
| A.2 COOPERATIVE REQUIREMENTS ENGINEERING SUPPORT TOOL – CRETA..... | 59 |
| A.3 CODIPSE-REQ..... | 61 |
| ANEXO B – DIAGRAMAS DO SOFTWIKI..... | 63 |

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Ilustração 1: Tela inicial – exibição de lista de Sistemas cadastrados..... | 33 |
| Ilustração 2: Tela inicial – exibição de lista de funcionalidade de determinado Sistema..... | 33 |
| Ilustração 3: Tela de cadastro de funcionalidade..... | 34 |
| Ilustração 4: Tela de exibição de funcionalidade com exibição do debate..... | 36 |
| Ilustração 5: Quadro de concordância em relação ao levantamento de um requisito..... | 37 |
| Ilustração 6: Exibição das listas de alteração de status e de prioridade..... | 37 |
| Ilustração 7: Janela de cadastro de comentário..... | 38 |
| Ilustração 8: Janela de escolha das opções para a geração da versão de impressão..... | 39 |
| Ilustração 9: Versão para impressão de uma funcionalidade..... | 39 |
| Ilustração 10: Diagrama de Classes Conceitual em UML do Nodipo..... | 52 |
| Ilustração 11: Processo colaborativo de elicitação de requisitos com o Nodipo..... | 54 |
| Ilustração 12: Tela de exibição de funcionalidade com exibição do questionário..... | 55 |
| Ilustração 13: Arquitetura do SoftWiki..... | 63 |
| Ilustração 14: A ontologia para Engenharia de Requisitos SoftWiki. SoftWiki Ontology for Requirements Engineering – SWORE..... | 63 |

RESUMO

O desenvolvimento distribuído de software é uma realidade mundial. Muitos estudos são feitos com o objetivo de se encontrarem soluções para os problemas inerentes aos ambientes distribuídos. Nesse contexto, a área da Engenharia de Requisitos busca formas eficazes de se especificarem requisitos com participantes fisicamente distantes.

Em relação aos participantes, existem áreas de estudo que consideram vital a atuação de usuários-finais no desenvolvimento de software. Essa classe de usuários por vezes são os especialistas do domínio de aplicação do software e demonstram resistência à participação nesse processo. Para atenuar essa resistência, torna-se interessante o uso de instrumentos que mobilizem e estimulem o envolvimento desses usuários-finais e apoiem a colaboração de todos os envolvidos no processo de desenvolvimento de software. Os softwares sociais são um exemplo desses instrumentos que, além disso, viabilizam a colaboração em contextos distribuídos.

Uma vez constatadas as qualidades técnicas e os benefícios que se podem obter com o uso de Software Livre, percebe-se, entretanto, que em algumas áreas de aplicação existem poucos ou até mesmo inexistem Softwares Livres. No sentido de mudar esse quadro, o projeto Via Digital propôs em 2007 a ferramenta Nodipo, que visa a aproximar os diversos atores do processo de levantamento de requisitos distribuído – particularmente desenvolvedores de Software Livre e usuários-finais – e com a qual se possa colaborativamente criar especificações de requisitos de qualidade para o desenvolvimento de Software Livre.

Dada a relevância do tema, este trabalho aprofunda os fundamentos teóricos acerca da ferramenta, passando também pelos elementos que a compõem, bem como pelo desenvolvimento propriamente dito. Por fim, a partir de uma avaliação da ferramenta e do estudo de ferramentas similares, sugerem-se diversas alterações ao Nodipo no sentido de melhorar suas capacidades de atuar como meio integrador de colaboração para a especificação de requisitos de Softwares Livres.

Palavras-chave: Engenharia Social de Requisitos. Modelagem Contínua de Software. Levantamento de requisitos. Colaboração. Software Livre.

1 INTRODUÇÃO

O desenvolvimento de software de forma distribuída – no qual os envolvidos estão geograficamente distantes – vem sendo adotado por diversas empresas de desenvolvimento, devido a fatores como redução de custos (HERBSLEB; MOITRA, 2001 apud LOPES; AUDY, 2003). A Engenharia de Software, disciplina voltada a uma abordagem sistemática e quantificável para o processo de desenvolvimento, operação e manutenção de software (PRESSMAN, 2006), estabelece o levantamento de requisitos como uma etapa necessária desse processo. Nessa etapa obtêm-se dos usuários-finais, e de outros possíveis envolvidos, os objetivos do sistema, o que deve ser realizado, como o sistema se encaixa na realidade do negócio e como ele deve ser utilizado no dia-a-dia (PRESSMAN, 2006).

O levantamento de requisitos é objeto de estudo de uma subdisciplina da Engenharia de Software chamada Engenharia de Requisitos, que se dedica à análise, documentação e evolução contínua das necessidades do usuário bem como à especificação do comportamento externo de um sistema que satisfaça àquelas necessidades (PRESSMAN, 2006).

Surge ainda, no mesmo sentido, o conceito de Engenharia Social de Requisitos, na qual a Engenharia de Requisitos é encarada como uma atividade social em que a utilização dos chamados softwares sociais é interessante – pelo caráter naturalmente colaborativo dos mesmos – para o levantamento colaborativo e distribuído de requisitos de software (LOHMANN et al., 2009).

Nas comunidades de desenvolvimento de FLOSS¹ – Free/Libre/Open Source Software, ou simplesmente Software Livre –, nas quais o desenvolvimento distribuído e colaborativo de software² é bastante presente (RAYMOND, 1999), muito dificilmente se encontram bem definidas as etapas de desenvolvimento de software descritas pela Engenharia de Software tradicional (GASSER et al., 2003), em que os processos possuem uma estrutura padrão e a etapa de levantamento de requisitos adota técnicas como a prototipagem, a aplicação de entrevistas e o uso de questionários e pesquisas (NUSEIBEH; EASTERBROOK, 2000). Por outro lado, há estudos que levam em conta as dinâmicas de desenvolvimento

1 Resumidamente, por FLOSS entende-se o software cuja licença de uso permite a utilização, a cópia, a modificação e a redistribuição do mesmo pelo usuário que o recebe – sendo o acesso ao código-fonte do software pré-requisito para as duas últimas condições citadas – (FREE SOFTWARE FOUNDATION). Apesar de haver pequenas diferenças entre os conceitos de Software Livre e Software de Código-aberto – Free Software e Open Source Software, em inglês, respectivamente – as mesmas não serão abordadas neste trabalho pois a distinção entre os conceitos não afeta os temas nele discutidos.

2 Também conhecido como modelo de desenvolvimento de software Bazar (RAYMOND, 1999).

colaborativo das comunidades de Software Livre, prevendo, por exemplo, a participação de pessoas não treinadas – como os usuários-finais – no processo de levantamento distribuído de requisitos, como é o caso da Modelagem Sociotécnica de Sistemas.

Ainda que o uso de FLOSS possa desempenhar um papel importante na sociedade (HALL, 2008), Savi e De Lucca (2007) afirmam que há escassez de Softwares Livres em determinados domínios que não são das áreas de interesse típicas das comunidades de FLOSS, como sistemas operacionais, ferramentas de desenvolvimento de software e aplicativos de escritório. Tal fato se dá porque os desenvolvedores de Software Livre não conhecem suficientemente os requisitos, as demandas e as regras de negócio desses domínios de aplicação para poderem contribuir significativamente (SAVI; DE LUCCA, 2007). Dessa forma, torna-se interessante o uso de ferramentas que aproximem desenvolvedores de Softwares Livres e usuários-finais de modo a viabilizar a discussão e o compartilhamento de conhecimento entre eles para que sejam criadas especificações de requisitos dos softwares necessários a esses domínios de aplicação.

Nesse contexto, neste trabalho se procura discutir mais profundamente os diversos aspectos que afetam e dificultam o levantamento colaborativo e distribuído de requisitos, bem como apresentar e discutir a ferramenta de levantamento colaborativo de requisitos para Software Livre Nodipo, que pode ser útil nesse tipo de ambiente.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Discutir aspectos teóricos e práticos do processo de levantamento de requisitos para Software Livre a partir da concepção e do desenvolvimento de uma ferramenta colaborativa, o Nodipo.

1.1.2 Objetivos específicos

- a) Aprofundar os fundamentos teóricos relativos à Engenharia Social de Requisitos e áreas de estudo correlatas;
- b) Detalhar a modelagem e as estruturas da implementação do Nodipo;
- c) Pesquisar soluções existentes de levantamento colaborativo de requisitos;
- d) Avaliar a utilização da ferramenta com a inserção de requisitos no Nodipo;
- e) Propor melhorias de acordo com o aprofundamento teórico, com as pesquisas de soluções similares e com a avaliação da ferramenta.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte maneira: apresentado este capítulo introdutório, trata-se, no capítulo 2 – Revisão de literatura –, das fundamentações teóricas que circundam, contextualizam e baseiam a ferramenta Nodipo. Esse capítulo é finalizado com uma seção de informações sobre ferramentas similares ao Nodipo que apresenta uma tabela comparativa de funcionalidades e características dessas ferramentas.

Em seguida, no terceiro capítulo – Projeto –, descreve-se o histórico de desenvolvimento, o contexto de surgimento e a proposta do Nodipo, um possível processo de levantamento de requisitos ao utilizar-se o Nodipo, assim como detalhes da modelagem e da estrutura do mesmo.

O capítulo 4 – Desenvolvimento – detalha as tecnologias utilizadas para o desenvolvimento do Nodipo e descreve as interfaces gráficas e o estágio atual do desenvolvimento. Além disso, esse capítulo trata da experiência de avaliação de usabilidade do Nodipo e seus resultados.

O quinto capítulo, de conclusão, traz as considerações finais sobre o assunto, com um apanhado geral do tema tratado neste trabalho e projeções das implicações da utilização de uma ferramenta como o Nodipo no cenário nacional.

Por fim, o sexto e último capítulo traz sugestões de funcionalidades a serem implementadas no Nodipo e possíveis direcionamentos de estudos futuros relacionados à ferramenta.

2 REVISÃO DE LITERATURA

2.1 SOFTWARE LIVRE

Dado o fato de ser pré-requisito o acesso ao código-fonte para que um software seja considerado FLOSS, em relação às implicações técnicas, Brown et al. (2002 apud GASSER et al., 2003) afirmam que o Software Livre é cada vez mais citado como uma das abordagens mais promissoras e mundialmente demonstradas para o reuso de software.

Por reuso de software compreende-se basicamente utilizar artefatos de software existentes no desenvolvimento de um novo software (KRUEGER, 1992); enquanto que artefatos de software não se limitam apenas a fragmentos de código-fonte, mas também a estruturas de modelagem, estruturas de implementação a nível de módulo, especificações e documentações, por exemplo (FREEMAN, 1983 apud KRUEGER, 1992). Ademais, o Grupo de Trabalho em Reuso de Software da NASA – Earth Science Data Systems Software Reuse Working Group, ESDSSRWG – estabelece que o objetivo do reuso de software é diminuir custo, tempo, esforço e risco; e aumentar produtividade, qualidade, desempenho e interoperabilidade (ESDSSR WORKING GROUP, 2008b).

Considerando a definição e as vantagens que se podem obter com o reuso de software, a afirmação anterior de Brown et al. (2002 apud GASSER et al., 2003) ajuda a reforçar a ideia de que o Software Livre pode tornar a modelagem e o desenvolvimento de programas complexos mais rápidos, melhores e mais baratos, conforme afirma Scacchi (2004b).

O ESDSSRWG ressalta também que os mecanismos do modelo de licenciamento de software proprietário – oposto ao modelo de licenciamento de Software Livre – são uma potencial barreira para o reuso de software, já que esse tipo de licenciamento normalmente requer que sejam negociados termos e condições junto ao detentor da propriedade intelectual a cada vez que se deseja reusar um artefato – processo que tende a ser desgastante e trabalhoso (ESDSSR WORKING GROUP, 2008a).

Em contrapartida, o Grupo de Trabalho em Reuso de Software argumenta que a utilização de licenças de Software Livre ajudará a simplificar a disseminação de software entre a comunidade científica e a promover um modelo mais colaborativo de desenvolvimento para softwares científicos. O Grupo entende, inclusive, que o modelo colaborativo de

desenvolvimento das comunidades de Software Livre é particularmente interessante à comunidade científica, uma vez que os avanços científicos são muitas vezes o resultado de esforços combinados de pesquisadores de muitas organizações diferentes (ESDSSR WORKING GROUP, 2008a).

Somadas a essas vantagens técnicas, é levantada a seguir uma série de questões relativas às implicações da utilização de Softwares Livres na sociedade como um todo.

2.1.1 Uso de Software Livre pela administração pública

Hall (2008) defende que os governos, pelo fato de lidarem com verba pública, precisam pensar a longo prazo no que diz respeito à tecnologia. Ele alega que um governo precisa, por exemplo, considerar as consequências para o seu país caso um software venha a se tornar indisponível. Se uma empresa fornecedora de software fechar, será necessário considerar se o governo terá acesso ao código-fonte do software. E se tiver acesso, se será possível encontrar profissionais que trabalhem com a tecnologia. Além disso, Hall (2008) salienta que é necessário que os governos considerem o custo da atualização de softwares proprietários – que pode, além disso, incluir custos adicionais com novos *hardwares* – e a possibilidade de um embargo econômico ser posto em prática pelo país da empresa que fornece o software – de forma que governo pode deixar de ter suporte disponível para o mesmo.

Hall (2008) considera que ao utilizar Softwares Livre os governos têm controle e propriedade sobre o software que utilizam e que, dessa forma, “os governos podem fazer alterações de acordo com suas necessidades, em vez de alterar suas necessidades de acordo com as possibilidades do software” (HALL, 2008).

Dentre as argumentações para uso de softwares proprietários pela administração pública, Hall (2008) descreve e contra-argumenta da seguinte maneira:

- a) Se o país for pequeno, é possível que não se encontrem no país todos os programadores necessários para desenvolverem todos os softwares necessários ao governo. Contudo, os governos têm a possibilidade de aliam-se às comunidades de FLOSS e a outros governos interessados em desenvolverem os softwares que ambos precisam, existindo, ainda sim, a possibilidade de o governo modificar o software como bem quiser;

- b) O software proprietário é fonte de renda para uma população, pois empresas vendem, instalam, integram, certificam profissionais e dão capacitação para determinado software. Entretanto, tudo isso se aplica ao Software Livre e ainda em maior amplitude. Tendo-se o código-fonte disponível, torna-se muito mais fácil a alteração e integração do software em uma solução melhor, perfeitamente adequada ao que se precisa;
- c) Em sua maioria, os *royalties* de software são inferiores a 25% do custo do mesmo e, portanto, insignificantes. Porém, não parecem exatamente insignificantes 25% na redução do custo de um produto, ainda mais considerando que esses 25% ocasionam saída de divisas do país se o software for de um país estrangeiro. Com o uso de Softwares Livres, esses 25% poderiam ser economizados ou investidos, por exemplo, em um programador local para adaptar o software às necessidades do governo. Ademais, esse programador pagaria impostos, consumiria e residiria no próprio país.

2.1.2 Ações do Governo Federal brasileiro

Dada a evolução tecnológica de amplitude mundial das últimas décadas, o Governo Federal brasileiro implementa, desde 1999, medidas que visam a alinhar o país a essa nova realidade (GOVERNO FEDERAL, 2005, p.41). O programa Sociedade da Informação – Decreto nº 3.294 de 15 de dezembro de 1999 – foi uma delas e com ele objetivou-se:

[...] integrar, coordenar e fomentar ações para a utilização de tecnologias de informação e comunicação, de forma a contribuir para a inclusão social de todos os brasileiros na nova sociedade e, ao mesmo tempo, contribuir para que a economia do país tenha condições de competir no mercado global. (GOVERNO FEDERAL, 2000, p. 10.)

Outra iniciativa relevante nesse sentido foi a criação do Comitê Executivo do Governo Eletrônico – Decreto de 18 de outubro de 2000 –, cuja função é formular políticas, estabelecer diretrizes, coordenar e articular as ações de implantação do Governo Eletrônico (PORTAL DO GOVERNO ELETRÔNICO DO BRASIL, 2010a), que é voltado à prestação de serviços e

informações ao cidadão. A definição menos genérica de Governo Eletrônico abrange: “a utilização das modernas tecnologias de informação e comunicação – TICs – para democratizar o acesso à informação, ampliar discussões e dinamizar a prestação de serviços públicos com foco na eficiência e efetividade das funções governamentais” (PORTAL DO GOVERNO ELETRÔNICO DO BRASIL, 2010b). Ainda, o Governo Eletrônico foi pensado como instrumento de transformação da sociedade brasileira, com o qual se estabelecem diretrizes e parâmetros para a criação de uma sociedade digital (GOVERNO FEDERAL, 2005).

