

Análise de interações nas Comunidades Virtuais de Software Livre

Licia de Cassia Nascimento¹, Flávia Maria Santoro²

^{1,2}Departamento de Informática Aplicada

Universidade Federal do Estado do Rio de Janeiro (UNIRIO) – Rio de Janeiro, RJ -
Brasil

{licia.nascimento, flavia.santoro}@uniriotec.br

Abstract. *The way in which Free Software is developed is directly related to collaboration and knowledge exchange between pairs organized in Communities of Practice. The interaction and content of the record shared by the participants are key in this process. Therefore, providing mechanisms to support dissemination of knowledge produced during interactions reveals extremely important in this context. This paper presents a research conducted in order to analyze a range of communities and how they treat knowledge as the participants interact and how it is done to knowledge management. Based on this analysis is proposed a knowledge management model in free software communities.*

Resumo. *A forma como Software Livre é desenvolvido está diretamente relacionada à colaboração e troca de conhecimento entre pares organizados em Comunidades de Prática. A interação e registro do conteúdo compartilhado pelos participantes são fundamentais neste processo. Desta forma, prover mecanismos para apoio à disseminação do conhecimento mostra-se extremamente relevante. Este artigo apresenta uma pesquisa realizada com vistas a analisar um conjunto de comunidades e a forma como tratam o conhecimento, como os participantes interagem e como é realizada a gestão de seu conhecimento. Com base nesta análise, é proposto um modelo para gestão de conhecimento em comunidades de software livre.*

1. Introdução

O conceito de Software Livre (SL) está cada vez mais presente na sociedade atual. Software Livre pressupõe liberdade para usuários usarem, copiarem, estudarem, modificarem e redistribuírem um programa de computador sem restrições. A forma como estes programas são desenvolvidos está diretamente relacionada à colaboração e troca de conhecimento entre pares que estão organizados em Comunidades de Prática (CoP) e se comunicam através de ferramentas colaborativas disponíveis na Internet.

Em pesquisas recentes de Rudzki e Jonson (2003), von Krogh et al. (2005) e Sowe et al. (2008) observou-se que nem sempre os ambientes de suporte às comunidades virtuais de software livre provêm funcionalidades adequadas para a gestão de seu conhecimento. Essas pesquisas analisam o conhecimento por meio da espiral do conhecimento (Nonaka e Takeushi, 1997), das ferramentas de apoio à

reutilização de conhecimento em projetos de desenvolvimento e da troca de conhecimento existente que apoia os desenvolvedores.

Este artigo apresenta uma pesquisa realizada com vistas a analisar um conjunto de comunidades e a forma como manipulam o conhecimento, como os participantes interagem e como é gerido o conhecimento. O objetivo é propor um modelo que garanta as condições necessárias para suporte à gestão de conhecimento.

O artigo inicia na seção 2 abordando como a Gestão do Conhecimento acontece nas comunidades virtuais de software livre, segundo a literatura; a seção 3 apresenta uma análise com foco nos mecanismos de Gestão do Conhecimento que acontecem nestas comunidades; a seção 4 apresenta um modelo de solução; e a seção 5 apresenta a conclusão.

2. Gestão do Conhecimento em Comunidades Virtuais de Software Livre

O software livre, conforme a definição da Free Software Foundation (2008), “é o software que dá ao usuário a liberdade para compartilhar, estudar e modificar seu código. Este software é chamado de livre, porque o usuário está livre.” (GNU, 2008).

O estilo de desenvolvimento pode ser Catedral ou Bazar, (Raymond, 1998). O estilo Catedral, centralizado e utilizado por uma equipe de desenvolvedores, é o modelo de desenvolvimento típico para software proprietário. O autor cita como exemplo o projeto Emacs (Raymond, 1998). Já no estilo Bazar, adotado pelas comunidades de software livre, o código é desenvolvido de forma totalmente aberta e pública, utilizando a Internet, tendo como exemplo o projeto Fetchmail (Raymond, 1998).

Segundo Reis (2003), “Projetos de Software Livre são organizações virtuais formadas por indivíduos que trabalham juntos no desenvolvimento de um software livre específico. Estes indivíduos trabalham geograficamente dispersos, utilizando ferramentas simples para coordenar e comunicar seu trabalho através da Internet.” As comunidades virtuais de software livre vêm se difundindo baseadas na descentralização (apesar da distância geográfica e das diferentes agendas dos voluntariados), na colaboração, no uso de repositórios de desenvolvimento de projetos, na disseminação do conhecimento e no desenvolvimento tecnológico.

