Padrões de Socialização de Novatos em Projetos de Software Livre

Ana Claudia Maciel[1], **Igor Steinmacher**[2], **Marco Aurélio Graciotto Silva**[2]

[1]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)

[2]Departmento Acadêmico de Computação
Universidade Tecnológica Federal do Paraná (UTFPR)

anamaciel@usp.br, {igorfs,magsilva}@utfpr.edu.br

Abstract. *Numerous open source software (OSS) projects are based on volunteers collaboration and require a continuous influx of newcomers for their continuity. However, newcomers face barriers when starting their contributions. Using a method based upon mining of software repositories and social network analysis, we aim at detecting socialization patterns for newcomers in OSS projects. As research subject, we use the Apache project Hadoop Common. We analysed seven years of messages and issues history. The results point that most newcomers stays for few months in the project, and the few persistent newcomers employ just one interaction method and interact mostly with experienced developers. Due to the small account of fruitful interactions, we could not detect further socialization patterns.*

Resumo. *Muitas comunidades que mantêm projetos de software livre demandam a colaboração de voluntários e necessitam da entrada contínua de novatos. No entanto, os novatos enfrentam obstáculos ao iniciar sua interação em um projeto. Este trabalho utiliza um método dividido em etapas e baseado em mineração de repositórios de software e análise de redes sociais com o objetivo de encontrar padrões de socialização de novatos em projetos de software livre. O projeto analisado foi o Hadoop Common. Foram analisados sete anos do histórico de mensagens e tarefas. Observou-se que a maioria dos novatos permanecem pouco tempo no projeto, aqueles que permanecem utilizam apenas um meio de interação e comunicam-se basicamente com veteranos. Devido à pequena quantidade de interações não foi possível a identificação de outros padrões.*

1. Introduçao

Projetos de Software Livre são conduzidos principalmente por voluntários: desenvolvedores que participam livremente dos projetos que consideram atraentes (MADEY et al., 2002), o que demanda a constante entrada e retenção de novos contribuintes (PARK; JENSEN, 2009). Dessa forma, o sucesso de um projeto de software livre é improvável sem que haja uma comunidade que forneça uma plataforma para que desenvolvedores e usuários colaborem uns com os outros (YE; KISHIDA, 2003).

Entretanto, os primeiros passos desses novatos em projetos de software livre podem oferecer diversos obstáculos. Dagenais et al. (2010) comparam novatos em projetos de software a exploradores que precisam se orientar em um ambiente hostil. De fato, os novatos geralmente precisam aprender aspectos sociais e técnicos sozinhos, explorando as informações existentes em listas de e-mails, repositórios de código fonte e gerenciadores de tarefas (SCACCHI, 2002). Não é fácil acessar essas informações devido ao grande volume, à falta de ferramentas para navegar nos repositórios e à dificuldade de fazer as conexões entre os itens relacionados em fontes diferentes (CUBRANIC et al., 2005).

Mesmo em meio a essas adversidades, muitos projetos de software livre são bem sucedidos. De fato, os projetos de software livre oferecem uma chance de usuários e desenvolvedores, sejam eles novatos ou experientes, trabalharem para um mesmo objetivo prático em busca de resultados concretos, formando assim uma comunidade (CAMPOS, 2006). Uma forma de compreender as características da comunidade de um projeto de software livre é a sua representação como uma rede social, que consiste de um conjunto de atores e as relações definidas entre eles (BALIEIRO et al., 2007).

A partir da análise das redes sociais, é possível compreender a interação e a organização social de um grupo. A semântica do relacionamento depende da análise que se deseja conduzir nesta rede. Especificamente em Engenharia de Software, utiliza-se a análise de redes sociais para entender a colaboração entre os membros da equipe de desenvolvimento (MAGDALENO et al., 2010).

Não obstante, observa-se a carência de estudos sobre os novatos nestas redes sociais (HE et al., 2012), em especial como eles são inseridos na rede e como eles interagem com outros personagens dela. Este trabalho tem por objetivo identificar padrões de entrada e migração dos novatos baseado em análise de redes sociotécnicas de projetos de software livre, analisando-se também as alterações do relacionamento entre os desenvolvedores, tanto novatos quanto os experientes no projeto. Para alcançar este objetivo, definimos os seguintes objetivos específicos:

- Identificar padrões de entrada de novatos;
- Identificar padrões de migração de novatos em uma rede social ao longo do tempo;
- Identificar padrões de interação entre os desenvolvedores, mais especificamente dos veteranos com novatos, e entre os novatos;
- Identificar permanência dos novatos na rede social.