Com o passar do tempo, novas necessidades foram identificadas nesse contexto, o que fez o Governo modificar as ações determinadas em 1999 e 2000 para ajustá-las ao novo cenário. Portanto, foram estabelecidos – via Decreto de 29 de outubro de 2003 – comitês técnicos específicos no âmbito do Comitê Executivo do Governo Eletrônico. São Comitês: de Implementação do Software Livre, de Inclusão Digital, de Integração de Sistemas, de Sistemas Legados e Licenças de Software, de Gestão de Sítios e Serviços Online, de Infra-Estrutura de Rede, de Governo para Governo – G2G –, de Gestão de Conhecimento e de Informação Estratégica (GOVERNO FEDERAL, 2005). Para tanto, o Decreto de 29 de outubro de 2003 estabeleceu que o Governo Eletrônico passaria a ser implementado segundo sete princípios:

[...] como referência geral para estruturar as estratégias de intervenção, adotadas como orientações para todas as ações de Governo Eletrônico, gestão do conhecimento e gestão da TI no governo federal:

- Promoção da cidadania como prioridade;
 - Indissociabilidade entre inclusão digital e o governo eletrônico;
 - Utilização do software livre como recurso estratégico;
 - Gestão do Conhecimento como instrumento estratégico de articulação e gestão das políticas públicas;
 - Racionalização dos recursos;
 - Adoção de políticas, normas e padrões comuns;
 - Integração com outros níveis de governo e com os demais poderes.
- (GOVERNO FEDERAL, 2005, p. 43)

Ademais, como será visto no decorrer desta seção, a utilização do Software Livre é relacionada e/ou pode apoiar outros dos 6 princípios listados como referência para as ações do Governo Eletrônico, tais como: inclusão digital, racionalização de recursos e uso de normas e padrões comuns.

2.1.2.1 Padrões de interoperabilidade de Governo Eletrônico – e-PING

Criada pelo Governo Federal, a e-PING trata-se de uma arquitetura que estabelece premissas, políticas e especificações técnicas que regulamentam a utilização das Tecnologias de Informação e de Comunicação na interoperabilidade de Serviços de Governo Eletrônico. Ela define as condições de interação do Governo Federal – Poder Executivo – com os demais Poderes e esferas de governo, bem como com a sociedade em geral (GOVERNO FEDERAL, 2005; GOVERNO FEDERAL, 2009).

Alinhada às ações estabelecidas para a implantação do Governo Eletrônico, a e-PING foi criada com o objetivo de se construírem mecanismos que favoreçam a eficiência da administração pública (GOVERNO FEDERAL, 2005).

Mais especificamente, o documento de referência da arquitetura e-PING define como escopo o intercâmbio de informações entre os sistemas do Governo Federal e as interações com:

- Cidadãos;
- Outros níveis de governo (estadual e municipal);
- Outros Poderes (Legislativo, Judiciário) e Ministério Público Federal;
- Organismos Internacionais;
- Governos de outros países;
- Empresas (no Brasil e no mundo);
- Terceiro Setor.

(GOVERNO FEDERAL, 2009, p. 7)

Ainda quanto ao escopo, o documento delimita a interoperabilidade como foco de atuação da arquitetura, de modo que a e-PING trata apenas de “especificações que forem relevantes para garantir a interconectividade de sistemas, integração de dados, acesso a serviço de governo eletrônico e gerenciamento de conteúdo” (GOVERNO FEDERAL, 2009, p. 8). Cabe salientar que em maio de 2004 foi lançada a primeira versão – versão 0 – do documento de referência da e-PING.

Dentre as políticas gerais que fundamentam as políticas e especificações técnicas de cada segmento de aplicação da e-PING, é válido ressaltar as seguintes:

a) Adoção preferencial de **padrões abertos**:

A e-PING define que, sempre que possível, serão adotados padrões abertos nas especificações técnicas. Padrões proprietários são aceitos, de forma transitória, mantendo-se as perspectivas de substituição assim que houver condições de

migração. Sem prejuízo dessas metas, serão respeitadas as situações em que haja necessidade de consideração de requisitos de segurança e integridade de informações. (GOVERNO FEDERAL, 2009, p. 9)

a) Priorização de uso de software público e/ou Software Livre:

A implementação dos padrões de interoperabilidade deve priorizar o uso de software público e/ou software livre, em conformidade com diretrizes do Comitê Executivo de Governo Eletrônico [...] (GOVERNO FEDERAL, 2009, p. 9)

Em relação à definição de **padrão aberto**, a União Europeia estabelece quatro requisitos para que um padrão seja considerado aberto (EUROPEAN UNION, 2004). São eles:

- a) O padrão foi adotado e será mantido por uma organização sem fins lucrativos e o desenvolvimento do mesmo ocorre em um processo aberto de decisões – tais como consenso ou decisão da maioria – acessível a todos os interessados;
- b) O padrão foi publicado e o documento de especificação do mesmo é acessível gratuitamente ou por uma taxa simbólica. Deve ser permitido a todos copiar, distribuir e usar o documento de especificação gratuitamente ou por taxa simbólica;
- c) Toda a propriedade intelectual – ou seja, patentes que porventura estejam presentes – do padrão, ou de partes dele, é concedida irrevogavelmente sem *royalties*.
- d) Não há restrições quanto ao reuso do padrão.

Ademais, além dos requisitos que caracterizam um padrão aberto, o mesmo documento da União Europeia sustenta que o Software Livre tende a usar e ajuda a definir padrões abertos e especificações publicamente acessíveis (EUROPEAN UNION, 2004). Ainda segundo o documento, os Softwares Livres são naturalmente especificações publicamente acessíveis e a disponibilidade do código-fonte promove debates abertos e democráticos em torno dessas especificações, o que torna os Softwares Livres e os padrões abertos mais robustos e interoperáveis (EUROPEAN UNION, 2004).

Então, pode-se perceber que, além de corroborar e fazer parte das determinações do Governo Eletrônico, o Software Livre está significativamente inserido na arquitetura e-PING.

2.1.2.2 Guia Livre

Dentro do contexto apresentado e colaborando com as orientações descritas na arquitetura e-PING, no final de 2004 o Governo Federal publicou a versão 1.0 do documento intitulado Guia Livre – Referência de Migração para Software Livre do Governo Federal (GOVERNO FEDERAL, 2005). Em suma, o documento traz instruções, diretivas – técnicas e de gestão –, passos para o planejamento, dicas, estudos de casos, exemplos específicos de migração de diversos softwares, entre outros conteúdos de auxílio na migração para Software Livre, bem como argumentações para que instituições públicas estabeleçam programas de migração. Dentre as argumentações, apontam-se:

- a) necessidade de adoção de padrões abertos para o Governo Eletrônico – e-Gov;
- b) nível de segurança proporcionado pelo Software Livre;
- c) eliminação de mudanças compulsórias que os modelos proprietários impõem periodicamente a seus usuários, em face da descontinuidade de suporte a versões ou soluções;
- d) independência tecnológica;
- e) desenvolvimento de conhecimento local;
- f) possibilidade de se fazer auditoria dos sistemas;
- g) independência de fornecedor único. (GOVERNO FEDERAL, 2005)

O documento salienta também que os benefícios acima listados – acrescidos ao fato de que as despesas com licenciamento não se aplicam ao Software Livre – resultam em economia progressiva aos que o utilizam; valores os quais podem ser reaplicados em investimentos na área de Tecnologia da Informação, por exemplo.

Por fim, a conclusão do Guia Livre sobre o benefício das medidas adotadas ao Estado e, conseqüentemente, à população:

Portanto, o Estado se beneficia diretamente com a adoção do Software Livre, tanto no aspecto de sua estruturação para atendimento às demandas sociais, como no seu papel de promover desenvolvimento. Desse modo, possibilitamos a integração das políticas de modernização administrativa, inclusão social baseadas na Tecnologia da Informação e no desenvolvimento industrial. (GOVERNO FEDERAL, 2005, p. 46.)

Considerando o cenário apresentado, percebe-se que as ações governamentais no sentido de adequar o Brasil ao então novo cenário digital deram atenção especial ao incentivo ao uso de Software Livre.

2.1.3 Plano de migração para Software Livre no Exército Brasileiro

O Exército Brasileiro – EB –, em conformidade com as políticas de incentivo à migração de Software Livre do país, estabeleceu um plano de migração para Software Livre nas suas Organizações Militares – OM.

Em fevereiro de 2007 foi lançada a 3ª edição do documento PLANO DE MIGRAÇÃO PARA SOFTWARE LIVRE NO EXÉRCITO BRASILEIRO, cuja finalidade é “regular a estratégia para a consolidação da implantação do Software Livre – SL – em todos os escalões do Exército Brasileiro” (EXÉRCITO BRASILEIRO, 2007).

São listados os seguintes objetivos no documento:

- a) Apresentar uma proposta de reformulação dos processos que envolvam a utilização e a aquisição de software no Exército Brasileiro.
- b) Propiciar uma substancial economia de custo de aquisição e manutenção de softwares.
- c) Incentivar a formação e consolidação de uma Comunidade Interna de Software Livre no EB, sob a égide do Núcleo de Estudos em Software Livre – NESOL –, com procedimentos e ferramentas de colaboração bem definidos.
- d) Restringir o crescimento do legado baseado em tecnologia proprietária.
- e) Priorizar a aquisição de hardware compatível às plataformas livres.
- f) Permitir o compartilhamento do conhecimento, fomentando a criação de uma Base Interna de Conhecimento em Software Livre, prioritariamente focada em soluções de problemas advindos da utilização das ferramentas de software recomendadas.
- g) Fomentar a criação de um “Banco de Talentos em Software Livre”, sob gerenciamento do NESOL, a fim de cadastrar as diversas capacidades e conhecimentos, na área de SL, dos integrantes do EB. (EXÉRCITO BRASILEIRO, 2007)

Além disso, nas considerações iniciais do documento, é frisado que a questão do software no Brasil está relacionada com a Política do Estado Brasileiro, a qual estabelece que:

Art. 2º A Política Nacional de Informática tem por objetivo a capacitação nacional nas atividades de informática, em proveito do **desenvolvimento social, cultural, político, tecnológico e econômico** da sociedade brasileira, atendidos os seguintes princípios:

I - **ação governamental** na orientação, coordenação e estímulo das atividades de informática;

[...]

IV - proibição à criação de situações monopolísticas, de direito ou de fato;

V - ajuste continuado do processo de informatização às peculiaridades da sociedade brasileira;

VI - **orientação de cunho político das atividades de informática**, que leve em conta a necessidade de preservar e aprimorar a identidade cultural do País, a natureza estratégica da informática e a influência desta no esforço desenvolvido pela Nação, para alcançar melhores estágios de bem-estar social;

VII - **direcionamento de todo o esforço nacional no setor**, visando ao atendimento dos programas prioritários do desenvolvimento econômico e social e ao fortalecimento do Poder Nacional, em seus diversos campos de expressão. (Lei nº 7.232, de 29 Out 1984 apud EXÉRCITO BRASILEIRO, 2007)

Os trechos destacados do texto da lei advêm do próprio plano de migração do Exército Brasileiro, o qual frisa inclusive que: “a adoção da solução livre, ou aberta, é considerada definitiva para todo o Exército Brasileiro” (EXÉRCITO BRASILEIRO, 2007).

Cabe ressaltar que ambos os documentos que tratam de migração – Guia Livre e o Plano de Migração do Exército Brasileiro – possuem instruções para que as migrações sejam cuidadosamente estudadas e executadas, de forma, por exemplo, a não prejudicar as atividades existentes executadas em plataformas proprietárias até então.

2.2 ESCASSEZ DE SOFTWARES LIVRES EM DETERMINADOS SETORES

Segundo Savi e De Lucca (2007), o Software Livre tem presença bastante significativa em alguns setores, nos quais as novas opções de software surgem a todo momento. Por outro lado, em outras áreas, a presença de opções livres é mais escassa e até inexistente.

Para a análise da presença de Softwares Livres em categorias de software, utilizou-se neste trabalho a taxonomia proposta por Savi e De Lucca (2007), reproduzida a seguir:

- a) **Software básico e de infraestrutura:** sistemas operacionais, redes/protocolos, bancos de dados, segurança, outros servidores, *drivers* de dispositivos,...
- b) **Software de produção:** que podemos decompor nos segmentos
 - de Uso geral: software de comunicação (clientes de e-mail, navegadores, clientes de mensagens instantâneas,...), software de escritório (editores de texto, planilhas, organizadores,...) e ferramentas associadas (antivírus, multimídia, compactação, visualizadores de documentos,...);
 - para Áreas específicas: subdividido em diversos setores, como softwares para engenharia, artes gráficas, educação, entretenimento, agronegócios e gestão de organizações. Aqui estaria também o setor de Ferramentas para desenvolvimento de software (compiladores e correlacionados, ferramentas de apoio e gestão de projetos de desenvolvimento...).
- c) **Software embarcado:** em geral, aplicável a uma arquitetura de equipamento específica.

Os autores afirmam que há claramente disponibilidade de soluções livres nas categorias de software básico e de infraestrutura – como sistemas operacionais, protocolos,

bancos de dados, segurança – e no setor de ferramentas para desenvolvimento de software – compiladores, gestão de projetos de desenvolvimento – e em outros setores específicos como comunicação – e-mail, navegadores, mensagens instantâneas – e ferramentas básicas de produtividade – editores de texto, planilhas, organizadores pessoais e de grupos. Sobre exemplos em outras áreas, Savi e De Lucca (2007) afirmam que: “existem mas de forma isolada e de acordo com esforços específicos, mas em nada comparáveis à presença, por exemplo, na categoria de softwares básico e de infraestrutura, que pode ser considerado o berço do desenvolvimento do Software Livre”.

De uma forma geral, por parte da comunidade de desenvolvedores de Softwares Livres, existe maior interesse em fazer parte de projetos de software básico – protocolos, *kernels*, ambientes gráficos, servidores variados, dentre outros –, de projetos com grande visibilidade, de softwares utilizados pelo grande público – do qual esses desenvolvedores também fazem parte –, tais como editores de texto e navegadores (SAVI; DE LUCCA, 2007). Os desenvolvedores têm bom conhecimento sobre esses softwares e conhecem bem os requisitos, as dificuldades, os recursos ausentes e os *roadmaps* dos mesmos (SAVI; DE LUCCA, 2007). A concepção de Scacchi (2002 apud SAVI; DE LUCCA, 2007) colabora com essa constatação, uma vez que o autor entende que os projetos de Software Livre geralmente contam com usuários que também são desenvolvedores, os quais implementam as funcionalidades que desejam, pessoal ou coletivamente, nos programas.

Por outro lado, os softwares aplicativos são muito específicos. Os autores afirmam que “a comunidade de desenvolvedores de Software Livre não conhecem suficientemente os requisitos, as demandas e regras de negócio para poder contribuir de forma significativa” (SAVI; DE LUCCA, 2007).

Portanto, em um domínio de aplicação com que os desenvolvedores não tenham contato e em que não há acesso a software que possa servir de exemplo, a ausência de conhecimento dos requisitos tende a prejudicar o desenvolvimento de Softwares Livres para atender àquela demanda, como confirmam Nakagawa et al. (2006).

2.3 ENGENHARIA DE REQUISITOS EM CONTEXTOS DISTRIBUÍDOS

Segundo Herbsleb e Moitra (2001 apud LOPES; AUDY, 2003), o mercado de desenvolvimento de software é afetado diretamente pela crescente globalização no ambiente

de negócios. Muitas empresas procuram vantagens competitivas na distribuição do desenvolvimento de software, como baixo custo, e muitas vezes encontram acentuadas as dificuldades já existentes nos processos de desenvolvimento de software não distribuído. Questões como diferenças culturais e de fusos horários têm de ser levadas em consideração ao se optar pelo desenvolvimento distribuído. Nesse contexto, considerando-se as dificuldades inerentes ao trabalho coletivo à distância, os processos relativos à Engenharia de Requisitos – que já são considerados como críticos (SOMMERVILLE; SAWYER, 1997 apud LOPES; AUDY, 2003) entre os processos de desenvolvimento de software – tendem a tornarem-se ainda mais difíceis.

Dentre as abordagens de Engenharia de Requisitos utilizadas para se avaliar a dimensão dos problemas relativos ao desenvolvimento distribuído de software, apontam-se da compilação de Lopes e Audy (2003) três delas, que são as que melhor têm relação com este trabalho.

A abordagem de Damian e Zowghi (2002 apud LOPES; AUDY, 2003) tem como foco principal o entendimento e descrição do impacto exercido pela distância na negociação de requisitos de software. Em uma publicação dos autores que trata do assunto, foi construído um modelo dos desafios encontrados que relaciona os quatro maiores problemas no desenvolvimento global de software – diversidade cultural, comunicação inadequada, gerência de conhecimento e diferença de tempo –, as dificuldades específicas decorrentes dos problemas e as atividades da Engenharia de Requisitos afetada por cada dificuldade específica encontrada.

Nos resultados dos estudos por essa abordagem, constatou-se que um dos mais importantes desafios encontrados foi o tratamento de requisitos conflitantes de grupos de múltiplas nacionalidades, com diferenças nas crenças e valores culturais. Valores culturais conflitantes causam impacto na priorização e negociação dos requisitos. Além disso, foram observados conflitos no entendimento dos requisitos, também atribuídos a diferenças culturais, entre outros problemas.