As comunidades virtuais de software livre têm como objetivos: divulgação, apoio e uso de ferramentas de software livre; disseminação do conhecimento do software livre; e apoio ao desenvolvimento de projetos de software livre. Essas comunidades são formadas por voluntários interessados em participar destes projetos. Seus mantenedores realizam a moderação e a organização da comunidade. A motivação para participar de uma comunidade, de acordo com Taurion (2004), está no reconhecimento, no prestígio e na reputação para com os seus pares, até porque não existe remuneração financeira envolvida.

Os ambientes onde as comunidades se organizam são sites da Internet, nos quais são divulgadas notícias e artigos de interesse. Os interessados se cadastram e participam da comunidade, os projetos e as tecnologias são anunciados, e o conhecimento é disseminado. A comunicação entre os participantes normalmente acontece por meio das listas de e-mail. A comunicação é informal e informativa para a divulgação de conhecimento e apoio aos projetos coletivos.

As comunidades virtuais de software livre podem ser consideradas Comunidades de Prática (CoP), (Wenger, McDermott e Snyder, 2002), sendo que o conhecimento de seus membros é seu maior valor. Desta forma, as atividades de captura, registro e recuperação de conhecimento, típicas de um processo de Gestão de Conhecimento são essenciais para que estas comunidades possam se desenvolver e estimular seus membros a continuar interagindo. Na visão de Terra (2007), a Gestão do Conhecimento (GC) “significa usar o poder transformador e democratizador das novas tecnologias em prol do aprendizado contínuo e coletivo. Significa, ademais, colocar o indivíduo no centro do processo de aprendizado, mas levando-se fortemente em consideração que seu aprendizado é indissociável da cultura, do social e do coletivo”.

No contexto das comunidades virtuais de software livre, GC está voltada para a transferência de conhecimento que ocorre entre os seus participantes e o reconhecimento do que o projeto desenvolvido proporcionará à sociedade. Pode-se afirmar que esta transferência de conhecimento está concentrada basicamente em duas vertentes: (i) nos artefatos gerados de forma colaborativa (códigos e documentação associada); e, (ii) nos resultados das interações e colaboração existentes entre os participantes dos projetos. A literatura relata pesquisas nestas duas direções. Em geral, CoP têm foco em um tema de interesse comum dos seus participantes que discutem sobre esse tema, já as comunidades virtuais de software livre além do tema de interesse comum e das interações, produtos coletivos são gerados colaborativamente (artefatos de software, i.e. código), e esse código pode ser modificado, produzindo novos artefatos, por qualquer participante a qualquer momento.

Rudzki e Jonson (2003) realizaram um trabalho de identificação e diagnóstico do que chamaram de “práticas em projetos de desenvolvimento de software livre relacionadas ao conhecimento”. As práticas analisadas foram: FAQs, Web Sites, Howtos, IRC, Sistemas de Monitoramento de Problemas, Mailing Lists, Weblogs e Wikis. Os projetos analisados foram: Apache, GCC, Gnome, KDE, Desenvolvimento do kernel do Linux e Mozilla. O método proposto incluiu a análise da codificação e personalização do conhecimento, e visão da Gestão de Conhecimento da Engenharia de Software. A codificação do conhecimento tem como foco as formas de transformação do conhecimento: exteriorização, combinação, internalização e socialização (Nonaka e Takeuchi, 1997). No ponto de vista da personalização, os autores Rudzki e Jonson (2003) observaram o tratamento do conhecimento em relação aos seguintes fatores: acurácia e valor, dinâmica, validade, simultaneidade, confidencialidade, acessibilidade e papéis, categorização e mapeamento, busca e localização especializada. As necessidades da Engenharia de Software relacionadas ao conhecimento e que estão presentes em todos os projetos de desenvolvimento estão em: adquirir conhecimento sobre novas tecnologias, acessar o conhecimento sobre um domínio, compartilhar conhecimento sobre políticas e práticas, identificar especialistas e fontes de conhecimento e colaborar.

Rudzki e Jonson (2003) observaram cada uma das práticas segundo os critérios anteriores, mas não existiu um aprofundamento em cada um deles. Por exemplo, uma vez que assuntos diferentes são tratados em canais diferentes de IRC, consideraram que esta prática resolve o problema de categorização e mapeamento de conhecimento. Apesar dos autores afirmarem que as práticas atendem ao ciclo da GC, elas ainda não provêm os mecanismos de suporte a este ciclo em níveis adequados.

