

# Nodipo: ferramenta de levantamento colaborativo de requisitos para software livre

José Eduardo De Lucca<sup>1</sup>, Yuri Gomes Cardenas<sup>2</sup>

<sup>1</sup> Departamento de Informática e Estatística

<sup>2</sup> Curso de Bacharelado em Sistemas de Informação

<sup>1,2</sup> Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brazil

*delucca@inf.ufsc.br, yuri@geness.ufsc.br*

**Abstract.** *This paper introduces a collaborative requirements elicitation tool, called Nodipo, which aims to fill the gap of some areas where free software is not or is too little available. Based on Web 2.0 and on researches that focus on the enduser participation on software design and requirements elicitation, the tool's structure is made of elements that make possible the interaction and collaboration between endusers and other stakeholders, so that they are able to share knowledge about the application domain and elicit requirements with certain formalization degree.*

**Resumo.** *Este artigo apresenta a ferramenta de levantamento colaborativo de requisitos Nodipo, com a qual se objetiva suprir áreas de aplicação na sociedade em que o software livre é ausente ou pouco presente. Baseada na Web 2.0 e em estudos que levam em conta a participação de usuários-finais no processo de elicitação de requisitos e na modelagem de software, a estrutura da ferramenta possui elementos que viabilizam a interação e a colaboração entre os próprios usuários-finais e os demais participantes, de forma que os mesmos possam compartilhar conhecimento sobre o domínio da aplicação e elicitar requisitos com um nível mínimo de formalização.*

## 1. ESTRUTURA DO ARTIGO

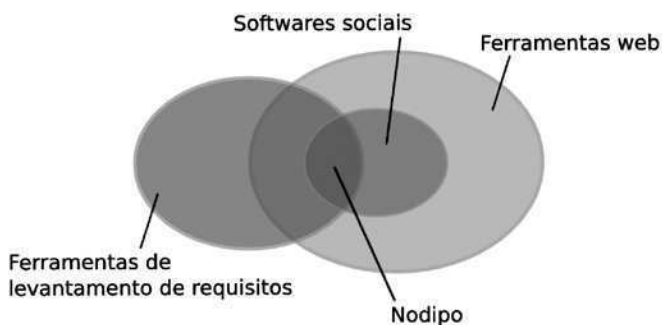
Neste artigo serão abordadas algumas das teorias que fundamentam a ferramenta apresentada, tais como a Engenharia Social de Requisitos, a Modelagem Sociotécnica de Software e a modelagem contínua inerente às comunidades de Software Livre. Em seguida, é explanado o contexto em que a proposta do Nodipo surgiu e é detalhada a proposta, a qual é complementada por uma descrição do processo de levantamento de requisitos pensado pelos idealizadores da ferramenta. Por fim, são apresentadas as conclusões e indicado o direcionamento de trabalhos futuros do tema em questão.

## 2. ENGENHARIA SOCIAL DE REQUISITOS E MODELAGEM SOCIOTÉCNICA DE SOFTWARE

Hammouda *et al.* (2008 apud LOHMANN *et al.*, 2009) definem a Engenharia Social de Requisitos 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*.

Dentro da concepção de que softwares de apoio à 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 ambiente de Engenharia de Requisitos. Também apelidadas como software social, essas aplicações *web* focadas em comunidades possuem diversas características (HIPPER, 2006 apud LOHMANN *et al.* 2009), tais como simplicidade, colaboração ágil, auto-organização, *feedback* social e foco em usuários e em comunidades.

Essas características, conforme Lohmann *et al.* (2009), são fundamentais nos sistemas para Engenharia de Requisitos que dão suporte à Engenharia Social de Requisitos. Ainda segundo esse autor, a aplicação dos conceitos de software social em uma aplicação *web* pode encorajar *stakeholders* não-convencionais a participar do processo de modelagem de software. A figura 1 ilustra os conjuntos de softwares citados e suas interseções, indicando a área onde se situa a ferramenta Nodipo.



**Figura 1: Interseção entre categorias de software**

Scacchi (2004), por sua vez, alerta que a participação de usuários não-treinados na modelagem de software pode ser frustrante. Contudo, o autor apontava como tendência 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.* (dez. 2003), apresentada adiante.

Em relação à Modelagem Sociotécnica de Software, segundo Scacchi (2004), ela está intimamente ligada à participação direta dos usuários-finais no processo de modelagem de sistemas. Esse tipo de abordagem distingue-se dos métodos de modelagem da Engenharia de Software tradicional, que dão maior ênfase às atividades e responsabilidades dos engenheiros de software, os quais utilizam ferramentas (computacionais) de modelagem e notações para obter e formalizar os resultados deste tipo de processo.