A abordagem de Lloyd et al. (2002 apud LOPES; AUDY, 2003) mostra um estudo empírico sobre como ferramentas de *groupware* podem ser utilizadas no auxílio da Engenharia de Requisitos em ambientes distribuídos. O estudo foi realizado com grupos de estudantes universitários, divididos nos papéis de clientes e engenheiros, construindo as especificações remotamente. Como resultado, obteve-se a efetividade das técnicas de especificação de requisitos de acordo com o perfil dos grupos e a conclusão de que a Engenharia de Requisitos em ambientes distribuídos é mais efetiva quando os *stakeholders*

participam ativamente nas atividades síncronas do processo de requisitos.

A abordagem de Mahemoff e Johnston (1998 apud LOPES; AUDY, 2003) apresenta um estudo sobre o relacionamento de internacionalização de software e Engenharia de Requisitos. Sendo os fatores culturais cruciais para o processo de internacionalização de software, no artigo é proposta uma categorização destes fatores, de forma a permitir a criação de um repositório central de informações culturais dos locais aos quais o software será desenvolvido para auxiliar no processo. Uma vez criado esse repositório, os desenvolvedores terão acesso a informações como: direção do texto, calendários utilizados, ordenação de palavras, data e hora, unidades de medição, moeda, entre outras.

Dessa maneira, uma vez bem especificada a internacionalização de um software, pode-se tornar o processo de localização do software relativamente direto, bastando a substituição de um arquivo de um idioma para outro, atualização de um banco de dados, ou outra pequena modificação onde quer que estejam os dados de localização.

Por fim, o modelo de referência proposto por Lopes e Audy (2003) lista as seguintes categorias que influenciam a Engenharia de Requisitos em ambientes distribuídos:

- a) **Comunicação:** idioma, fusos horários e meios de comunicação;
- b) **Cultura:** tanto a cultura nacional quanto organizacional podem influenciar na Engenharia de Requisitos, dificultando quanto mais diferentes forem;
- c) **Gestão do conhecimento:** papéis dos envolvidos, responsáveis por atividades e necessidade de informações culturais compartilhadas; e
- d) **Aspectos técnicos:** padrões de documentos, processos bem definidos e gerência de configuração – controle de versões e rastreabilidade.

Portanto, têm-se levantados pontos-chave que devem ser levados em consideração ao utilizarem-se ferramentas de levantamento colaborativo e distribuído de requisitos.

2.4 ENGENHARIA DE REQUISITOS E ENGENHARIA SOCIAL DE REQUISITOS

Reforçando o conceito já apresentado, Riechert et al. (2007) afirmam que Engenharia de Requisitos é a parcela da Engenharia de Software e do desenvolvimento de projetos que trata dos requisitos para um sistema a ser planejado. Trata-se de um processo iterativo e cooperativo que, segundo Pohl (1996, 2007 apud RIECHERT et al., 2007), objetiva atingir as seguintes metas:

- a) **Levantamento e compreensão de todos os requisitos relevantes:** deve-se obter os requisitos de participantes-chave ou desenvolver os requisitos junto a eles, pois normalmente os requisitos não são conhecidos previamente;
- b) **Concordância:** acordo quanto aos requisitos é um pré-requisito para a aceitação de um sistema. Obviamente, não é esperado que todos os participantes concordem com todos os requisitos explicitados. Portanto, a Engenharia de Requisitos deve identificar e resolver conflitos entre os participantes – por meio de discussões, votações ou pela decisão de outros participantes que não os envolvidos no conflito, por exemplo.
- c) **Documentação:** sistemas distintos requerem diferentes tipos de especificações de requisitos. Assim, sistemas críticos tal como o sistema de controle de uma aeronave requerem uma especificação de requisitos estrita e formal de maneira que o sistema não ponha em risco a vida de pessoas. Consequentemente, à Engenharia de Requisitos é essencial a documentação dos requisitos de acordo com as necessidades do domínio do sistema.

Uma abordagem comum para se atingirem as metas da Engenharia de Requisitos definidas por Pohl (1996, 2007 apud RIECHERT et al., 2007) é o uso de objetivos e cenários como passos intermediários para a obtenção de requisitos.

Os objetivos descrevem as intenções dos participantes com o sistema em questão (POHL, 1998 apud RIECHERT et al., 2007). Eles facilitam, por exemplo, a resolução de conflitos, uma vez que é mais simples chegar-se a um acordo quando se está em um nível abstrato, como o de objetivos, do que em um nível concreto, como o de requisitos.

Os cenários são descrições de exemplos de utilização do sistema planejado para se atingir um objetivo específico (POHL, 1998 apud RIECHERT et al., 2007). Uma das vantagens de se utilizar cenários é a possibilidade de elicitar-se um ou mais requisitos a partir de um cenário, dependendo de quão completa é a descrição do mesmo.

Objetivos e cenários juntos oferecem boa estrutura para a derivação de requisitos bastante detalhados – tanto funcionais como não funcionais – para os sistemas em que se está trabalhando (POHL, 1998, 2007 apud RIECHERT et al., 2007).

Lohmann et al. (NIKULA et al., 2000; DUAN; CLELAND-HUANG, 2006 apud LOHMANN et al., 2009) concluem que a Engenharia de Requisitos foi modelada, a princípio, para dar suporte a um grupo pequeno de especialistas na captura, estruturação, desenvolvimento e gerenciamento de requisitos de software. Os autores afirmam também que a participação direta de grupos maiores – dos quais façam parte usuários-finais, por exemplo –

normalmente não acontece, pois as ferramentas existentes são muito complexas para serem utilizadas por pessoas não treinadas que têm pouca experiência com Engenharia de Requisitos.

Hammouda et al. (2008 apud LOHMANN et al., 2009) definem Engenharia Social de Software como a aplicação de processos, métodos e ferramentas que possibilitem a criação, o gerenciamento, a implantação e o uso de software por comunidades em ambientes *online*. Dessa forma, considera-se que a Engenharia Social de Requisitos está para a Engenharia Social de Software assim como a Engenharia de Requisitos está para a Engenharia de Software.

Dentro da concepção de que softwares para Engenharia de Requisitos deveriam conter funcionalidades que incitem a colaboração, ou seja, dar suporte à Engenharia Social de Requisitos, Lohmann et al. (2009) ressaltam que o grande sucesso atual de aplicações Web focadas em comunidades deve estimular a agregação das funcionalidades colaborativas dessas aplicações em ambientes de Engenharia de Requisitos. Também apelidadas como software social, essas aplicações Web focadas em comunidades possuem as seguintes características (HIPPIER, 2006 apud LOHMANN et al. 2009):

- a) simplicidade;
- b) colaboração ágil;
- c) auto-organização;
- d) *feedback* social;
- e) transparência;
- f) estruturas emergentes;
- g) são orientadas aos usuários e às comunidades.

Essas características, conforme Lohmann et al. (2009), são fundamentais a programas de Engenharia de Requisitos que dão suporte à Engenharia Social de Requisitos. Ainda segundo esses autores, a aplicação dos conceitos de software social em uma aplicação Web pode encorajar participantes não convencionais a fazerem parte do processo de modelagem de software. Adicionalmente, Lohmann et al. (2008) afirmam que, de uma forma geral, a interface de soluções para colaboração via Web precisa ser fácil de usar para que se tenha grande participação dos atores envolvidos.

Scacchi (2004a), por sua vez, alerta que a participação de usuários não treinados na modelagem de software pode ser frustrante. Contudo, o autor já apontava em 2004, como tendência de estudo na área, as **técnicas de participação contínua dos usuários**, o que tange inclusive às características da modelagem de software contínua praticada pelas comunidades

de FLOSS estudadas por Gasser et al. (2003).

Por fim, percebe-se que há artifícios que podem viabilizar a participação de usuários não treinados no processo de levantamento colaborativo de requisitos.

2.5 A MODELAGEM CONTÍNUA INERENTE ÀS COMUNIDADES DE FLOSS

No sentido de entender a dinâmica de desenvolvimento de FLOSS, o estudo de Gasser et al. (2003) levantou algumas tendências relativas a esse tipo de desenvolvimento. São elas:

- a) As especificações das funcionalidades, da usabilidade e da estrutura do software não são conhecidas quando a modelagem e a publicação do mesmo são feitas. Ao invés disso, outras ferramentas computacionais capturam e rastreiam as preferências dos usuários das comunidades, o que forma um ciclo contínuo de inovação. Especificações formais ou processos de modelagem formais quase nunca são encontrados. O próprio código-fonte possui, em comentários, diversos traços de discussões e debates, constituindo, assim, uma espécie de especificação;
- b) Os requisitos são elicitados, analisados, especificados, validados e gerenciados através do que se pode chamar de um conjunto de informalismos de desenvolvimento de software (SCACCHI, 2002) no lugar de processos e especificações formais. Esse conjunto é constituído de ferramentas de compartilhamento de conhecimento semiestruturadas e relativamente informais, representações e práticas.

Além disso, Gasser et al. (2003) consideram ser surpreendente o fato de o processo de modelagem de FLOSS ligar diversos processos interativos, contínuos, abertos e coletivos – o que poderia facilmente tornar-se caótico – e ter como resultado programas eficientes, estáveis e úteis.

Disso, ao compreender-se a modelagem contínua de projetos de Software Livre que definem Gasser et al. (2003), vê-se que as comunidades de Software Livre de maneira geral não realizam as tarefas de levantamento, análise e especificação de requisitos como a Engenharia de Requisitos apresenta tradicionalmente, mas contínua, coletiva e colaborativamente.

Portanto, uma vez que se tem melhor compreensão do complexo funcionamento das comunidades de FLOSS, pode-se ter melhor noção de que elementos deve ser constituída uma ferramenta que pretende capturar o conhecimento de diversos atores – desenvolvedores, usuários-finais, colaboradores – dentro do processo de levantamento colaborativo de requisitos.

2.6 MODELAGEM SOCIOTÉCNICA DE SOFTWARE

Segundo Scacchi (2004a), *Social-Technical Design*, ou Modelagem Sociotécnica, está intimamente ligada à defesa da direta participação dos usuários-finais no processo de modelagem de sistemas de informação³. Segundo o autor, esse tipo de abordagem opõe-se aos métodos de modelagem da Engenharia de Software tradicional, os quais focam a atenção exclusiva ou principalmente nas atividades dos engenheiros de software, que utilizam ferramentas computacionais de modelagem e notações para obter e formalizar os resultados deste tipo de processo.

Historicamente, credita-se ao Tavistock Institute for Human Relations, de Londres, a origem do conceito e da prática de sistemas sociotécnicos, nos anos de 1940. Os esforços do instituto eram direcionados à modelagem de sistemas de trabalho em fábricas, escritórios e a sistemas de manufatura não computacionais. Modelagem de sistemas sociotécnicos, psicologia social e ecologia social eram os três principais focos do instituto no que tange à melhora das relações entre as pessoas vistas como desumanizadas pela sociedade industrial moderna. Nos anos de 1970, o instituto Tavistock começou a focar a atenção em modelagem de sistemas computacionais, para a utilização de sistemas sociotécnicos em ambientes organizacionais (SCACCHI, 2004a).

As contribuições do instituto Tavistock e de outras iniciativas de estudo de sistemas sociotécnicos são hoje a inspiração para estudantes e analistas de sistemas de informação contemporâneos, mas os conceitos e práticas para a modelagem sociotécnica evoluíram: os estudos acadêmicos atuais sobre o assunto são em sua maioria sobre resultados empíricos, com pesquisas, que fazem os trabalhos acadêmicos apontarem para o desenvolvimento de boas práticas de modelagem de sistemas sociotécnicos (SCACCHI, 2004a).

³ O autor se refere sempre a sistemas de informação. Apesar de um sistema de informação não ser necessariamente sinônimo de software, pode-se considerá-los sinônimos para o escopo deste trabalho.

Segundo Scacchi (2004a), comparativamente, a abordagem clássica para o envolvimento participativo na modelagem dizia pouco sobre quais usuários devem participar ativamente do processo modelagem. Se os usuários não forem treinados, ou não tiverem tido contato com modelagem de sistemas de informação, a participação deles pode ser frustrante. Dessa forma, os estudos na área foram reestruturados, passando-se à análise de participação e mapeamento das redes de interação sociotécnica na modelagem de sistemas de informação (BEYER; HOLTZBLATT, 1997 apud SCACCHI, 2004a). Sendo assim, segundo Scacchi (2004a), as pesquisas sobre o assunto tendem aos seguintes temas:

- a) Redes de interação sociotécnica das pessoas, recursos, políticas organizacionais e regras institucionais que circundam sistemas de informação, e como acontece as interações e a interface entre esses elementos;
- b) Técnicas de participação contínua dos usuários, necessárias para determinar quais dos requisitos existentes foram modificados, dada à contínua evolução dos requisitos de um sistema de informação;
- c) Formas de representação de sistemas sociotécnicos e de redes de interação sociotécnicas: como representá-los, visualizá-los, mapeá-los e relacioná-los;
- d) Distanciamento de abordagens prescritivas e aproximação cada vez maior de práticas de trabalhos coletivos e concretos que possam facilmente ser adotados e utilizados – sendo talvez o principal exemplo disto, segundo o autor, as práticas de desenvolvimento de software das comunidades e de projetos de FLOSS; e
- e) Uso crescente de métodos etnográficos e técnicas de modelagem contextualizadas para estudar como as pessoas cumprem suas funções no contexto organizacional utilizando tecnologia da informação, pessoas, recursos e circunstâncias que têm ao alcance.

Além disso, há uma necessidade básica de se descobrirem formas e meios de fazer desenvolvedores transformarem-se em e/ou compreenderem melhor os usuários e vice-versa, de maneira a prover suporte ao esforço coletivo deles de continuamente modelar e remodelar sistemas de informação.

2.7 WEB 2.0

Basicamente, neste trabalho o termo Web 2.0 refere-se à mudança ocorrida na World

Wide Web nos últimos anos, em que esta deixou de ser um veículo de publicação e tornou-se uma plataforma colaborativa, em que não é necessário conhecimento de programação para publicar conteúdo, da qual qualquer pessoa pode fazer parte, em uma contínua conversação em que o conhecimento é descoberto e construído (FREEDMAN, 2006).

Essa mudança ocorreu e ocorre através de ferramentas como *blogs*, *wikis*, redes sociais, entre outros. Dentre essas ferramentas, muitas permitem a interação assíncrona, ou seja, que cada pessoa possa participar na hora em que lhe convier, havendo o histórico da conversação disponível a todos (FREEDMAN, 2006).

2.8 FERRAMENTAS RELACIONADAS

Apresenta-se, a seguir, uma tabela que resume e compara as características das ferramentas similares ao Nodipo estudadas neste trabalho. O Anexo A apresenta textos descritivos sobre as mesmas. A análise subsequente é feita a partir dessas informações.

O caractere **X** indica a presença do recurso ou característica na ferramenta, o caractere **P** indica presença parcial e a notação N/D indica informação não disponível. A não presença de quaisquer um desses caracteres indica ausência do recurso/característica.

| Recurso/característica | Nodipo | SoftWiki | CRETA | CodipseReq |
|--|--------|----------|-------|------------|
| Acompanhamento da evolução do processo de levantamento de um requisito | X | N/D | X | X |
| Espaço para debate sobre um requisito | X | X | X | X |
| Priorização de requisitos | X | X | N/D | X |
| Busca por requisitos | | X | N/D | X |
| Histórico de alterações sobre as informações sobre os requisitos | P | X | N/D | |
| Histórico de versões da especificação de um requisito | X | X | N/D | |
| Estrutura semântica para formalização e pós-processamento dos requisitos | | X | | |
| Foco em dar suporte à Engenharia Social de Requisitos | X | X | | |
| Foco em dar suporte à Engenharia de Requisitos tradicional | | | X | X |
| Utilização de elementos da Web 2.0 | X | X | | |

Tabela 1 – recursos e características das ferramentas estudadas

Pode-se notar que as ferramentas CRETA e CODIPSE-Req foram modeladas de modo a apoiar Engenharia de Requisito em sua forma tradicional. Por exemplo, a arquitetura da ferramenta CRETA conta com papéis de usuários bem definidos – gerentes de projeto, patrocinadores, engenheiros de requisitos e os especialistas de domínio –, nos moldes da Engenharia de Requisitos tradicional. O projeto SoftWiki, por outro lado, possui características mais próximas à proposta do Nodipo, por apresentar uma ferramenta orientada ao usuário (LOHMANN et al., 2008a), ou seja, por dar ênfase ao apoio à Engenharia Social de Requisitos (LOHMANN et al., 2009).

Ainda que o CODIPSE-Req apoie a Engenharia de Requisitos tradicional com ênfase mais forte no caráter distribuído do que a ferramenta CRETA, nele são utilizadas terminologias e elementos técnicos e específicos da Engenharia de Requisitos, tais como pré e pós-condições, fluxo principal e fluxo alternativo de casos de uso. Outro exemplo disso é a funcionalidade de agendamento de reuniões, que é a representação virtual do que acontece na realidade nos processos da Engenharia de Requisitos tradicional. Tais aspectos podem, inclusive, aumentar a resistência de participação por parte de usuários especialistas de domínio não familiarizados com esses termos e elementos, o que também torna o CODIPSE-Req conceitualmente mais distante do Nodipo. Diferentemente da ferramenta SoftWiki, que possui uma série de características semelhantes.