Sowe et al. (2008) pesquisaram sobre o compartilhamento de conhecimento no

projeto Debian, onde listas de e-mail são utilizadas. O estudo foi feito através de métricas e foi discutida a tendência de compartilhamento nas listas, a correlação entre as atividades de postagem e resposta entre os participantes da lista e a natureza da auto-organização das atividades individuais de compartilhamento de conhecimento. Apesar de ser uma avaliação bastante pontual baseada em um projeto e um tipo de interação, é possível analisar as atividades de compartilhamento de conhecimento e entender e aprender a forma como o conhecimento é compartilhado e criado nos projetos de software livre, além de abrir possibilidades e questões de pesquisa nesta linha.

Von Krogh et al. (2005) analisaram 15 projetos de software livre para identificar as formas de reutilização de artefatos. O estudo foi feito por meio de entrevistas com desenvolvedores que responderam às seguintes questões: qual o conhecimento é reutilizado no desenvolvimento de software livre, e quais fatores influenciam na escolha dos conhecimentos a serem reutilizados? A conclusão foi que os desenvolvedores de software livre reutilizam o conhecimento em três circunstâncias: (i) por uma única linha de código, (ii) por componentes e (iii) por métodos. Além disso, o esforço necessário para a pesquisa, integração e manutenção do código impacta na forma como o conhecimento é reutilizado.

No contexto deste artigo, o foco está em analisar o nível de suporte aos participantes de comunidades virtuais de software livre na captura, registro e reutilização de conhecimento. Com isto, pretende-se observar aspectos mais específicos do que os verificados nas pesquisas descritas.

3. Estudo Exploratório

Foi realizado um estudo exploratório em duas fases: (i) análise de ambientes de suporte a comunidades de software livre e (ii) levantamento de impressões de participantes das comunidades através de um questionário. O objetivo foi caracterizar as interações que ocorrem entre os participantes de uma comunidade virtual de software livre, com foco nas atividades de disseminação, busca, armazenamento e troca de conhecimento. A pesquisa teve foco na identificação dos mecanismos existentes para apoio à gestão de conhecimento e pontos de deficiência, de forma a prover insumo para proposta de um modelo para apoio à GC em projetos de desenvolvimento de software livre.

3.1. Primeira Parte: Análise dos Ambientes de Apoio

O método de pesquisa utilizado foi composto das seguintes etapas: identificação das comunidades virtuais de software livre com participação significativa e projetos de porte na Internet, acesso às ferramentas usadas, pequenas inserções nestas comunidades para observar como funcionam, definição de critérios específicos para análise, avaliação dos critérios definidos em cada comunidade, organização dos resultados em tabela comparativa e interpretação dos resultados. Os critérios de análise foram definidos de forma a abranger desde o porquê da criação da comunidade até a forma de comunicação entre os seus participantes, para observar o ciclo de Gestão do Conhecimento. As comunidades selecionadas possuem focos diversos, ou seja, divulgação de tecnologia, compartilhamento de conhecimento, ferramentas e desenvolvimento de software livre, o que apóia o entendimento das diversas faces das comunidades.

3.1.1. Critérios de Análise

Cada comunidade foi identificada por meio de seu objetivo, projeto que está apoiando, ambiente de suporte (endereço da sua página na Internet). Os membros são os integrantes da comunidade, ou seja, os interessados no objetivo da comunidade. Estes membros podem ser profissionais, usuários, estudantes e professores. A comunidade é geralmente composta por um moderador e diversos participantes, onde todos contribuem de alguma forma para alcançar o objetivo da comunidade. As comunidades Under-Linux, VivaOLinux, Source Forge.net, FreshMeat.net, Tigris, Google Code e Koders (Tabela 1) foram selecionadas devido a terem como objetivo o desenvolvimento de projetos e não apenas divulgação de tecnologia livre.

A análise foi baseada nos seguintes critérios, além do Objetivo da comunidade:

- Mecanismos de apoio à interação e discussão: a comunicação é um dos alicerces para que a comunidade promova o conhecimento, já que a troca de informações é intensa e dinâmica e precisa atingir diversos participantes dispersos geograficamente. Os mecanismos de apoio à interação e discussão se referem às ferramentas usadas pelos membros da comunidade para se comunicarem e trocarem informações e questões sobre os projetos que estão desenvolvendo. Exemplos de mecanismos de interação e discussão são mensagens eletrônicas, chats, fóruns etc.
- Modelo de estruturação das interações: o modelo de estruturação das interações permite que a argumentação em uma discussão seja organizada e as mensagens trocadas categorizadas. Em modelos de argumentação tais como IBIS (Conklin, 1998) as mensagens são do tipo Questão, Argumento e Posição. Estes modelos podem ajudar na identificação de um conhecimento específico.
- Mecanismos de busca de informação: a busca por um determinado assunto ou artefato é frequente nas comunidades de prática, uma vez que as interações são em grande parte assíncronas. A forma mais simples é a busca por palavras-chave. Outras formas mais sofisticadas requerem taxonomias ou outros modelos de indexação associados aos itens que se deseja encontrar.
- Existência de repositório estruturado: as comunidades podem ter seus próprios repositórios com artigos, notícias, dicas e o código fonte das ferramentas desenvolvidas. Os repositórios criam uma Memória Organizacional (Conklin, 1992) que armazenam o conhecimento para posterior utilização. Repositórios são importantes na GC, pois permitem a captura, armazenamento e recuperação do conhecimento (Abecker e Decker, 1999). Em comunidades de software livre parte do conteúdo dos repositórios são os produtos desenvolvidos (em forma de código fonte ou outros documentos de software), além dos registros das interações.
- Mecanismos de recomendação: os mecanismos de recomendação funcionam como divulgação de um determinado tema, onde artefatos compartilhados podem ser apontados para um participante de acordo com seu interesse específico. Recomendações podem ser feitas pelos participantes ou de forma automática pelo sistema. A maioria das comunidades permite a recomendação de notícias. Algumas inserem o mecanismo de reputação do

participante, o qual é a forma de identificar a contribuição do participante de forma positiva ou negativa na comunidade.

- **Artefatos de conteúdo:** os artefatos de conteúdo se referem a todos os itens que contribuem para a geração e disseminação de conhecimento na comunidade, ou seja, notícias, wiki, artigos, tutoriais, dicas, configurações, scripts, pergunta sobre tecnologia, documentação de um projeto, os fóruns dos projetos, blog, as bases de conhecimento com informações sobre projetos e o próprio código fonte.
- **Mecanismos de divulgação de tecnologia livre:** os mecanismos de divulgação de tecnologia livre são a forma de divulgar a tecnologia e também divulgar a comunidade e foram identificados: notícias, artigos, informações sobre os projetos de desenvolvimento.

3.1.2. Resultados da análise das comunidades

A Tabela 1 apresenta os resultados da análise de cada um dos itens da análise das comunidades escolhidas. As comunidades virtuais de software livre possuem seus objetivos e um público-alvo bem definidos, o que garante muitas vezes que os seus participantes atuem ativamente na comunidade. Nas comunidades virtuais de software livre, o conhecimento é trocado dinamicamente por meio de mensagens síncronas ou fóruns de discussão, por exemplo, quando um participante envia uma mensagem perguntando, por exemplo, como instalar um software, e outro responde, descrevendo o passo-a-passo desta instalação. Esta prática pode apoiar o participante no momento de absorver o conhecimento, porém em outros momentos os outros participantes com dúvidas semelhantes têm dificuldade de encontrar o resultado desta interação, criando uma sobrecarga para a comunidade.

Tabela 1. Análise das Comunidades Virtuais de Software Livre

ANÁLISE	UNDER-LINUX	VIVAOLINUX	SOURCE FORGE.NET	FRESH MEAT.NET	TIGRIS	GOOGLE CODE	KODERS
Objetivo	Compartilhamento de conhecimento e apoio a projetos de software livre.	Desenvolvimento de aplicativos Linux	Desenvolvimento de software livre	Desenvolvimento de software livre	Desenvolvimento de software livre	Disponibilizar softwares desenvolvidos pelo Google de forma livre	Desenvolvimento de software livre.
Mecanismo de apoio à interação e discussão	Lista de discussão, Fóruns, Blogs	Lista de discussão, Fóruns, chats, enquetes	Lista de discussão, Fóruns, Blog	Lista de discussão, Chat, Forum	Lista de discussão, Fórum	Lista de discussão, Fórum	Lista de discussão, Fórum, Blog
Modelo de estruturação das interações	Questão, Respostas e Citações referentes a alguma resposta com uma nova resposta.	Questão e Respostas	Questão, Respostas e Citações	Questão e Respostas, porém a visualização depende do seu perfil na comunidade	Questões e Respostas	Questões e Respostas (ao autor e ao participante do fórum)	Questões e Respostas
Mecanismos de busca de informação	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves
Repositório estruturado	Sim, repositório de código fonte e de documentos	Sim, repositório de documentos com categorização	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos
Mecanismos de recomendação	Sim de notícias com comentários e possui o conceito de reputação	Sim de artigos, conceitos (notas) e comentários	Não possui, somente possui o conceito de reputação	Sim de artigos com comentários.	Não	Não	Não mas tem um sistema de feedback sobre o software