### 3. 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, que são contrárias às práticas convencionais de desenvolvimento de software para sistemas de grande porte. São elas:

- As especificações das funcionalidades, da usabilidade e da estrutura do software não são conhecidas quando de sua modelagem e desenvolvimento. Ao invés disso, artefatos capturam e rastreiam continuamente as preferências dos usuários das comunidades, o que forma um ciclo permanente 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.
- 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, fev. 2002), no lugar de processos e especificações formais. Esse conjunto é constituído de ferramentas de compartilhamento de conhecimento semi-estruturadas e relativamente informais, representações e práticas.

Além disso, o autor considera 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.

Portanto, ao compreender-se a modelagem contínua de projetos de software livre que define 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. Mas realizam, sim, levantamento e especificação contínuos, coletiva e colaborativamente.

#### 4. ESCASSEZ DE SOFTWARES LIVRES EM DETERMINADAS ÁREAS

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

Considerando a taxonomia de categorias de software proposta pelos autores, os mesmos afirmam que a disponibilidade está claramente nas categorias de software básico e de infraestrutura, no setor de ferramentas para desenvolvimento de software e em outros setores específicos como comunicação e ferramentas básicas de produtividade. Sobre exemplos em outras áreas, Savi e De Lucca (2007) afirmam que existem, mas isoladamente e a partir de esforços específicos, e em nada comparáveis à presença na categoria de softwares básico e de infraestrutura, por exemplo.

De uma forma geral, por parte da comunidade de desenvolvedores de softwares livres, existe maior interesse em fazer parte de projetos de software básico ou de softwares com grande visibilidade – dos quais esses desenvolvedores também são usuários –, tais como editores de texto e navegadores [SAVI; DE LUCCA, 2007]. A visão de Scacchi (2002) colabora com essa constatação, pois entende que os projetos de software livre geralmente contam com usuários que também são desenvolvedores dos mesmos softwares e, por isso, conhecem bem os requisitos, as dificuldades, os recursos ausentes e os *roadmaps* dos mesmos e passam a implementar as funcionalidades que desejam, pessoal ou coletivamente, nos programas.

Do 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 conta-

to 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 confirma Nakawaga (2006).

## 5. 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: o Nodipo.

Disso, foi modelada e desenvolvida uma ferramenta colaborativa em que usuários sem experiência em desenvolvimento de software podem autonomamente expressar suas necessidades em relação a determinado sistema. O Nodipo oferece condições para descrição de processos e demandas, inicialmente utilizando o discurso e conceitos do usuário especialista do domínio, que podem ser refinados iterativamente com a participação de outros usuários e com a intervenção de desenvolvedores para o esclarecimento de dúvidas e construção de um nível mínimo de formalização.

Baseado na *web*, o Nodipo possui forte viés de colaboração assíncrona – seguindo tendências da Web 2.0 –, de modo 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 na compreensão do sistema desejado. Dessa forma, o Nodipo tem, em sua gênese, o conceito de rede social; sendo, entretanto, uma rede de propósito específico, focada em levantamento colaborativo de requisitos.

Dentro de sua meta, o Via Digital pensou os artefatos resultantes desse processo 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.

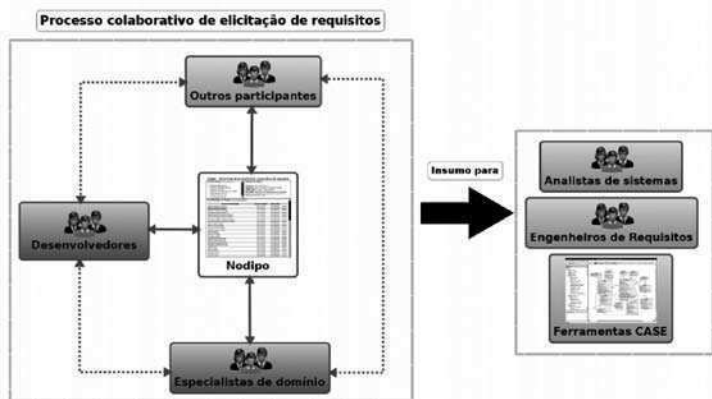
## 6. DESCRIÇÃO DO PROCESSO DE LEVANTAMENTO DE REQUISITOS NO NODIPO

A mais notória função do Nodipo é atuar como meio integrador de diferentes participantes no processo de elicitação 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 elicitação com o Nodipo. Para melhor compreensão desse processo de interação e de levantamento colaborativo de requisitos com o Nodipo, a figura 2 e o seguinte cenário ilustram a utilização ferramenta:

- Especialistas de determinado domínio de aplicação têm uma demanda por um software que automatize os processos de sua área de atuação. O primeiro pas-

so destes especialistas é descrever, no Nodipo, características e funcionalidades que imaginam que o software deveria apresentar, conduzidos por um questionário configurável. A função desse questionário é obter a descrição de todas as etapas de um processo organizacional específico, bem como dos atores e suas atuações. A modelagem possibilita a utilização de diferentes abordagens e técnicas de entrevistas para a descrição de processos. No Nodipo está implementado o método *Quality Sense Making*.