É interessante observar que os primeiros estudos listados neste trabalho relativos ao SoftWiki são contemporâneos à concepção e ao início do desenvolvimento do Nodipo – ambos em 2007. O foco na Engenharia Social de Requisitos (LOHMANN et al., 2009) também reforça a proximidade entre as duas ferramentas. Não é por acaso que são identificadas como dificuldades e problemas das etapas de desenvolvimento de software do projeto SoftWiki tópicos similares aos levantados neste trabalho (THE SOFTWIKI PROJECT, 2010) e que o conteúdo resultante de ambas ferramentas são pensados como insumo para outras ferramentas. Em suma, as similaridades entre o SoftWiki e Nodipo passam por embasamento teórico, necessidades, funcionalidades e propósitos.

3 PROJETO

3.1 HISTÓRICO – O PROJETO VIA DIGITAL

A ferramenta a ser apresentada neste trabalho foi pensada, projetada e desenvolvida em um projeto que fez parte do movimento nacional de incentivo ao uso de Software Livre. Originalmente denominado FLOPREF – Free/Libre/Open Software para prefeituras – e iniciado em 2005, o projeto Via Digital teve como objetivos:

- a) Viabilizar a informatização de pequenas prefeituras, de maneira sustentada e que promova o desenvolvimento local, através de um ponto de referência online de soluções livres em gestão municipal (PORTAL VIA DIGITAL, 2010b; PORTAL VIA DIGITAL, 2010d);
- b) “Estimular uma nova dinâmica em torno da oferta de soluções de Software Livre para prefeituras, envolvendo desenvolvimento tecnológico, geração de oportunidades de negócio, emprego e renda, capacitação e informação” (PORTAL VIA DIGITAL, 2010a).

O projeto foi idealizado a partir da pesquisa **Software Livre nas Prefeituras Brasileiras: Novas Alternativas para a Informatização da Administração Pública** (SOFTEX et al., 2005), realizada e editada em 2005 pela Sociedade para Promoção da Excelência do Software Brasileiro – Softex – e pelo Instituto Nacional de Tecnologia da Informação – ITI. Na pesquisa, pôde-se constatar que todos os municípios pesquisados julgaram essencial a existência de uma biblioteca pública, de livre acesso às prefeituras, onde estivessem disponíveis informações relevantes sobre vários aspectos da informatização de prefeituras; assim como softwares, avaliações de ferramentas por especialistas, melhores práticas e catálogos de fornecedores e de profissionais capacitados (PORTAL VIA DIGITAL, 2010c).

Dessa constatação, surgiram o projeto e o portal Via Digital (PORTAL VIA DIGITAL, 2010a), através do qual se pretendeu disponibilizar “um ambiente virtual de capacitação técnica, troca de experiências, acesso a fontes idôneas de informação e criação de oportunidades de negócio a todos os participantes da cadeia de produção do setor: técnicos e executivos da administração pública, profissionais de empresas de software e serviços relacionados, participantes das comunidades de SL e empreendedores” (PORTAL VIA

DIGITAL, 2010b).

O projeto foi idealizado e executado pelo centro de Geração de Novos Empreendimentos em Software e Serviços – GeNESS –, do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, e contou ainda com mais quatro participantes: Agente Softex da Universidade Federal de Campina Grande – CGSOFT –, Observatório Digital da Sociedade Softex, Centro de Pesquisas Renato Archer/MCT – CenPRA – e a empresa OpenS Tecnologia. O projeto foi viabilizado através da Financiadora de Estudos e Projetos – FINEP (PORTAL VIA DIGITAL, 2010c).

3.2 A PROPOSTA DO NODIPO

Dentro do cenário em que as comunidades de Software Livre não participam de setores específicos e/ou não podem colaborar efetivamente por desconhecerem os requisitos, e conhecida a natureza do processo de levantamento de requisitos dessas comunidades, os integrantes do Via Digital idealizaram, em 2007, uma ferramenta que atuasse como elo entre comunidades de Software Livre e as prefeituras municipais na concepção das funcionalidades de softwares necessários a estas. Disso, foi pensado pela equipe do Via Digital e modelado e desenvolvido pelo autor deste trabalho, entre 2007 e 2008, o Nodipo, uma ferramenta colaborativa em que um usuário sem experiência em desenvolvimento de software pudesse autonomamente expressar suas necessidades em relação a determinado software.

O Nodipo oferece condições para descrição de processos e demandas, inicialmente utilizando o discurso e os conceitos do usuário especialista do domínio do sistema, que podem ser refinados iterativamente com a participação de outros usuários e com a intervenção de desenvolvedores para o esclarecimento de dúvidas e a construção de um nível mínimo de formalização.

Baseado na Web, o Nodipo possui forte viés de colaboração assíncrona – seguindo tendências da Web 2.0 –, o que tende a viabilizar a interação da comunidade de usuários especialistas, conhecedores do domínio de aplicação, geograficamente distantes. Por meio de funcionalidades de colaboração e cooperação – como debates, *rankings* e votações – os participantes podem refinar suas demandas e requisitos em sucessivos ciclos para detalhar, confirmar, classificar, priorizar, agregar e/ou corrigir as necessidades inicialmente definidas, auxiliando em uma compreensão mais clara e consensual do sistema desejado.

Dentro de sua meta, o Via Digital pensou os artefatos resultantes do processo de levantamento de requisitos com o Nodipo como requisitos semiformais de um novo sistema que podem servir de base para o desenvolvimento por empresas – públicas ou privadas –, profissionais, estudantes, voluntários ou quaisquer demais interessados.

3.3 DESCRIÇÃO DO PROCESSO DE LEVANTAMENTO DE REQUISITOS COM O NODIPO

A mais notória função do Nodipo é atuar como meio integrador de diferentes participantes no processo de eliciação⁴ de requisitos para Software Livre. O principal fator que distingue esses participantes é o nível de conhecimento: tanto em desenvolvimento de software quanto em um domínio de aplicação específico. Portanto, os desenvolvedores de Software Livre e os especialistas de domínio são considerados os atores principais do processo de eliciação com o Nodipo, ainda que inúmeros outros atores possam participar do mesmo. Para melhor compreensão desse processo, o seguinte cenário ilustra a utilização ferramenta:

- a) Um especialista de determinado domínio de aplicação deseja entender que funcionalidades teria um software que automatizasse os processos de sua área de atuação na organização onde trabalha. O primeiro passo do especialista é cadastrar esse software no Nodipo e o subsequente cadastro de funcionalidades que ele imagina que o software teria;
- b) Em seguida, os desenvolvedores interessados no levantamento de requisitos para o software do especialista passam a interagir nos debates das funcionalidades já cadastradas e a inserir mais funcionalidades – que correspondem a requisitos que eles sabem que existem naquele tipo de software;
- c) Isso faz com que o especialista do domínio responda a mais questionários, até o ponto em que os participantes têm mais bem definidos os processos correspondentes às funcionalidades inseridas, o que gera segundas versões das descrições das mesmas;
- d) Disso, um grupo de desenvolvedores não contente com a eliciação efetuada para

⁴ Apesar de não se terem encontrado referências à palavra eliciação em dicionários de língua portuguesa, ela é utilizada neste trabalho como sinônimo de levantamento, pois na literatura nacional da área de Engenharia de Software costuma-se utilizar o termo – um neologismo da palavra inglesa *elicitation*.

uma funcionalidade específica expõe os motivos do desacordo no ambiente de debate da mesma. A porcentagem de concordância, que era 50%, torna-se 93% depois da exposição e do debate em torno da definição da funcionalidade;

- e) O processo se dá dessa maneira até o ponto em que os participantes concordam que todas as funcionalidades relativas ao processo em questão foram inseridas e todas estão definidas com *status* consenso.

O resultado do cenário apresentado é o conhecimento do especialista do domínio explicitado semiformalmente. Um possível passo subsequente seria a atuação de analistas de sistemas a partir desse resultado para a geração de especificações mais formais de requisitos.

A ilustração 11 – apêndice C – esquematiza esse processo colaborativo de elicitação.

3.4 MODELAGEM

As entidades, que são os elementos que têm maior representatividade na modelagem do Nodipo, são destacados com a letra inicial maiúscula nesta seção. São elas: Categoria, Sistema, Público-alvo, Requisito, Descrição extra – do Requisito –, Prioridade, Status, Concordância, Questionário, Pergunta, Resposta e Comentário.

O elemento central do Nodipo é o Requisito. Um Requisito é atrelado a um Sistema, que por sua vez tem um Público-alvo específico e é atrelado a uma determinada Categoria de Sistemas. Além disso, o Requisito é composto por diversos elementos, tais como: Comentários, nível de Concordância, Questionário, Status, nível de Prioridade, data de criação, título – identificado como **nome** no Diagrama de Classes, vide ilustração 10, apêndice A –, autor, descrição inicial e outras possíveis versões de Descrições.

Por Categorias de Sistemas entende-se a área do domínio de aplicação do Sistema, como Planejamento Urbano, Controle Financeiro e Segurança Pública, por exemplo. Esse elemento é constituído de um título e uma descrição. O apêndice C exhibe a lista completa de categorias de software aplicadas ao Nodipo dentro do contexto do projeto Via Digital.

O Sistema contém, além do Público-alvo e da Categoria, uma descrição, um título e um autor.

Como Público-alvo, que contém apenas uma descrição, é compreendido o perfil do usuário que utilizará o Sistema cujos requisitos serão levantados através do Nodipo. São previstos como Público-alvo:

- a) **Usuários-finais;**
- b) **Desenvolvedores;**
- c) **Administradores de sistemas; e**
- d) **Outros.**

O elemento Prioridade descreve o nível de importância do Requisito: **alta, média** ou **baixa**.

A estrutura do Nodipo permite a criação de diferentes Questionários, o qual é um conjunto de Perguntas, para cada Requisito – conforme Diagrama de Classes da ilustração 10, vide apêndice A. As Respostas às Perguntas têm as seguintes propriedades: descrição, versão, data, hora e autor. O atributo versão serve como número sequencial de referência para cada nova Resposta para uma mesma Pergunta.

O questionário que o Nodipo traz como padrão advém do projeto Via Digital, de uma abordagem chamada Quality Sense Making (UNIVERSIDADE FEDERAL DE CAMPINA GRANDE apud BARROS; NAVINER, 2001). As perguntas que formam esse questionário são:

- a) Quem são servidos por esse processo?
- b) Como começa este processo?
- c) Como termina este processo?
- d) Quem participa do processo?
- e) As entradas do processo:
- f) As saídas do processo:
- g) Obs:

A função desse questionário é obter junto aos especialistas do domínio do sistema a descrição de todas as etapas de um processo organizacional específico, bem como os atores que fazem parte do mesmo. A modelagem apresentada, que permite o uso de outros questionários, possibilita a utilização de diferentes abordagens e técnicas de entrevistas para a descrição de processos.

As Descrições extras do Requisito contêm uma descrição, uma data de inserção, um autor e uma versão.

O elemento Concordância é composto por um valor de voto – sim ou não – e um autor. Esse voto de concordância pode ser alterado conforme a vontade dos participantes, mas cada participante tem direito apenas a um voto para cada Requisito.

Os Comentários são compostos por uma descrição, um autor e pela data e hora de inserção.

O Status de um Requisito corresponde ao estado em que o mesmo se encontra no processo de levantamento. São eles:

- a) **Inicial** – no qual o processo de levantamento do Requisito está em fase inicial de desenvolvimento, com pouca ou nenhuma informação além do título;
- b) **Avançado** – em que o processo de levantamento se encontra em estágio avançado;
- c) **Em análise** – no qual o processo de levantamento do Requisito se encontra em estágio avançado mas há informações do mesmo que necessitam de consenso entre os participantes;
- d) **Consenso** – quando o processo de levantamento está finalizado e os participantes concordaram totalmente ou em grande parte com as informações levantadas sobre o Requisito. Neste estado não são mais permitidas inserções de quaisquer dados sobre o Requisito.

Todos os campos textuais de todos os formulários de cadastro do Nodipo são obrigatórios, exceto os campos textuais das Respostas do Questionário apresentado no cadastrado de um Requisito. Esses campos recebem automaticamente o valor **Não especificado no cadastro da funcionalidade** caso não sejam preenchidos na tela de cadastramento de um Requisito.

Dada a modelagem apresentada, pretende-se que seja possível:

- a) Obter informações semiformais dos especialistas do domínio de aplicação do Sistema, através principalmente do Questionário e das novas versões das Descrições; além da possibilidade de inserção de mais de uma Resposta para cada Pergunta, o que tende a ajudar no entendimento dos processos do Sistema e reforça o caráter colaborativo da ferramenta;
- b) Observar a satisfação dos participantes quanto à elicitação de determinado requisito, com a votação da Concordância, que mede em porcentagem – votos positivos pelo total de votos – essa satisfação;
- c) Promover um meio de debate geral em relação ao Requisito, com os Comentários;
- d) Refinar as definições dos Requisitos, com as novas versões de Descrições;
- e) Visualizar as etapas de elicitação, para que os usuários acompanhem a evolução do processo, através do Status;
- f) Registrar o processo e o resultado do levantamento, possibilitando, a partir do histórico, o entendimento de como o resultado foi obtido; e

- g) Priorizar requisitos.

3.5 PADRÃO DE PROJETO DATA ACCESS OBJECT – DAO

O código-fonte do Nodipo conta com a implementação do padrão de projeto – *design pattern*, em inglês – DAO. Essencialmente, esse padrão de projeto implementa um mecanismo que encapsula e abstrai o acesso à fonte de dados na Programação Orientada a Objetos (CORE J2EE PATTERN CATALOG, 2010).

A vantagem de se usar o DAO é a possibilidade de se trocar a fonte de dados sem que seja alterado o **código de lógica de negócio**. Isso é possível pois o objeto DAO atua como uma interface que é sempre igual aos olhos da classe que a utiliza, mas que pode ser modificada internamente para o acesso a diferentes fontes de dados (CORE J2EE PATTERN CATALOG, 2010).

Por **código de lógica de negócio** entende-se a parte do código que trata das regras de negócio da aplicação (AHMED; UMRYSH, 2001), ou seja, que é responsável pelo processamento que não corresponde à camada de apresentação – interface que o usuário vê – nem à camada de dados – armazenamento de dados – (MCLAUGHLIN, 2002). Essa subdivisão é estabelecida pela arquitetura de software em 3 camadas, utilizada por padrões como o Java Platform, Enterprise Edition – Java EE, anteriormente J2EE (MCLAUGHLIN, 2002; AHMED; UMRYSH, 2001) – para o desenvolvimento de aplicações *enterprise* – este, termo amplamente conhecido e utilizado para identificar sistemas de grande porte.

Para o Nodipo, a implementação do *design pattern* DAO permite, por exemplo, que sejam utilizados outros Sistemas de Gerenciamento de Bases de Dados, que não o PostgreSQL, ajustando-se apenas a implementação das classes DAO sem efeitos colaterais nas demais partes do código.

4 DESENVOLVIMENTO

A maior do parte do Nodipo foi implementada entre julho de 2007 e junho de 2008. Outros ajustes e correções foram feitos depois desse período, além de ter tido grande parte do código internacionalizado entre junho e julho de 2009. Ademais, o Nodipo foi utilizado como repositório de requisitos em dezembro de 2009, quando foi efetivamente posto em prática durante a experiência de avaliação do mesmo, a ser relatada a seguir.

É também importante salientar que os requisitos no Nodipo foram chamados de **Funcionalidades** na interface gráfica da ferramenta, no intuito de melhorar o entendimento por parte dos usuários-finais do aplicativo que não sejam familiarizados com as nomenclaturas utilizadas no meio informático.

4.1 ESPECIFICAÇÕES TÉCNICAS

Para o desenvolvimento do Nodipo foram utilizadas as seguintes tecnologias:

- a) Linguagem de programação **PHP**, versão 5;
- b) Linguagem de *script* **Javascript** – para o uso de **AJAX**;
- c) Sistema de Gerenciamento de Banco de Dados **PostgreSQL**, versão 8.2;
- d) **PLPGSQL** – linguagem procedural do PostgreSQL;
- e) Linguagem de marcação **XHTML**;
- f) Linguagem de folha de estilo **CSS**.

4.2 DESCRIÇÃO DAS INTERFACES GRÁFICAS

Como poderá ser observado nas ilustrações a seguir, a interface gráfica do Nodipo foi feita tomando-se como referência a formatação do portal do Via Digital (PORTAL VIA DIGITAL, 2010a), principalmente no que diz respeito a cores, estilo e tamanho das letras.

4.2.1 A tela inicial

Na tela inicial é exibida a listagem dos Sistemas cadastrados no Nodipo – vide ilustração 1, a seguir. Ao selecionar-se um dos Sistemas, na mesma tela são exibidas as informações básicas – nome, Categoria, Público-alvo e descrição – bem como a lista de funcionalidades do mesmo, como pode ser visto na ilustração 2. Ademais, na mesma tela tem-se acesso às janelas de cadastro de novo Sistema e de novos Requisitos do Sistema selecionado.