ANÁLISE	UNDER-LINUX	VIVAOLINUX	SOURCE FORGE.NET	FRESH MEAT.NET	TIGRIS	GOOGLE CODE	KODERS
Artefatos de conteúdo	Notícias com inserção comentários, Wiki, Artigos, Tutoriais, Dicas	Dicas, Configurações, Scripts, Perguntas com respostas sobre tecnologia	Notícias com comentários, código fonte, Wiki, Documentação do projeto	Notícias, comentários, código fonte, estatísticas	Fórum dos projetos, documentos	Blog, Wiki do projeto, notícias, vídeos, congressos, Google Code University (base de conhecimento)	desenvolvido. Base de conhecimento com informações sobre projetos de desenvolvimento.
Mecanismo de divulgação de tecnologia livre	Notícias	Artigos	Informações sobre os projetos de desenvolvimento	Notícias	Notícias	Notícias	Notícias

A comunicação é um ponto importante e acontece em grande parte por meio das listas de discussão que divulgam os eventos, as notícias e as dicas de tecnologia. Este mecanismo é essencial em todas as comunidades, para que os participantes estejam cientes do que acontece no cenário de software livre e na divulgação do conhecimento. As interações são feitas na maioria das vezes na forma de fóruns e chat. No entanto, observa-se que as mensagens não são estruturadas e existe uma classificação por questão. A falta de estruturação nas mensagens e de classificação organizada das informações disponibilizadas muitas vezes prejudica a identificação de informações procuradas por um participante. As comunidades virtuais de software livre que se destacam na análise são: VivaoLinux e Under-Linux, pois se organizam de forma a facilitar a interação, separando por assunto, a notícia a ser comentada ou o fórum discutido.

O mecanismo de busca de informação disponível nas comunidades pode não estar adequado às necessidades do participante, pois nem sempre o resultado obtido com a pesquisa por meio da palavra-chave corresponde à busca ou pode trazer diversas informações não relevantes. Não foram identificados mecanismos de busca mais sofisticados nas comunidades, nem uso de taxonomias ou ontologias para categorizar conhecimento. Os repositórios nem sempre estão estruturados de forma clara nas comunidades, como por exemplo, na Under-Linux, a organização do repositório é feita por assunto e por mês/ano. A maioria das comunidades analisadas possui repositórios, e não só focam em projetos de desenvolvimento, como na divulgação de ferramentas e tecnologia livre. Nessas comunidades, a forma como o conhecimento é armazenado e recuperado é estruturada, o que demonstra que a necessidade da existência de um repositório.

O estudo exploratório identificou que os ambientes que apóiam as comunidades precisam ser mais bem estruturados de forma a facilitar o armazenamento e a busca do conhecimento. Nas comunidades virtuais de software livre existe conhecimento de forma abundante e desestruturada e, a colaboração acontece de forma explícita entre os participantes. O ciclo do conhecimento (transformação de explícito em tácito e vice-versa) fica evidente, porém é necessário um processo de estruturação do conhecimento para que a GC possa ser implementada e seus benefícios alcançados.

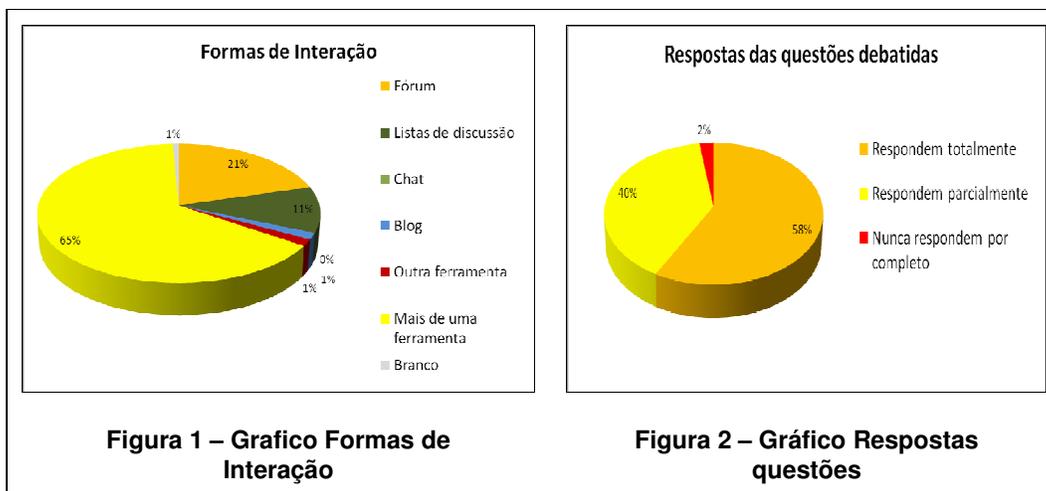
3.2. Segunda Parte: Levantamento através de Questionário