- Ato contínuo, as diversas visões e processos dos diferentes especialistas conduzem a um debate sobre as características e funcionalidades cadastradas, inclusive gerando a descrição de novas características.
- Os ciclos de debate e o exercício de reescrita das descrições as enriquecem continuamente, conduzindo a um potencial consenso quanto às mesmas, inclusive com possibilidades de votações.
- A atividade avança até o momento em que os participantes concordam que as descrições dadas às funcionalidades representam o que desejam.



**Figura 2: O processo de levantamento colaborativo de requisitos com o Nodipo**

O resultado do cenário apresentado é o conhecimento dos especialistas do domínio explicitado e disponível. A participação de desenvolvedores durante os debates pode levar a uma convergência mais rápida e/ou uma definição final mais próxima da semântica da Engenharia de Software. Já no caso de uma interferência posterior por parte de um desenvolvedor, em que ele reescreve as definições dadas, a partir de sua compreensão de todo o histórico do debate, é possível fazer uma segunda rodada de debates e/ou de votações, como forma de validação da interpretação do desenvolvedor, transformando as descrições em requisitos. Outro possível passo subsequente (este não contemplado pelo Nodipo) seria a atuação de analistas de sistemas/engenheiros de software sobre o resultado para a geração de especificações mais formais de requisitos com ferramentas tradicionais.

Dado o cenário apresentado, em resumo o Nodipo foi modelado de maneira que seja possível:

- Promover um meio de debate em relação a requisitos e características de sistemas, aproximando a comunidade FLOSS dos sistemas aplicativos e de desenvolvedores com especialistas de domínio;

- Obter informações semiformais dos especialistas do domínio de aplicação, através dos debates, consolidados nas respostas ao questionário e nas novas versões das descrições gerais das funcionalidades;
- Observar a satisfação dos participantes quanto à elicitação de determinado requisito, com a votação de concordância.

## 7. CONCLUSÃO

Para procurar suprir algumas áreas de aplicação em que o software livre é pouco presente foi criada a ferramenta Nodipo, cuja modelagem possui elementos característicos da Engenharia Social de Requisitos e de softwares sociais. Esses elementos são propícios para fomentar a interação entre integrantes das comunidades FLOSS e potenciais usuários/clientes, de forma que os mesmos possam colaborativamente trocar conhecimentos, descrever os processos dos sistemas e elicitar os requisitos necessários a estes.

Imaginando um cenário em que a ferramenta apresentada neste artigo é utilizada em ampla escala, atuando como referência de repositório de requisitos de sistemas, seriam especificados muitos dos softwares das áreas de aplicação nas quais o FLOSS não é presente, oferecendo oportunidades para a atuação de profissionais e a criação de empresas nacionais no desenvolvimento, manutenção e suporte desses softwares.

Academicamente falando, a ferramenta pode servir como ambiente de avaliação para métodos de coleta de requisitos baseados em entrevistas e para evolução destes métodos em direção a uma engenharia de requisitos mais social (conforme definido).

Quanto a trabalhos futuros, parece factível e interessante a adição das características e implementações semânticas da ferramenta do projeto SoftWiki (THE SOFTWIKI) ao Nodipo – sendo aquele um projeto contemporâneo e de objetivos similares aos apresentados neste artigo –, o que tenderia a complementá-lo, uma vez que o Nodipo não conta com elementos semânticos em sua estrutura.

## REFERÊNCIAS

- GASSER, Les et alli. *Understanding Continuous Design in F/OSS Projects*. 16<sup>th</sup> International Conference on Software Engineering & its Applications (ICSSEA03), Paris, França, dezembro, 2003.
- LOHMANN, Steffen; DIETZOLD, Sebastian; HEIM, Philipp; HEINO, Norman. *A Web Platform for Social Requirements Engineering*. Workshopband, Alemanha, 2009.
- NAKAGAWA, Elisa Y. et alli. *Relevância dos requisitos no desenvolvimento de software livre*. VII Workshop Software Livre, Porto Alegre, 2006.
- SAVI, Rafael; De LUCCA, José Eduardo. *Produção compartilhada de software para governos municipais*. VIII Workshop de Software Livre, Porto Alegre, 2007.
- SCACCHI, Walt. *Understanding the Requirements for Developing Open Source Software Systems*. IEE Proceedings: Software, 149(1), p 2439. fev. 2002.
- SCACCHI, Walt. *SocioTechnical Design*. The Encyclopedia of HumanComputer Interaction. Berkshire Publishing Group, 2004.
- THE SOFTWIKI Project. Social Requirements Engineering. Disponível em <<http://softwiki.de/Project>>. Acesso em: 08 de maio de 2010.