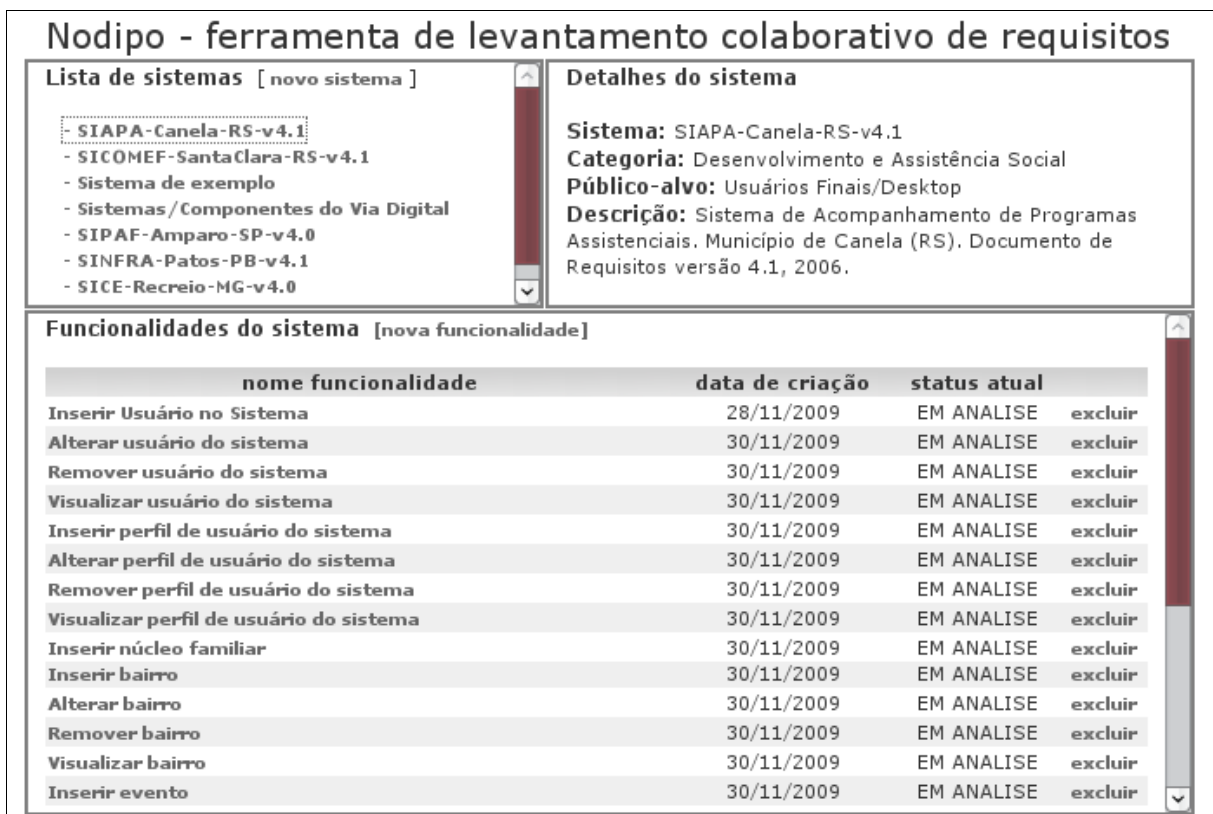


Ilustração 2: Tela inicial – exibição de lista de funcionalidade de determinado Sistema.

4.2.2 Cadastro de um Sistema

Acessada através do botão **novo sistema**, que é localizado no quadro de listagem de Sistemas, a janela de cadastro de um Sistema mostra os campos nome, descrição e Público-alvo a serem preenchidos para a inserção do novo Sistema.

4.2.3 Cadastro de um Requisito

A janela de cadastro de um Requisito – funcionalidade, na interface – exhibe os campos de nome, descrição, Status, Prioridade e o Questionário, o qual traz as Perguntas e os campos para as Respostas iniciais.

A imagem mostra uma janela de software intitulada "Cadastro de nova funcionalidade" com um botão "[fechar]" no canto superior direito. O sistema atual é identificado como "SIAPA-Canela-RS-v4.1".

O formulário contém os seguintes campos e controles:

- Nome:** Um campo de texto único.
- Descrição:** Um campo de texto de área.
- Status:** Um menu suspenso com o valor "EM ANALISE" selecionado e um botão "500" ao lado.
- Prioridade:** Três opções de rádio: "Alta", "Média" (selecionada) e "Baixa".
- Cadastrar:** Um botão centralizado para salvar o registro.
- Quem são servidos por esse processo?:** Um campo de texto para a resposta.
- Como começa este processo?:** Um campo de texto para a resposta.

4.2.4 Exibição de um Requisito

Nesta interface são exibidas as informações básicas do Requisito – Sistema do qual ele faz parte, nome, data de criação, autor e descrição inicial – bem como a Concordância dos participantes, o nível de Prioridade, os Comentários e o Questionário, sendo que se o Questionário é exibido, os Comentários são ocultados e vice-versa. Em resumo, a interface abrange todas as informações do requisito, além de dar acesso a:

- a) inserção de Comentários;
- b) inserção de novas Respostas ao Questionário;
- c) inserção de novas versões de Descrições para o Requisito;
- d) alteração da Prioridade e do Status;
- e) geração de versão para impressão; bem como ao
- f) quadro de Concordância, no qual se pode votar favorável ou contrariamente à elicitación do requisito.

A ilustração 12 do apêndice D mostra a tela de exibição de uma funcionalidade.

4.2.5 Questionário

Assim como as Respostas do Questionário apresentado no cadastro do Requisito, os participantes podem inserir mais Respostas às Perguntas do mesmo Questionário. Isso possibilita um pequeno debate focado no tema da Pergunta, ou até mesmo reformulações das Respostas no sentido de melhor descrever os processos do Sistema. Para tal, o participante deve utilizar o botão **responder** localizado ao lado das perguntas – a exemplo da ilustração 12 do apêndice D.

4.2.6 Debate

Para inserir comentários sobre um Requisito, o participante deve acessar a seção de debates do mesmo via botão **ver debate** – vide ilustração 4, a seguir. Na sequência, o

participante tem acesso à janela de inserção de Comentários pelo botão **novo comentário**.

Nodipo - ferramenta de levantamento colaborativo de requisitos

Sistema de exemplo [versão para impressão] [fechar]

Funcionalidade: Funcionalidade de teste [mudar prioridade]

Autor: Anônimo

Descrição [adicionar descrição]
Na descrição das funcionalidades cabem características gerais de um determinado processo que o sistema deverá automatizar.

As descrições recebem versões, conforme forem sendo aprimoradas de acordo com as discussões e respostas ao questionário que surgirem.

Anônimo, em 02/12/2009 (versão 2) [editar] [excluir]

Prioridade: Média

Concordância

100%


1


0

Status: EM ANALISE [mudar status]

Data de criação: 02/12/2009

[ver questionário](#) [ver debate](#)

Debate [novo comentário]

"Um debate pode (e deve) ser iniciado para se chegar a um consenso sobre esta funcionalidade."

Anônimo, 02/12/2009 às 18:12:21h [editar] [excluir]

"Há possibilidade de debater todos os elementos da ferramenta (Descrição, perguntas do questionário e no próprio debate)."

Anônimo, 02/12/2009 às 18:13:31h [editar] [excluir]

"As funcionalidades podem ter Prioridade Alta, Média ou Baixa"

Anônimo, 02/12/2009 às 18:15:06h [editar] [excluir]

"Cada elemento do questionário está sujeito a debate (os participantes da discussão podem clicar em "Responder" para acrescentar sua visão sobre cada tópico do questionário)."

Anônimo, 02/12/2009 às 18:15:13h [editar] [excluir]

4.2.7 Voto de Concordância

O voto de Concordância é dado em relação a um Requisito de um Sistema. Com ele visa-se a proporcionar aos usuários a visualização de quão alta ou baixa é a aceitação daquele Requisito entre os envolvidos no processo de elicitação do Sistema correspondente. Para votar, basta o usuário utilizar o botão com sinal de positivo para voto favorável, ou o botão com sinal de negativo para voto contrário, como mostrado na ilustração a seguir. Além do número de votos a favor e contra, o quadro de concordância mostra a porcentagem de aprovação – quantidade de votos a favor pelo número total de votantes.



Ilustração 5: Quadro de concordância em relação ao levantamento de um requisito

4.2.8 Alteração de Status e Prioridade

A alteração do Status e da Prioridade de um Requisito são muito parecidas quanto ao formato: ambas mostram a lista de opções para modificação ao ser utilizado o botão correspondente. Por exemplo, através do botão **mudar prioridade** é exibido o elemento HTML *select* – vide ilustração 6 – que possibilita a exibição da lista de opções alta, média e baixa, diferentemente do valor simples que é exibido normalmente, exemplificado na ilustração 4. Ao selecionar-se uma opção diferente, o valor da Prioridade é alterado. O mesmo acontece para o Status, mas com os valores correspondentes do mesmo e através do botão **mudar status**.

Nodipo - ferramenta de levantamento colaborativo de requisitos

SIPAF-Amparo-SP-v4.0 [versão para impressão]
[fechar]

Funcionalidade: Remover perfil de usuário do sistema

Autor: Anônimo

Descrição [adicionar descrição]
Funcionalidade de remoção de perfil de usuário do sistema.

Prioridade: Alta ▾

Concordância

--

0

0

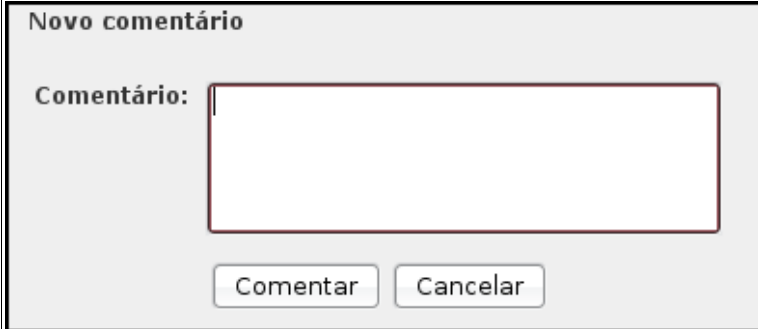
Status: EM ANALISE ▾

Data de criação: 15/12/2009

[ver questionário](#) [ver debate](#)

4.2.9 Cadastro de Comentários, Respostas e Descrições extras

O cadastro desses elementos, na tela de exibição de um Requisito, é o mais simples. Trata-se apenas da janela de cadastro com o campo textual de descrição do elemento correspondente, como é exemplificado abaixo com o elemento Comentário:



A imagem mostra uma janela de diálogo intitulada "Novo comentário". Dentro da janela, há um campo de texto rotulado "Comentário:" que está atualmente vazio. Abaixo do campo de texto, há dois botões: "Comentar" e "Cancelar".

4.2.10 Versão para impressão de um Requisito

É possível obter uma versão de impressão de um Requisito no Nodipo. Na tela do Requisito, através do botão **versão para impressão** – localizado na parte superior esquerda ao lado do nome do Sistema do Requisito, como é mostrado na ilustração 6, por exemplo –, é exibida a janela de seleção de opções para a geração da versão para impressão – ilustração 8.

Nessa janela pode-se escolher quais elementos devem aparecer na versão para impressão além das informações básicas do Requisito. São eles: as **descrições extras**, os **comentários** e o **questionário com as respostas dos participantes**. Qualquer combinação pode ser feita para a exibição desses elementos na versão da impressão.

Através do botão **Gerar versão de impressão** é criada uma nova janela do navegador – janela *pop-up* – com as informações selecionadas organizadas para impressão, conforme ilustração 9. O participante tem acesso à janela de impressão do navegador via botão **imprimir** localizado no canto superior esquerdo da janela.

Escolha as informações para a versão de impressão:

descrições extras debate questionário **Gerar versão de impressão** **Cancelar**

Funcionalidade: Remover perfil de usuário do sistema **Prioridade:** Alta

Autor: Anônimo

Descrição [adicionar descrição]
Funcionalidade de remoção de um perfil de usuário do sistema.

Concordância

0 0

 imprimir

Funcionalidade "Remover perfil de usuário do sistema"

Informações gerais

Autor: Anônimo **Prioridade:** 1

Sistema: SICOMEF-SantaClara-RS-v4.1 **Status:** EM ANALISE

Categoria: Desenvolvimento e Assistência Social **Concordância:** --

Data de criação: 02/12/2009

Descrição inicial
Funcionalidade de remoção de um perfil de usuário do sistema.

Descrições extras

Não há descrições extras.

Questionário

Quem são servidos por esse processo?
Anônimo, 02/12/2009 às 10:13:50h
Usuário Administrador.

Como começa este processo?
Anônimo, 02/12/2009 às 10:13:50h
Esta funcionalidade começa quando um usuário deseja remover as funcionalidades ou descrição de um perfil de Usuário do sistema. Primeiramente, devem ser listados todos os perfis do sistema e, ao escolher um dos listados, será obtida a visualização de seus dados. A partir disso, este mesmo perfil visualizado poderá ter seus dados removidos.

Como termina este processo?
Anônimo, 02/12/2009 às 10:13:50h

Ilustração 9: Versão para impressão de uma funcionalidade.

4.3 ESTÁGIO ATUAL DO DESENVOLVIMENTO

O Nodipo foi a princípio desenvolvido para ser integrado à plataforma de colaboração e gerenciador de projetos GForge, no qual é baseado o repositório do Via Digital (REPOSITÓRIO VIA DIGITAL, 2010). Contudo, devido à descontinuidade do projeto Via Digital, o Nodipo acabou por não ser integrado inteiramente ao repositório do projeto. Fato esse que posicionou a ferramenta em um indesejável meio termo: nem totalmente integrada ao repositório do Via Digital, nem completamente funcional quando utilizada sozinha.

A funcionalidade principalmente afetada por essa característica é relativa aos usuários: a estrutura do banco de dados não conta com usuários próprios. Para isso há um atributo de referência a um identificador de usuário em cada tabela que possui como atributo um usuário autor. Como eram esperados os usuários cadastrados no repositório do Via Digital, atualmente o Nodipo funciona com uma estrutura que define o nome **Anônimo** como usuário conectado.

Desta forma, são necessários ajustes estruturais na ferramenta para que ela possa ser utilizada por múltiplos usuários e possa ser colocado em prática o caráter colaborativo da mesma.

Entretanto, o Nodipo pode – conforme será visto a seguir – ser utilizado como repositório de requisitos sem qualquer implementação além da existente, pois tem implementadas todas as funcionalidades inerentes ao processo de elicitação de requisitos – o que pode ser observado na seção de descrição das interfaces gráficas.

4.4 AVALIAÇÃO DA FERRAMENTA

Entre novembro e dezembro de 2009 a ferramenta apresentada neste trabalho foi colocada em prática com a função de repositório de requisitos (NODIPO VIA DIGITAL, 2010), para que se pudesse avaliar a utilização da mesma principalmente no que concerne à usabilidade. Participaram deste exercício de avaliação o autor deste trabalho e Wagner Saback Dantas – membro da banca examinadora.

Foram utilizados para a avaliação seis softwares que fazem parte do Via Digital,

cujas especificações de requisitos já estavam prontas. São eles (NODIPO VIA DIGITAL, 2010):

- a) SIAPA-Canela-RS-v4.1;
- b) SICOMEF-SantaClara-RS-v4.1;
- c) Sistemas/Componentes do Via Digital;
- d) SIPAF-Amparo-SP-v4.0;
- e) SINFRA-Patos-PB-v4.1;
- f) SICE-Recreio-MG-v4.0.