Um questionário desenvolvido em Plone foi disponibilizado no site do Projeto COLMÉIA (www.uniriotec.br/padct), o qual é um projeto em parceria entre o PESC-COPPE/UFRJ e o PPGI-CCET/UNIRIO cujo objetivo está em buscar caminhos para

ampliar as vantagens e a cultura de trabalho preconizado pelos processos de desenvolvimento de software livre através da compreensão de sua colaboração intrínseca, da proposta de ferramental de apoio à sua realização eficaz. O link foi enviado para diversas comunidades virtuais de software livre brasileiras, por meio de notícias e lista de e-mails, com o objetivo obter mais insumos para o estudo exploratório. Cento e setenta e sete (177) pessoas responderam o questionário, porém foram descartadas quarenta e seis (46) respostas devido ao fato de não terem identificado a comunidade de que faziam parte.

Os participantes fazem parte das comunidades: VivaOLinux, Ubuntu, Debian, BR-Linux, Dicas-L, BROffice, Fedora, Gnome, LimeSurvey, PSL-Brasil, Openswan-BR, Projeto ACBr, Linux, GUS-BR, ArchLinux, Drupal-br, GNU, Gentoo, ASOLBR, Crux Linux, MandrivaBrasil.org, GeoServer, Inkscape Brasil, Linux Acessível, LDP-BR, GTK, JUG, Phyton, Plone, Zope, Netbeans.org, Mysql, PHP. Com destaque principal para a VivaOLinux, Ubuntu, Debian, Dicas-l, Fedora.

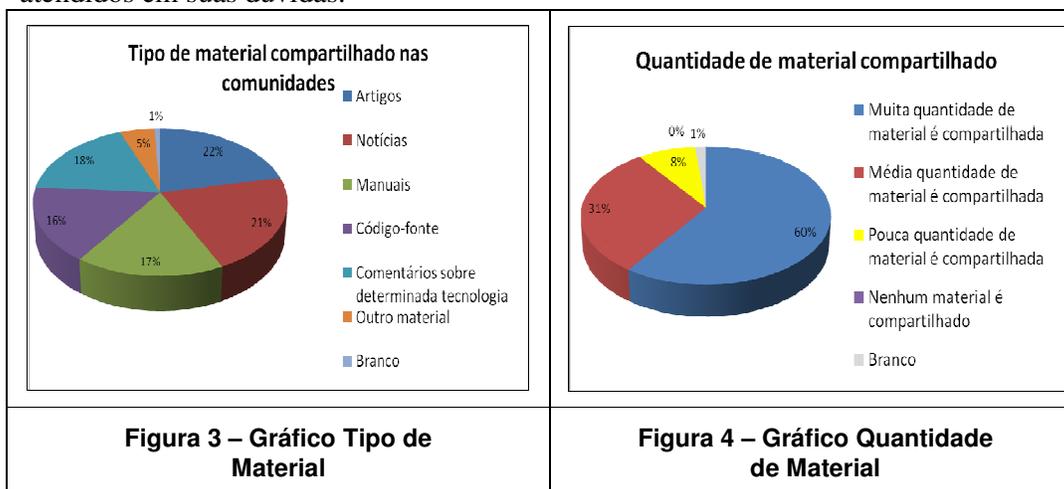
As perguntas do questionário focaram a identificação das formas de interação entre os participantes, o tipo de material que se costuma compartilhar, a quantidade de material, onde é feito o armazenamento, se as mensagens trocadas respondem as questões discutidas, se é fácil encontrar material, se é possível indicar um material e confiar nas informações disponibilizadas na comunidade. A expectativa do questionário era entender e mensurar como seria a visão dos participantes das comunidades, assim como confrontar com as conclusões da primeira parte deste estudo. As Figuras 1 a 6 apresentam parte dos resultados obtidos com a aplicação do questionário.



As formas de interação (Figura 1) apontam que não só os fóruns e as listas de discussão são utilizados, mas que as interações podem ser realizadas através de mais de uma ferramenta, como wiki, MSN e e-mails. O percentual em relação à utilizar mais de uma ferramenta de interação se destaca com 65 %, o que evidencia que as interações envolvem diversas ferramentas. Já os chats são menos utilizados que os fóruns e as listas de discussão, mostrando as interações em tempo real são menos frequentes.