Dessa experiência pôde-se obter conclusões sobre a utilização do Nodipo e, por consequência, sugestões de melhorias e correções. Os resultados da avaliação são listados a seguir:

- a) **Exibir avisos sobre utilização concorrente:** Apesar de o caráter colaborativo não ter sido diretamente testado na avaliação, pois se incluíram requisitos já especificados, pôde-se constatar que não há alertas quando dois usuários utilizam a ferramenta ao mesmo tempo. Tal fato é relacionado ao estágio atual do desenvolvimento do Nodipo, no que diz respeito às funcionalidades referentes aos usuários, como foi detalhado na seção 4.3. Contudo, funcionalidades que conscientizem os usuários sobre utilizações concorrentes são fundamentais a ferramentas colaborativas;
- b) **Realçar campos de formulários selecionados:** O destaque em campos de formulários selecionados podem agilizar a utilização da ferramenta. Por exemplo, se houver destaque, é maior a probabilidade de o usuário perceber que o primeiro campo de cada formulário é automaticamente selecionado na ferramenta, o que evita que o usuário selecione manualmente o primeiro campo de um formulário desnecessariamente;
- c) **Janelas realçadas e flutuantes:** Mover janelas de uma posição a outra é uma funcionalidade bastante utilizada em diversos tipos de software. A falta dessa implementação pode frustrar os usuários, pois a ferramenta conta com janelas estáticas. Melhor destacá-las em relação aos demais elementos também pode ser uma boa ideia;
- d) **Rever títulos das janelas de cadastro e possíveis ambiguidades:** Por exemplo, a janela de cadastro de sistemas traz o título **Cadastro de novo sistema**, enquanto o título do campo relativo ao público-alvo é **Público-alvo**. Nesse caso, o usuário poderia ficar em dúvida se o público-alvo é referente ao tipo de usuário

que irá utilizar o Nodipo no processo de elicitação – somente administradores de sistemas levantariam os requisitos, por exemplo – ou se o público-alvo é referente ao perfil de usuários que utilizarão o sistema sendo cadastrado uma vez que desenvolvido, ou seja, em produção. Sendo a última alternativa a correta – afinal, o processo de elicitação com o Nodipo não é, nem deve ser, restrito a um perfil de participante –, um título como **Dados sobre o novo sistema** parece instruir melhor o usuário, pois indica que público-alvo é um dado sobre o sistema que terá os requisitos elicitados e não sobre o perfil de usuário que participará do processo de levantamento de requisitos;

- e) **Informar quais campos dos formulários são obrigatórios:** Apesar de existirem poucos campos que não o são, é importante que isso seja informado ao usuário;
- f) **Instruções mais completas para os formulários:** Por exemplo, as respostas do questionário são os únicos campos não obrigatórios – que não precisam ser preenchidos no momento do cadastro – e não são exibidas instruções relativas a isso;
- g) **Aumentar número de linhas dos campos de respostas do questionário:** Ao digitar textos extensos o usuário passa a ter uma visualização muito limitada dos mesmos. Trata-se de aumentar o número de linhas ou a altura – via CSS – do elemento HTML *textarea*;
- h) **Possibilitar inserção de arquivos a elementos textuais:** Essa funcionalidade pode ser interessante porque possibilita, por exemplo, a agregação de esquemas, diagramas e documentos a elementos textuais – tais como respostas e comentários – que podem ajudar no entendimento dos processos dos sistemas;
- i) **Versão para impressão de todos os requisitos de um sistema:** Atualmente as versões para impressão são atreladas a um requisito. Gerar automaticamente versões para impressão – talvez em formato PDF – de todos os requisitos de um sistema pode ser interessante para procedimentos pós-elicitação;
- j) **Melhorar exibição do progresso do levantamento de requisitos dos sistemas:** Já é possível verificar o progresso através da tabela de funcionalidades do sistema, pelo *status* atual das mesmas. Para incrementar isso, por exemplo, poderia ser exibida na mesma tabela a porcentagem de concordância dos participantes para cada funcionalidade. Atualmente só é possível fazê-lo olhando-se requisito por requisito.

Cabe ressaltar que a validação foi feita sem qualquer metodologia de ergonomia ou de usabilidade de software, de modo que os usuários lidaram com a ferramenta da forma que desejaram. Portanto, as conclusões refletem essa característica, pois são baseadas puramente no conhecimento tácito dos avaliadores, na condição de usuários e desenvolvedores de software que são.

4.5 FUNCIONALIDADES PARA A VERSÃO *STANDALONE*

Para funcionar independentemente de outros programas, é necessário que o Nodipo possua as seguintes funcionalidades:

- a) **Diferenciação de usuários administrativos e usuários comuns:** Importante para atividades de moderação, por exemplo, em que os usuários administradores têm poderes para banir usuários desrespeitosos, excluir comentários de *spammers*, entre outros.
- b) **Interface administrativa para gerenciamento de usuários:** Além de funcionalidades que permitam que os usuários cadastrem-se autonomamente.
- c) **Interface para gerenciamento de categorias de sistemas:** Apesar de existir extensa lista de categorias de programas disponível, atualmente só é possível adicionar novas categorias via banco de dados no Nodipo.
- d) **Interface para gerenciamento de questionários:** Para adição de novas perguntas, montagem de questionários a partir de perguntas já cadastradas e escolha de questionário padrão para determinados requisitos ou até mesmo para sistemas.

5 CONCLUSÃO

Pôde-se visualizar neste trabalho as vantagens técnicas atreladas ao uso de Software Livre e que o mesmo é tomado como fator estratégico pelo Governo Brasileiro, no sentido de alinhar o país ao novo cenário mundial no que diz respeito a tecnologias de informação e comunicação.

Por outro lado, há o fato de o Software Livre ser ausente ou pouco presente em áreas de aplicação específicas, pois as comunidades de FLOSS são distantes delas. Para preencher essa lacuna, o projeto Via Digital propôs uma ferramenta, desenvolvida e modelada pelo autor deste trabalho, que pudesse ser utilizada como agente integrador entre as comunidades de Software Livre e as pessoas das áreas de aplicação em que o FLOSS é escasso: o Nodipo.

A modelagem da ferramenta Nodipo possui elementos característicos da Engenharia Social de Requisitos e de softwares sociais, que são propícios para a interação entre os integrantes das comunidades de FLOSS e os demais participantes, de forma que os mesmos possam colaborativamente trocar conhecimentos, descrever os processos dos sistemas e elicitar os requisitos necessários a estes.

Dentre as ferramentas estudadas neste trabalho, viu-se que algumas têm elementos e termos bastante específicos da Engenharia de Software tradicional, que não parecem ser adequados para a utilização em ambientes colaborativos que contam com a participação dos usuários-finais. Por outro lado, outra ferramenta, do projeto SoftWiki, é bastante similar ao Nodipo em diversos aspectos e, ainda, agrega funcionalidades semânticas à Engenharia Social de Requisitos de modo a enriquecer o processo colaborativo e distribuído de levantamento e especificação de requisitos.

Foi constatado também que a ferramenta apresentada neste trabalho necessita de ajustes para que possa ser efetivamente utilizada. Ficou claro, inclusive, que a experiência de avaliação do Nodipo não testou o principal fator que a ferramenta propõe: a colaboração. Dessa forma, apesar de possuir elementos já testados e validados em outras ferramentas – vide resultados positivos apresentados pela equipe do projeto SoftWiki, Anexo A –, tornam-se necessárias experiências de avaliação mais abrangentes, que envolvam principalmente a colaboração entre os participantes ao utilizarem o Nodipo. Além disso, outros ajustes e melhorias foram sugeridos a partir da experiência de avaliação de usabilidade do Nodipo e do estudo de outras ferramentas.

Imaginando um cenário em que a ferramenta apresentada neste trabalho seja utilizada em ampla escala, atuando como referência de repositório de requisitos de Software Livre no país, seriam especificados os requisitos de muitos dos softwares das áreas de aplicação nas quais o FLOSS não é presente, o que estimularia a atuação e a criação de empresas nacionais no desenvolvimento, adaptação, manutenção, capacitação e suporte desses softwares em todo o território nacional. Assim, ter-se-iam aquecidas as economias locais de cada cidade em que esse processo ocorresse e, por consequência, a economia nacional, em um ciclo virtuoso.

Portanto, uma vez que devidamente ajustado, a utilização do Nodipo junto à sociedade tende a ser de grande valia, pois pode auxiliar no fomento do uso de Software Livre no país, o que traz vantagens como independência tecnológica, redução de custos e estímulo à economia nacional.

6 TRABALHOS FUTUROS

6.1 FUNCIONALIDADES RELEVANTES A IMPLEMENTAR

Dentre as funcionalidades a implementar no Nodipo, apontam-se as já citadas: as que servem para que o mesmo funcione independentemente de outros programas – vide seção 4.5 – e as sugeridas a partir da experiência de avaliação de usabilidade – seção 4.4.

Além dessas, percebeu-se – a partir do conhecimento adquirido na elaboração deste trabalho e da análise das ferramentas similares – que há ajustes e funcionalidades novas que podem potencializar as características colaborativas do Nodipo. São eles:

- a) **Adesão de usuários a sistemas:** Pensando em um contexto de uso em grande escala, nem todo usuário deve estar necessariamente relacionado a todos os sistemas e requisitos inseridos no Nodipo, como acontece atualmente. Essa ideia é semelhante a uma funcionalidade presente na ferramenta CRETA (TOGNERI et al., 2002), em que os participantes são alocados a projetos quando estes são criados. Para o Nodipo, entretanto, é mais adequado que o participante escolha o sistema do qual ele deseja fazer parte, ao invés de outro participante definir isso por ele;
- b) **Reuso de requisitos para diferentes sistemas:** Desenvolvedores poderiam identificar requisitos já levantados em outros sistemas e copiá-los para um novo sistema em que se começa a levantar os requisitos, de modo estimular os participantes não familiarizados com o processo, por exemplo;
- c) **Busca por requisitos:** Principalmente considerando o reuso de requisitos, bem como um cenário de utilização em larga escala do Nodipo, essa funcionalidade torna-se bastante relevante;
- d) **Votação para a prioridade do requisito:** Na implementação atual da ferramenta apresentada neste trabalho, é permitido a qualquer usuário alterar a prioridade de um requisito. Dessa forma, as decisões sobre quão prioritário é o requisito têm de ser discutidas indiretamente – através do debate, por exemplo. Se implementado um mecanismo de votação, a exemplo da concordância, a priorização refletiria automaticamente a opinião dos participantes;

- e) **Modificação da interface:** O considerável número de sugestões de alteração e correção exibido na seção de avaliação demonstra o fato de a interface do Nodipo não ter sido desenvolvida com base em estudos de ergonomia e usabilidade. Fazer ajustes na interface levando em consideração essas áreas de conhecimento pode auxiliar, inclusive, na maior participação dos atores, conforme afirmam Lohmann et al. (2008);
- f) **Registro de alterações:** Essa funcionalidade, implementada parcialmente na ferramenta apresentada neste trabalho, faz parte das categorias que influenciam a Engenharia de Requisitos em ambientes distribuídos apontadas por Lopes e Audy (2003). Ademais, cabe mencionar que, segundo Togneri et al. (2002), o registro de alterações é relacionado à gestão do conhecimento, que permite identificar o histórico, as argumentações utilizadas e os raciocínios empregados em cada decisão tomada em determinado cenário.

6.2 DIRECIONAMENTOS TEÓRICOS

Dada a constatação da grande similaridade entre o projeto SoftWiki e o tema deste trabalho, e considerando que o aplicativo SoftWiki possui a camada semântica que traz vantagens – vide Anexo A –, parece factível e interessante a adição das características e implementações semânticas do SoftWiki ao Nodipo, o que o complementaria.

Ainda no que diz respeito às características semânticas, levando em conta que os pesquisadores do projeto SoftWiki concebem a Engenharia Social de Requisitos como um caso especial de Engenharia do Conhecimento ágil e distribuída (THE SOFTWIKI PROJECT, 2010), julga-se interessante direcionar os estudos no sentido de tornar o Nodipo uma ferramenta de gestão de conhecimento – de extração e explicitação de conhecimento – dentro do escopo do levantamento colaborativo de requisitos. Sendo essa uma maneira de acompanhar e reforçar os estudos já executados pela equipe do SoftWiki.

Outro possível direcionamento de estudos futuros é aprofundar o entendimento das dinâmicas de funcionamento das comunidades *online* e de Software Livre de maneira que se possam sugerir ações para a aplicação real do Nodipo em ambientes colaborativos.

REFERÊNCIAS

AHMED, Khawar Zaman; UMRYSH, Cary E. **Developing Enterprise Java Applications with J2EE and UML**. 1ª ed., Addison-Wesley Pub Co, dez. 2001.

BRITO, Regiane Andrade; VASCONCELOS, Alexandre Marcos Lins de. **Integrando Groupware e Gerenciamento de Requisitos no Suporte à Engenharia de Requisitos Distribuída**. Anais do IX Workshop em Engenharia de Requisitos, Rio de Janeiro, julho, 2006.

CORE J2EE PATTERN CATALOG. **Core J2EE Patterns – Data Access Object**.

Disponível em:

<<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>>. Acesso em: 08 de maio de 2010.

ESDSSR WORKING GROUP. **Open Source**. Earth Science Data Systems Software Reuse Working Group. Out. 2008a. Disponível em:

<http://www.esdswg.com/softwarereuse/Open_Source>

Acesso em: 08 de maio de 2010.

ESDSSR WORKING GROUP. **Reuse Definitions**. Earth Science Data Systems Software

Reuse Working Group. Nov. 2008b. Disponível em: <<http://www.esdswg.com/softwarereuse/Resources/library/reuse-definitions>>. Acesso em: 08 de maio de 2010.

EUROPEAN UNION. **EUROPEAN INTEROPERABILITY FRAMEWORK FOR PAN-EUROPEAN eGOVERNMENT SERVICES**. Luxembourg: Office for Official Publications of the European Communities, Bélgica, 2004. Disponível em:

<<http://ec.europa.eu/idabc/servlets/Doc?id=19528>>. Acesso em: 08 de maio de 2010.

EXÉRCITO BRASILEIRO. **PLANO DE MIGRAÇÃO PARA SOFTWARE LIVRE NO EXÉRCITO BRASILEIRO**. 3ª Edição. Brasília, fev. 2007. Disponível em:

<http://www.softwarelivre.gov.br/casos/Plano_Migracao_Soft_Livre_13FEV07.pdf>. Acesso em: 08 de maio de 2010.

FREEDMAN, Terry. **Coming Of Age: An Introduction To The NEW Worldwide Web**. 1ª ed., Terry Freedman, Inglaterra, 2006.

FREE SOFTWARE FOUNDATION. **The Free Software Definition**. Disponível em <<http://www.gnu.org/philosophy/free-sw.html>>. Acesso em 2 de julho de 2010.

GASSER, Les; SCACCHI, Walt.; RIPOCHE, Gabriel.; PENNE, Bryan. **Understanding**

Continuous Design in F/OSS Projects. 16th International Conference on Software Engineering & its Applications, Paris, França, dezembro, 2003.

GOVERNO FEDERAL. **e-PING – Padrões de Interoperabilidade de Governo Eletrônico**, versão 2010, Brasília, Brasil, 11 de dezembro de 2009. Disponível em: <<http://www.governoeletronico.gov.br/anexos/e-ping-versao-2010>>. Acesso em: 08 de maio de 2010.

GOVERNO FEDERAL. **Guia Livre – Referência de Migração para Software Livre do Governo Federal**, Brasília, Brasil, 2005. Disponível em: <<https://www.governoeletronico.gov.br/anexos/guia-livre-versao-1.0>>. Acesso em: 08 de maio de 2010.

GOVERNO FEDERAL. **Sociedade da Informação no Brasil – Livro Verde**, Brasília, Brasil, setembro de 2000. Disponível em: <http://www.mct.gov.br/upd_blob/0004/4809.zip>. Acesso em: 08 de maio de 2010.

HALL, Jon “Maddog”. **Quando 25% são mais que um quarto do custo.** Linux Magazine nº 49, Linux New Media do Brasil Editora, São Paulo, Brasil, Novembro de 2008. Disponível em: <http://www.linux-magazine.com.br/article/jon_maddog_hall_lm49>. Acesso em: 18 de maio de 2010.

KRUEGER, Charles W. **Software Reuse.** ACM Computing Surveys, Vol. 24, No. 2, jun. 1992.

LOHMANN, Steffen; DIETZOLD, Sebastian; HEIM, Philipp; HEINO, Norman. **A Web Platform for Social Requirements Engineering.** Workshopband, Alemanha, 2009.

LOHMANN, Steffen; HEIM, Philipp; SÖREN, Auer; DIETZOLD, Sebastian; RIECHERT, Thomas. **Semantifying Requirements Engineering – The SoftWiki Approach.** Proceedings of I-SEMANTICS '08, Graz, Áustria, Setembro, 2008a.

LOHMANN, Steffen; RIECHERT, Thomas; SÖREN, Auer; ZIEGLER, Jürgen. **Collaborative Development of Knowledge Bases in Distributed Requirements Elicitation.** 2008b.

LOPES, Leandro Teixeira; AUDY, Jorge Luiz Nicolas. **Em busca de um modelo de referência para engenharia de requisitos em ambientes de desenvolvimento distribuído de software.** Anais do VI Workshop em Engenharia de Requisitos – WER03 –, São Paulo, 2003.

MCLAUGHLIN, Brett. **Building Java Enterprise Applications, Vol I: Architecture.** 1ª ed.,

O'Reilly, 2002.

NAKAGAWA, Elisa Yumi; CRUZ, Norberto Fukuta; MALDONADO, José Carlos. **Relevância dos requisitos no desenvolvimento de software livre**. VII Workshop Software Livre, VII Fórum Internacional Software Livre, Porto Alegre, 2006.

NODIPO VIA DIGITAL. Versão do portal do Via Digital utilizada na avaliação da ferramenta. Disponível em <<http://nodipo.viadigital.org.br>>. Acesso em: 08 de maio de 2010.

NUSEIBEH, Bashar; EASTERBROOK, Steve. **Requirements Engineering: A Roadmap**. International Conference on Software Engineering, Proceedings of the Conference on The Future of Software Engineering, Pág. 35 - 46, ISBN:1-58113-253-0, Irlanda, 2000.

PORTAL DO GOVERNO ELETRÔNICO DO BRASIL. **Comitê Executivo**. 2010a. Disponível em <<https://governoeletronico.gov.br/o-gov.br/comites/comite-executivo>>. Acesso em: 08 de maio de 2010.