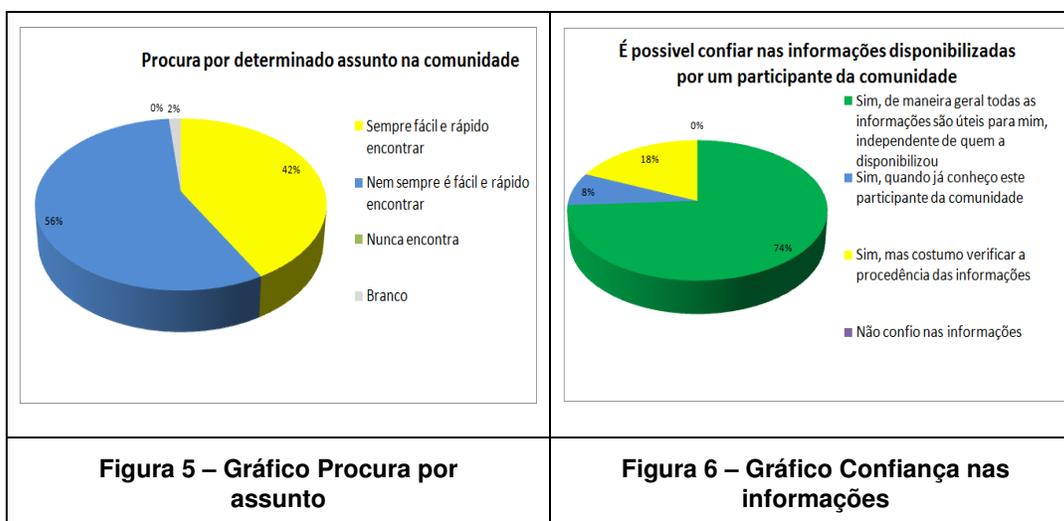
O gráfico sobre as respostas das questões debatidas durante as interações nas comunidades (Figura 2) mostra que na maior parte das vezes, quando uma questão é colocada, ela é respondida totalmente, ou seja, existe o compartilhamento do

conhecimento entre os participantes. O percentual de questões não respondidas é pequeno, portanto pode-se concluir que os participantes se sentem de maneira geral atendidos em suas dúvidas.



Os gráficos em relação ao material compartilhado (Figuras 3 e 4) identificam que os artigos são o tipo de material mais compartilhado e que os participantes afirmam a existência de muita quantidade de material compartilhado. Os participantes identificam ter muito material devido as atualizações diárias individuais, como por exemplo na comunidade VivaOLinux possuir em torno de 3065 artigos sobre Linux.

Os artigos e as notícias são os tipos de materiais mais compartilhados nas comunidades e destacam-se até em relação ao próprio código fonte. O código fonte é considerado um produto final dessas comunidades, pois estas são focadas em desenvolvimento, mas em termos de compartilhamento, pela pesquisa realizada, o código fonte não é tão compartilhado quanto os artigos, as notícias e até os manuais.



Os gráficos referentes à procura por determinado assunto (Figura 5) apontam que a existência de uma dificuldade para encontrar o que se deseja, evidencia que a busca é um problema nesse tipo de comunidade. Visto que existe muito material compartilhado, é necessária uma busca eficiente para apoiar os participantes das comunidades a

A arquitetura de ambiente que implementa tais conceitos deve contemplar as seguintes funcionalidades: captura pelos mecanismos de interação entre os participantes relacionando-as aos produtos gerados pela comunidade; categorização das interações, modelo de argumentação; busca de conhecimento baseada nos modelos de categorização; mecanismos de reputação para associar explicitamente conteúdo ao participante provedor; mecanismos de recomendação automática baseados em contexto das interações (Figura 8).