PORTAL DO GOVERNO ELETRÔNICO DO BRASIL. **Conheça o Gov.br**. 2010b. Disponível em: <<http://www.governoeletronico.gov.br/o-gov.br>>. Acesso em: 08 de maio de 2010.

PORTAL VIA DIGITAL. **Apresentação**. 2010a. Disponível em: <<http://www.viadigital.org.br>>. Acesso em: 08 de maio de 2010.

PORTAL VIA DIGITAL. **Descrição**. 2010b. Disponível em: <http://www.viadigital.org.br/index.php?option=com_content&task=view&id=3&Itemid=20> Acesso em: 08 de maio de 2010.

PORTAL VIA DIGITAL. **Histórico**. 2010c. Disponível em: <http://www.viadigital.org.br/index.php?option=com_content&task=view&id=4&Itemid=20> Acesso em: 08 de maio de 2010.

PORTAL VIA DIGITAL. **Objetivos e metas**. 2010d. Disponível em: <http://www.viadigital.org.br/index.php?option=com_content&task=view&id=5&Itemid=20> Acesso em: 08 de maio de 2010.

PRESSMAN, Roger S. Engenharia de Software. 6ª edição, McGraw-Hill, ISBN: 8586804576, 2006.

RAYMOND, Eric S. **The Cathedral & the Bazaar**. O'Reilly. ISBN 1-56592-724-9, 1999.

Disponível em <<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar>>. REPOSITÓRIO VIA DIGITAL. Repositório do Via Digital. Disponível em: <<http://repositorio.viadigital.org.br>>. Acesso em: 08 de maio de 2010.

RIECHERT, Thomas; LAUENROTH, Kim; LEHMANN, Jens; SÖREN, Auer. **Towards Semantic based Requirements Engineering**. Proceedings of the 7th International Conference on Knowledge Management (I-KNOW), Alemanha, 2007.

SAVI, Rafael; De LUCCA, José Eduardo. **Produção compartilhada de software para governos municipais**. VIII Workshop de Software Livre, Anais VIII Fórum Internacional de Software Livre, Porto Alegre, 2007.

SCACCHI, Walt. **Socio-Technical Design**. The Encyclopedia of Human-Computer Interaction. Berkshire Publishing Group, 2004a.

SCACCHI, Walt. **Understanding the Requirements for Developing Open Source Software Systems**. IEE Proceedings: Software, 149(1), p 24-39. fev. 2002.

SCACCHI, Walt. **When is Free/Open Source Software Development Faster, Better, and Cheaper than Software Engineering?** In: Koch, S. (ed.), Free/Open Source Software Development. IDEA Publishing Group, 2004b.

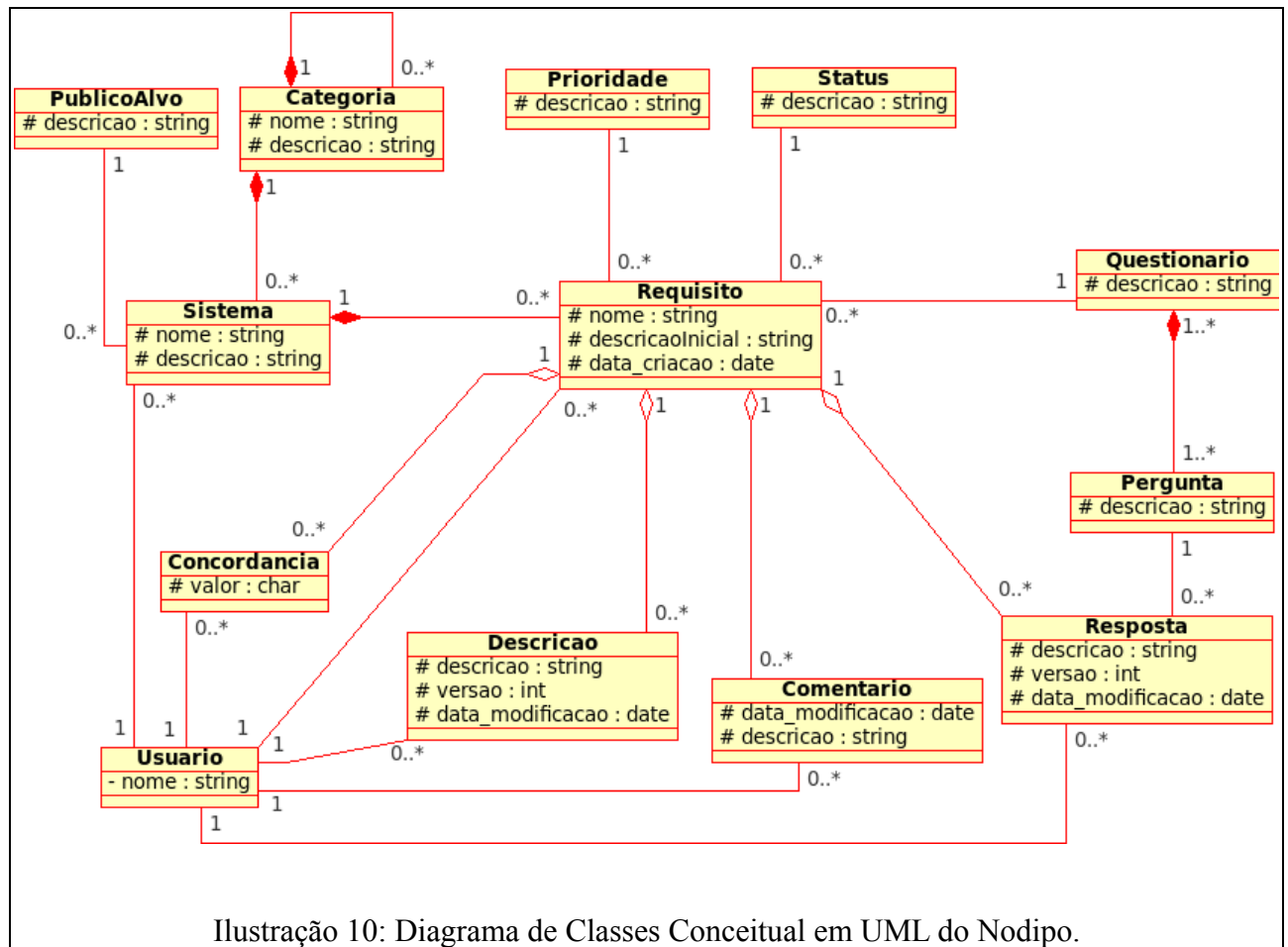
SOFTEX; INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Software Livre nas Prefeituras Brasileiras: Novas Alternativas para a Informatização da Administração Pública**. 2005. Disponível em <http://www.viadigital.org.br/index.php?option=com_docman&task=doc_download&gid=26>. Acesso em: 08 de maio de 2010.

THE SOFTWIKI PROJECT. **Social Requirements Engineering**. Disponível em <<http://softwiki.de/Project>>. Acesso em: 08 de maio de 2010.

TOGNERI, Denise Franzotti; FALBO, Ricardo de Almeida; MENEZES, Crediné Silva de. **Supporting Cooperative Requirements Engineering with an Automated Tool**. Anais do IV Workshop em Engenharia de Requisitos, Valencia, Espanha, 2002.

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE. **Quality Sense Making**. Disponível em: <<http://repositorio.viadigital.org.br/include/frame/comousar/Anexo1.pdf>>. Acesso em: 08 de maio de 2010.

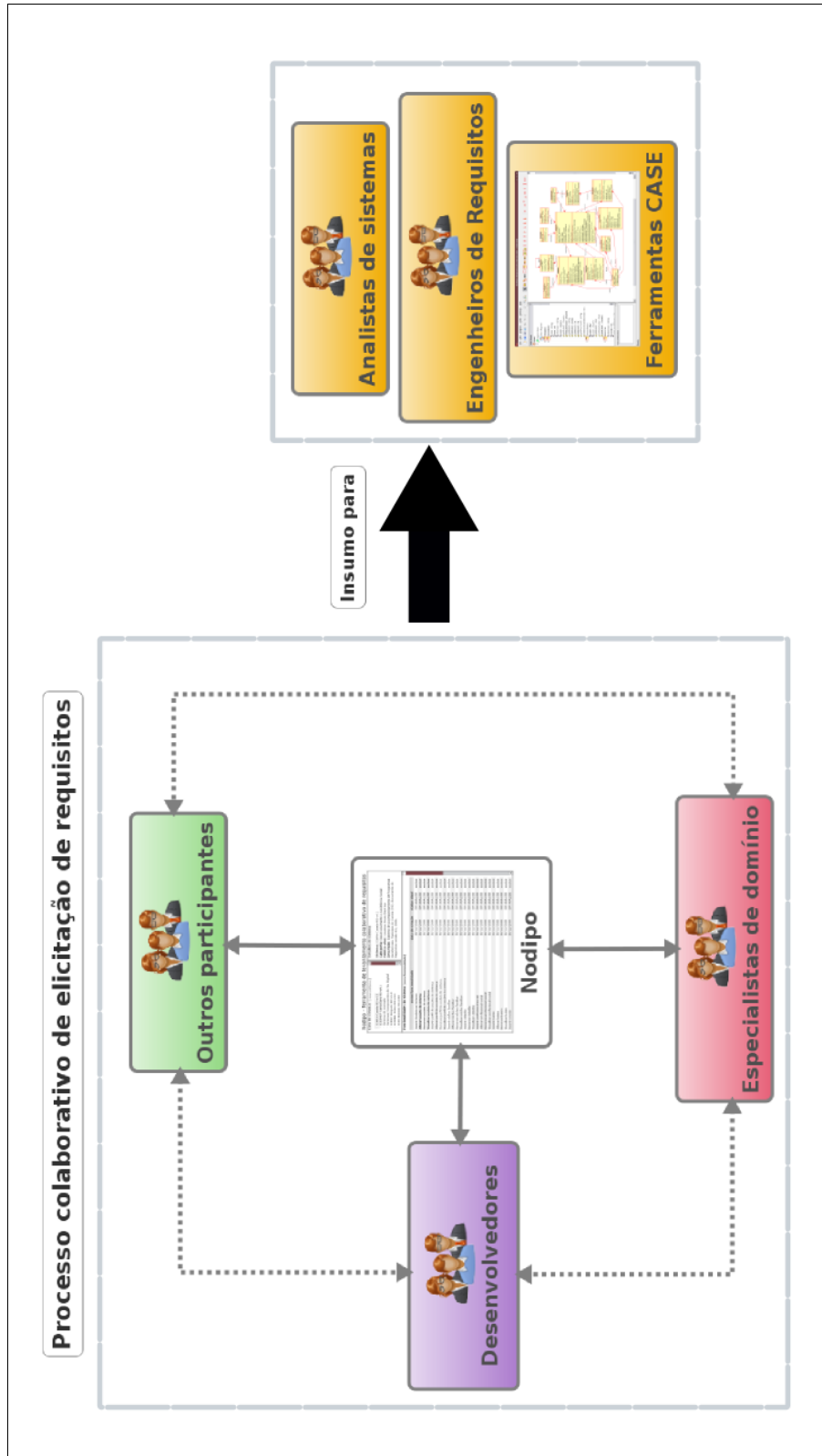
APÊNDICE A – DIAGRAMA DE CLASSES



APÊNDICE B – CATEGORIAS E SUBCATEGORIAS DE SOFTWARE

- Administração, Planejamento e Finanças
 - Contábil e Financeiro
 - Gestão de Recursos Logísticos
 - Planejamento e Orçamento
 - Recursos Humanos
- Desenvolvimento e Assistência Social
- Documentação
- Educação, Cultura, Esportes e Lazer
 - Gestão de Escolas
 - Programas Educativos
- Infraestrutura
 - Controle de Leis
 - Interação com legislativo
- Legislação
 - Controle de Leis
 - Interação com legislativo
- Obras e urbanismo
 - Meio-ambiente e saneamento
 - Planejamento urbano
- Outra
- Outros serviços públicos
 - Bibliotecas
 - Cemitérios
- Ouvidoria
- Protocolo e Tramitações de Documentos
 - Protocolo
- Saúde
 - Centros Hospitalares
 - Postos de Saúde
- Segurança pública
- Transporte
 - Trânsito

APÊNDICE C – O PROCESSO COLABORATIVO DE ELICITAÇÃO



APÊNDICE D – EXIBIÇÃO DE FUNCIONALIDADE



Nodipo - ferramenta de levantamento colaborativo de requisitos

SIPAF-Amparo-SP-v4.0 [versão para impressão] [fechar]

Funcionalidade: Visualizar usuário do sistema [mudar prioridade] **Prioridade:** Alta

Autor: Anônimo

Descrição [adicionar descrição]
Funcionalidade de visualização de cadastro de usuário do sistema.

Concordância
 --
 
 0 0

Status: EM ANALISE [mudar status] **Data de criação:** 12/12/2009

[ver questionário](#) [ver debate](#)

Questionário

Quem são servidos por esse processo?[responder]
Usuário Administrador, Usuário Responsável pela Frota e Usuário Gestor.
 Anônimo, 12/12/2009 às 18:07:06h [editar] [excluir]

Como começa este processo?[responder]
Esta funcionalidade começa quando um usuário deseja visualizar os dados de um usuário no sistema. Primeiramente, devem ser listados todos os usuários do sistema (através do requisito "Relatório de todos os usuários do sistema") e, ao escolher um dos listados, será obtida a visualização de um conjunto de informações referentes a um determinado cadastro de usuário do sistema.
 Anônimo, 12/12/2009 às 18:07:06h [editar] [excluir]

Como termina este processo?[responder]
Como resultado, tem-se a visualização de um cadastro de usuário do sistema.
 Anônimo, 12/12/2009 às 18:07:06h [editar] [excluir]

Quem participa do processo?[responder]
Usuário Administrador, Usuário Responsável pela Frota e Usuário Gestor.

ANEXO A – SOBRE AS FERRAMENTAS SIMILARES

A.1 ENGENHARIA DE REQUISITOS BASEADA EM SEMÂNTICA

No estudo feito sobre ferramentas similares à deste trabalho, encontrou-se o projeto SoftWiki, que, resumidamente, agrega funcionalidades semânticas a uma ferramenta de Engenharia Social de Requisitos.

O projeto é focado no que os idealizadores chamam de colaboração semântica relativa à Engenharia de Requisitos (LOHMANN et al., 2008a) ou Engenharia de Requisitos baseada em semântica (RIECHERT et al., 2007). No projeto, tecnologias da Web Semântica são utilizadas para dar apoio à colaboração e servem como meio de compartilhamento e interligação de requisitos, de forma a reduzir as dificuldades inerentes ao processo de levantamento distribuído e colaborativo de requisitos (LOHMANN et al., 2008a).

A.1.1 Engenharia Social de Requisitos e o SoftWiki

A equipe do SoftWiki considera a Engenharia Social de Requisitos como um caso especial de Engenharia do Conhecimento ágil e distribuída (THE SOFTWIKI PROJECT, 2010). Para dar suporte a esse caso especial, a equipe desenvolveu uma ferramenta que é baseada nos paradigmas da Web Semântica e da Web 2.0, que são (THE SOFTWIKI PROJECT, 2010):

- a) Uma base de conhecimento utilizada como um mapa de informações, com diferentes visualizações;
- b) Criação de conteúdo semântico intuitiva e baseada no conceito de ferramentas *wiki*;
- c) Colaboração entre os participantes:
 - Registro de alterações;
 - Possibilidade de se discutir e comentar cada parte dos requisitos;
 - Possibilidade de os participantes votarem e medirem a popularidade do

conteúdo;

- Valor às atividades dos usuários.

d) Estratégias de busca com melhorias semânticas.

Essas técnicas são aplicadas para se diminuir a barreira de entrada dos envolvidos em colaborar usando tecnologias semânticas (THE SOFTWIKI PROJECT, 2010).

Resumindo, no espírito da Web 2.0 o SoftWiki implementa uma arquitetura de participação a qual permite aos usuários agregarem valor à aplicação que utilizam (THE SOFTWIKI PROJECT, 2010).

A.1.2 A Engenharia de Requisitos e os padrões de representação da Web Semântica

Hall et al. (2002, apud RIECHERT et al., 2007) lembram que a importância da Engenharia de Requisitos foi identificada por inúmeros estudos como um fator crítico de sucesso de desenvolvimento de projetos. Tal constatação advém da necessidade de os diferentes participantes envolvidos em um projeto de desenvolvimento estabelecerem uma base comum de terminologias, bem como objetivos, cenários e requisitos que são expressos utilizando essas terminologias.

Disso, dentro do que os idealizadores do SoftWiki chamam de **iniciativa Web Semântica**, surgiram diversos padrões para a criação e o uso de terminologias, representados na forma de redes semânticas, taxonomias e ontologias. Ao passo que o desenvolvimento de projetos e de software tornam-se mais distribuídos, de modo que os participantes geograficamente distantes têm de ser integrados no ciclo de desenvolvimento (RIECHERT et al., 2006 apud RIECHERT et al., 2007), esses padrões de representação de conhecimento podem formar bases sólidas para a elicitação, representação, estruturação e gerenciamento de informações relevantes sobre requisitos (RIECHERT et al., 2007).

Por fim, as representações semânticas das informações sobre os requisitos obtidas através do SoftWiki podem servir posteriormente como insumo para a integração em diversas ferramentas de desenvolvimento de projetos (RIECHERT et al., 2007), tais como softwares de gerenciamento de projetos ou ferramentas CASE – Computer-Aided Software Engineering.

A.1.3 Arquitetura do SoftWiki

A ilustração 13 – vide anexo B – mostra a arquitetura do SoftWiki, da qual o cerne é a interação entre os usuários-finais do sistema e os desenvolvedores. Observando a arquitetura, vê-se que os usuários podem participar diretamente do desenvolvimento da base de conhecimento sobre os requisitos, enquanto os desenvolvedores atuam como moderadores das atividades de colaboração e dão suporte aos usuários.

Nesta arquitetura, o paradigma *wiki* (LEUF; CUNNINGHAM, 2001 apud LOHMANN et al., 2008a) é utilizado, cuja filosofia é “facilitar o conserto dos erros, ao invés de tornar difícil cometê-los”. A colaboração acontece utilizando-se funcionalidades comuns de ambientes *wiki*, como os mecanismos de controle de revisão, que possibilitam identificar, revisar e reverter alterações, bem como a facilidade para se discutir sobre os requisitos. Dessa forma, os participantes podem colaborativamente expressar seus conhecimentos em linguagem natural, sem qualquer restrição formal (LOHMANN et al., 2008a).

A camada semântica da arquitetura é adicionada – via SWORE, a ser apresentada a seguir – ao serem relacionados os conteúdos a classes e a instâncias ontológicas (LOHMANN et al., 2008b).

Ademais, a base de conhecimento criada colaborativamente no SoftWiki pode ser disponibilizada para outras atividades de desenvolvimento via adaptadores, como mostra a ilustração 13 do Anexo B, o que resulta em um fluxo de conhecimento que frequentemente acompanha o ciclo de desenvolvimento de software (LOHMANN et al., 2008b).

A.1.4 SWORE

A estrutura semântica do projeto é provida pela Ontologia para Engenharia de Requisitos SoftWiki – SWORE, do inglês SoftWiki Ontology for Requirements Engineering – que foi desenvolvida de acordo com práticas estabelecidas da Engenharia de Requisitos (LOHMANN et al., 2008b).

A SWORE apresenta esquemas ontológicos para instâncias do tipo objetivo, cenário e requisito – do inglês, respectivamente, *goal*, *scenario* e *requirement*. Essas instâncias têm

relações do tipo “definido por” – *defined by* – com os participantes – *stakeholders* – e são interconectadas pelas relações “detalhes” e “detalhado por” – respectivamente, *details* e *detailed by* –, conforme demonstra a ilustração 14, vide Anexo B. Além disso, a SWORE prevê a ligação de requisitos a conceitos do domínio em questão ou a partes de modelos de sistemas já existentes através das relações do tipo “refere-se a” – *refers to* – (LOHMANN et al., 2008b).

A.1.5 Resultados do projeto SoftWiki

Quanto à aplicação da ferramenta proposta pelo projeto SoftWiki, Lohmann et al. (set. 2008) afirmam que as experiências de utilização da mesma indicam muitos benefícios em relação à Engenharia de Requisitos não semântica, como por exemplo: mais fácil detecção de conflitos e de dependências entre requisitos, além de melhores meios para compartilhá-los.

A.2 COOPERATIVE REQUIREMENTS ENGINEERING SUPPORT TOOL – CRETA

A ferramenta CRETA surge em um contexto de estudos sobre o auxílio que a área de pesquisa em Trabalho Cooperativo Apoiado por Computador – do inglês Computer-Supported Cooperative Work, CSCW – pode dar à Engenharia de Requisitos (TOGNERI et al., 2002). Na prática, a proposta da ferramenta CRETA trata da agregação de funcionalidades de aplicações de *groupware* a aplicações que dão suporte à Engenharia de Requisitos (TOGNERI et al., 2002), com o objetivo de integrar engenheiros de requisitos, especialistas de domínio, usuários, gerentes de projeto e patrocinadores envolvidos no processo de desenvolvimento de software.

A área de pesquisa de CSCW estuda basicamente o trabalho em grupo. Mais especificamente, os estudos dessa área são direcionados à natureza das organizações e dos ambientes de trabalho, abordando desde análises sociológicas e antropológicas do trabalho a tecnologias que dão suporte ao trabalho em grupo (GRUDIN, 1994 apud TOGNERI et al., 2002). Nesse contexto, *groupware*, segundo Chaffey (1998 apud TOGNERI et al., 2002), é o

termo que designa as aplicações de CSCW desenvolvidas para promover o trabalho em grupo de modo eficiente e eficaz.

O conceito de cooperação bastante presente na ferramenta CRETA advém justamente da área de pesquisa de CSCW, a qual define cooperação como o esforço coletivo para se atingir um objetivo comum (NITZKE et al., 2002 apud TOGNERI et al., 2002). Definição essa que é atrelada à Engenharia de Requisitos no artigo de Togneri et al. (2002), a qual os autores consideram como um processo cooperativo.

A arquitetura da CRETA conta com 5 componentes (TOGNERI et al., 2002):

- a) Componente de processos, recursos e de modelo de conhecimento: é utilizado pelo gestor de conhecimento para registrar o conhecimento organizacional, incluindo os processos, recursos e modelos padrões da Engenharia de Requisitos. Os engenheiros de requisitos e os gerentes de projeto usam este componente para fazerem consultas à memória organizacional;
- b) Componente de definição de processo: é utilizado pelo gerente de projeto para o registro dos elementos que compõem os projetos, bem como para a definição e adaptação do processo de Engenharia de Requisitos para um projeto específico. Os patrocinadores, os engenheiros de requisitos e os especialistas de domínio utilizam este componente através de consultas apenas;
- c) Componente de Engenharia de Requisitos: é usado pelos participantes do projeto para as atividades de elicitação, análise, negociação, documentação, validação de especificação e gerenciamento de requisitos;
- d) Componente de trabalho cooperativo: permite aos usuários utilizar funcionalidades de *email*, *chat*, fórum, agendamento de compromissos, listas de discussão, agendamento de atividades em grupo, bem como visualizar os participantes conectados;
- e) Componente de rastreabilidade de projeto: utilizado pelo patrocinador e pelo gerente de projeto para gerenciar o curso das atividades.

Pela data de publicação do artigo de Togneri et al. (2002), vê-se que a utilização de ferramentas computacionais de apoio à Engenharia de Requisitos de forma distribuída já existe há alguns anos. Além disso, no mesmo artigo são apontadas outras ferramentas relacionadas, obviamente já existentes na época, mas que não agregam funcionalidades de *groupware*, ou seja, não dão suporte à cooperação.

A.3 CODIPSE-REQ

Similar ao projeto CRETA, o CODIPSE-Req também se trata de uma fusão de aplicativos: um de suporte ao trabalho cooperativo e distribuído, o Cooperative and Distributed Process Support Environment – CODIPSE –, e o outro de *groupware*, o eGroupware (BRITO et al., 2006). Segundo Brito et al. (2006), a fusão de ambas ferramentas possibilita o suporte efetivo à colaboração na etapa de Engenharia de Requisitos com equipes distribuídas.

De uma avaliação de 23 ferramentas de gerenciamento de requisitos existentes, Brito et al. (2006) concluíram que a maioria delas não fornece o suporte adequado para a comunicação e colaboração, o que, segundo os autores, dificulta a adoção dos mesmos por equipes distribuídas. Fator esse que, inclusive, motivou a proposta e o desenvolvimento da ferramenta CODIPSE-Req.

É interessante constar que há no artigo de Brito et al. (2006) uma crítica à ferramenta CRETA. Os autores afirmam que esta e outra ferramenta chamada Teamwave “não tratam diretamente sobre desenvolvimento com equipes distribuídas e não mencionam como o rastreamento de requisitos deve ser tratado” (BRITO et al., 2006).

A aplicação CODIPSE-Req possui três módulos principais:

- (1) Gerenciamento de Requisitos: agrupando funcionalidades para manutenção de projeto como visão/escopo, casos de uso, requisitos funcionais e não funcionais, glossário e atores;
 - (2) Manutenção do Rationale: agrupa funcionalidade para criar links com as ferramentas de colaboração do eGroupware. Os links podem ser criados em nível de projeto, casos de uso ou requisitos, permitindo a completa manutenção do rationale de requisitos;
 - (3) Exportação: esse módulo é responsável pela geração intermediária (em XML) dos dados presentes em um determinado projeto. A partir dessa versão e através de transformação XSL é possível exportar as informações para qualquer formato. [...]
- (BRITO et al., 2006)

Por *rationale* os autores entendem a ligação dos requisitos a documentos, mensagens de fóruns, entre outros elementos do CODIPSE-Req, que possibilitam, por exemplo, identificar as decisões que levaram à especificação de um requisito (BRITO et al., 2006).

Em relação às funcionalidades referentes à Engenharia de Requisitos, os seguintes recursos são apresentados – que, em termos da Engenharia de Software, são casos de uso ligados ao ator engenheiro de requisitos:

R1 - Manter Projeto: O sistema possibilita a inclusão, alteração e exclusão de diversos projetos. Para cada projeto, deve-se identificar o seu nome, visão, usuários, casos de uso, requisitos e glossário.

R2 - Manter Usuário: O sistema possibilita a inclusão, alteração e exclusão de usuários (ou atores) que participam do sistema.

R3 - Manter Requisito: O sistema possibilita a inclusão, alteração e exclusão de requisitos funcionais e não funcionais.

R4 - Manter Casos de Uso: O sistema possibilita a inclusão, alteração e exclusão de casos de uso, identificando os atores que os realizam, pré-condições, pós-condições, fluxo principal e fluxo alternativo.

R5 - Manter Glossário: O sistema possibilita a inclusão, alteração e exclusão de termos para compor o glossário do sistema.

R6 - Manter Rationale: O sistema deve permitir o registro de rationale (motivo, razão), para que seja possível identificar as decisões que levaram a definição do estado atual da especificação de requisitos. Para isso, o sistema permite a inclusão, alteração e exclusão de links (ligações entre os registros) em nível de registros de projeto, casos de uso ou requisitos, com qualquer informação que tenha originado modificações. Por exemplo, um projeto deve apresentar links para arquivos de documentos que descrevem sistemas relacionados que tenham sido utilizados na especificação e mensagens em fóruns onde se tenha discutido algo sobre o sistema, entre outros. (BRITO et al., 2006)

Em relação às funcionalidades de *groupware*, o CODIPSE-Req conta com os seguintes recursos (BRITO et al., 2006):

- a) Manter base de conhecimento – ligado ao ator engenheiro de conhecimento;
- b) Manter fórum – ligado ao ator *stakeholder*, ou seja, aos demais participantes;
- c) Manter agenda – ligado ao ator *stakeholder*;
- d) Manter tarefa – ligado ao ator *stakeholder*;
- e) Manter solicitação de mudança – ligado ao ator *stakeholder*;

Depois de apresentar um estudo de caso com um protótipo para demonstrar os benefícios de utilização do CODIPSE-Req, Brito et al. (2006) alegam que são necessários mais estudos de caso e/ou experimentos para a validação do mesmo.

ANEXO B – DIAGRAMAS DO SOFTWIKI

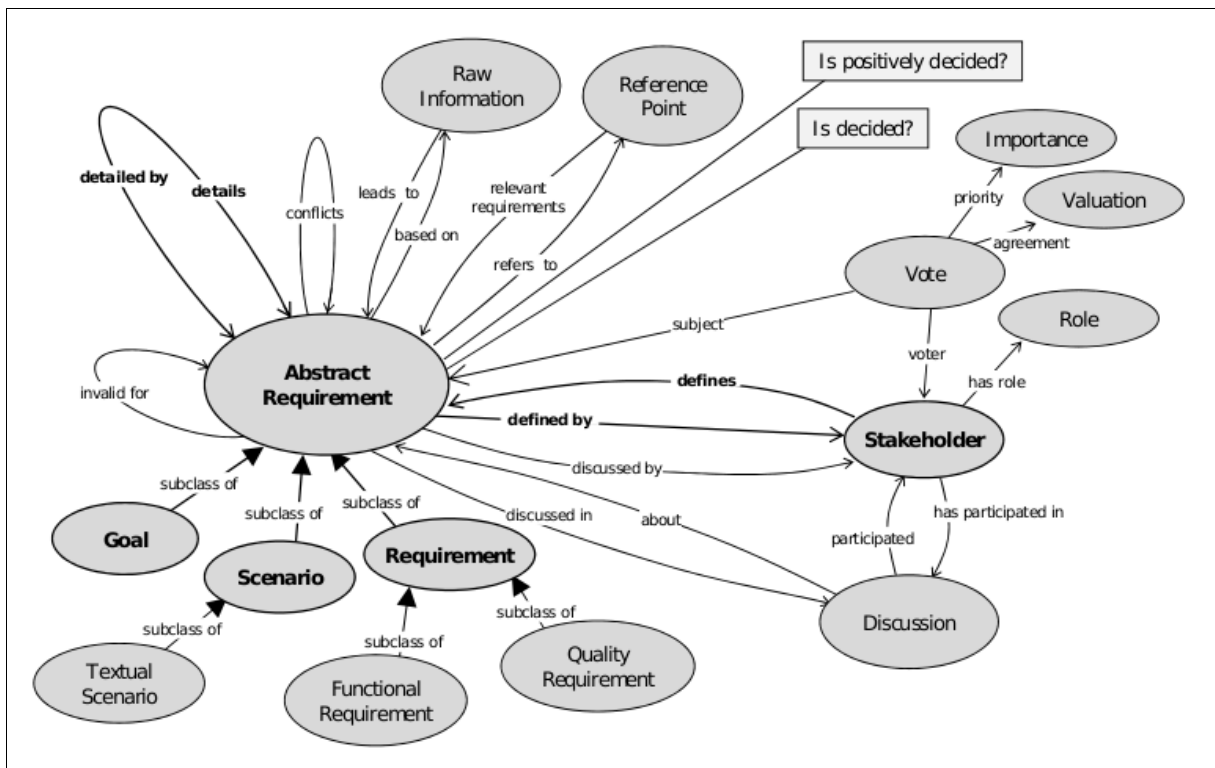
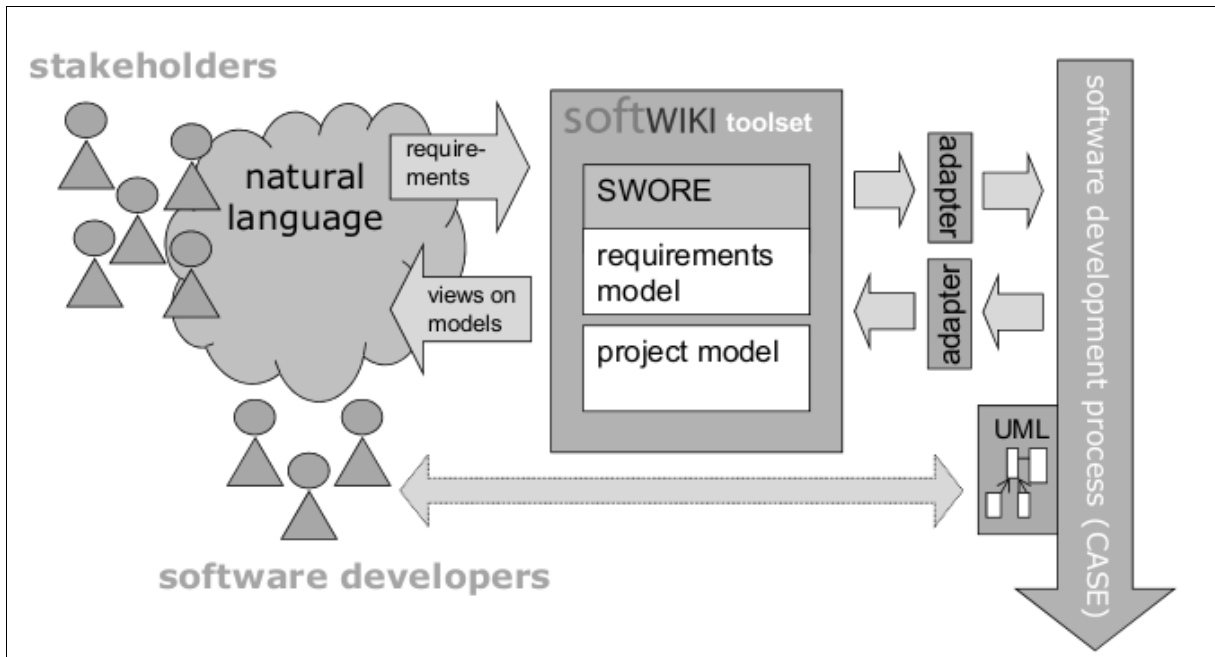


Ilustração 14: A ontologia para Engenharia de Requisitos SoftWiki. SoftWiki Ontology for Requirements Engineering – SWORE.