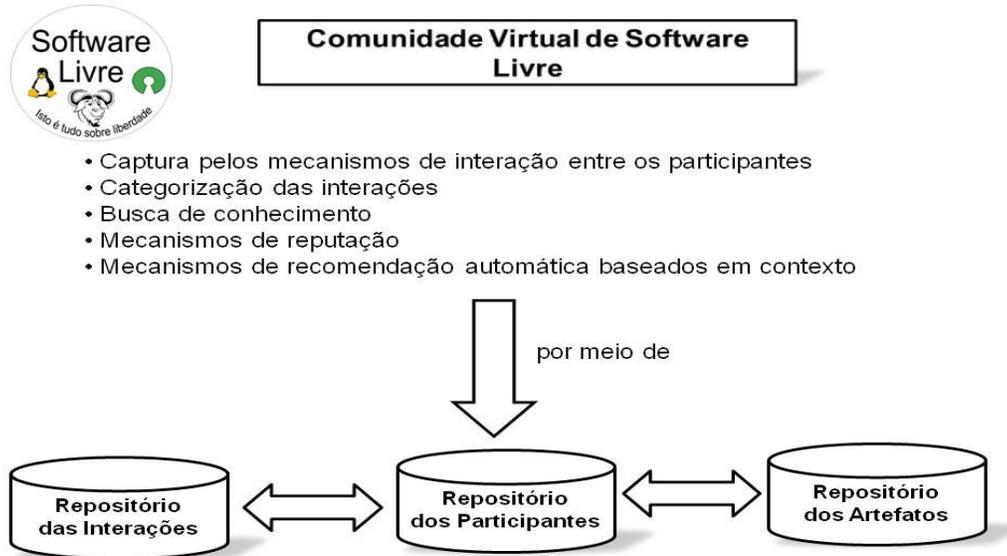


Figura 8 – Arquitetura do Ambiente para Comunidades de Software Livre com suporte à GC

5. Conclusões e Trabalhos Futuros

As comunidades virtuais de software livre são formadas por participantes dispersos geograficamente, com objetivos distintos e com as premissas de liberdade e colaboração. Este artigo analisa as comunidades virtuais de software livre, identificando seus objetivos e suas necessidades voltadas para a Gestão do Conhecimento.

A sistematização da Gestão do Conhecimento é uma demanda evidente nas comunidades. Através do estudo realizado observando critérios voltados para o ciclo da GC (captura, armazenamento e recuperação de conhecimento), concluímos que, por mais que o conhecimento seja divulgado por meio de mensagens de e-mail, dicas ou notícias, sua recuperação ainda não é tão simples ou objetiva, e o potencial do conteúdo das interações não é explorado. Desta forma, é proposto um modelo para a Gestão do Conhecimento.

As funcionalidades apresentadas no modelo estão sendo implementadas e serão expostas como serviços, de forma que possam ser incorporadas aos ambientes existentes com baixo custo. Estudos de caso serão realizados para avaliar a proposta.

Referências

Free Software Foundation, <http://www.fsf.org>, Junho de 2008.

- GNU, www.gnu.org, Junho de 2008.
- Abecker, A., Decker, S., (1999) “Organizational Memory: Knowledge Acquisition”, Integration and Retrieval Issues. In: Proc. XPS-99. Springer LNAI 1570, p.113-124.
- Conklin, E. J., (1992) “Capturing Organizational Memory”, Proceedings of GroupWare '92, D. Coleman (Ed.), Morgan Kaufmann, San Mateo, CA, 133-137.
- Conklin, J., (1998). “The Ibis Manual – Short Course in IBIS methodology”. Touchstone Whitepapers.
- Nonaka, I., Takeuchi, H. (1997) “Criação do Conhecimento na Empresa”. Rio de Janeiro, Campus, 1997.
- Raymond, E.S. (1998) “A Catedral e o Bazar”, <http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html>, Junho de 2008.
- Reis C.R. (2003) “Caracterização de um Processo de Software para Projetos de Software Livre”, Tese de M.Sc., USP, São Carlos, SP, Brasil.
- Rudzki, M.P. e Jonson F. (2003) “Identifying and Analyzing Knowledge Management Aspects of Practices in Open Source Software Development”, Tese de M.Sc., School of Engineering at Blekinge Institute of Technology, Ronneby, Suécia.
- Sowe, S.K, Stamelos, I., Angelis, L. (2008) “Understanding knowledge sharing activities in free/open source software projects: An empirical study”. The Journal of Systems and Software, volume 81, número 3, p.431-446.
- Taurion, C. (2004), Software Livre: potencialidades e modelos de negócios, Editora Brasport, 1^a Edição, Rio de Janeiro.
- Terra, J.C. (2007) “Gestão do Conhecimento: Tecnologia e Aprendizado Coletivo no Contexto da Educação Corporativa”, <http://www.terraforum.com.br/>, Junho 2008.
- von Krogh, G., Spaeth, S., Haefliger, S. (2005) “Knowledge reuse in open source software: an exploratory study of 15 open source projects”. In Proceedings of the 38th Hawaii International Conference on System Sciences, track 07, volume 07, p. 198b.
- Wenger, E., McDermott, R., Snyder, W. M. (2002) “Cultivating Communities of Practice: a guide to managing knowledge”. Harvard Business Scholl Press, Boston, MA.