O projeto selecionado para este trabalho foi o Hadoop Common[1], hospedado pela Apache Software Foundation[2]. Analisamos dados da lista de e-mails e do gerenciador de tarefas, a partir dos quais estabelecemos redes sociotécnicas e, com o auxílio de técnicas de análise de redes sociais, identificamos padrões de interação dos novatos na comunidade do projeto de software livre analisado.

2. Método de Pesquisa

Para conduzir o estudo, o primeiro passo consistiu na escolha do projeto de software livre a ser analisado. Posteriormente recuperamos os dados do projeto escolhido para que, na

[1]http://hadoop.apache.org
[2]http://www.apache.org/

próxima etapa, fosse realizada a obtenção dos dados do repositório de software. Os dados foram utilizados para representar as redes sociais baseadas nas interações dos membros no projeto escolhido anteriormente. Por fim, conduzimos a análise da rede social. Esses passos são apresentados nas subseções a seguir.

2.1. Extração dos dados

Cada projeto de software livre, na forma de uma comunidade de desenvolvimento, possui características intrínsecas. Tais particularidades influenciam nas colaborações entre os desenvolvedores e, portanto, nos padrões detectáveis entre essas interações.

Embora seja inviável analisar todos os projetos ou uma amostra significativa, de modo a detectar um conjunto de padrões comuns a projetos de software livre, é possível selecionar projetos que podem fornecer resultados interessantes para a identificação de padrões de novatos. Por exemplo, um projeto com uma comunidade saudável, com objetivos claramente definidos e com uma infraestrutura e organizações adequados provavelmente seria um bom objeto de estudo.

Uma forma indireta de medir a qualidade de um projeto é pelo seu grau de atividades (KOLASSA et al.,). Segundo Daffara (2007), pode-se dizer que um projeto está ativo quando o número de commits nos últimos 12 meses é de pelo menos 60% do número de commits nos 12 meses antes disso. Uma forma de verificar tal característica é por sites que analisam o grau de atividade de projeto de software livre, tal como `http://www.ohloh.net/`.

Após a escolha do projeto, o próximo passo do método foi a recuperação dos dados. Para a análise foram utilizados dados de gerenciadores de tarefas como o Jira ou Bugzilla, ambientes em que os colaboradores relatam erros e solicitam novas funcionalidades (no restante deste texto será utilizado o termo tarefas para representar ambos). Os membros podem comentar sobre as tarefas, dando sugestões e soluções para os problemas abordados.

Outra fonte importante de interações em projetos de software são as listas de e-mail. Arquivos, contendo todas as mensagens das discussões são geralmente disponibilizados pelos gerenciadores dessas listas e sites de indexação para tais discussões. Analisando-se os dados das listas de e-mails, é possível saber quem são os desenvolvedores envolvidos e as mensagens compartilhadas.

Para a coleta dos dados do gerenciador de tarefas, foi utilizada uma ferramenta que extrai os dados relativos às tarefas e os armazena em um banco de dados relacional. Para cada tarefa relatada os dados extraídos serão: descrição; usuário relator; responsável; data de criação; data de fechamento; prioridade; status atual; e comentários (com autor, data e mensagem).

Para extrair os dados de e-mails, primeiramente foram obtidos os arquivos que contêm todos os e-mails, incluindo cabeçalho e corpo da mensagem. As informações das mensagens contidas nos arquivos foram coletadas, analisando-se os cabeçalhos para adquirir informações do conteúdo da mensagem, assunto, identificador da mensagem, remetente e identificador da cadeia de mensagens (`In-reply-to`), que identifica a árvore de discussão (*thread*) a qual a mensagem pertence. Essas árvores foram reconstruídas verificando o campo `In-reply-to` do cabeçalho bem como o assunto do e-mail (exa-

minando os prefixos “Re:”, “Fwd:”) e o campo *references* do cabeçalho, para diminuir as chances de perda de mensagens relativas a uma discussão. Os e-mails obtidos foram armazenados em um banco de dados local contendo os detalhes das mensagens extraídas.

Para a análise das mensagens de e-mail, foram desconsideradas as mensagens enviadas automaticamente na criação, comentário ou mudança de estado de uma tarefa. Por exemplo, no Jira, utilizado no projeto Apache, tais mensagens são identificadas pelo endereço do remetente Jira@apache.org ou pelo prefixo “[Jira]” no assunto da mensagem.

2.2. Análise dos dados

Os dados obtidos foram utilizados para criar redes sociais baseadas nas interações dos membros em cada um dos meios analisados. Os dados das listas de discussão e daqueles provenientes do gerenciador de tarefas foram mesclados em uma única rede por meio dos autores das mensagens enviadas.

Para analisar a migração de determinado membro do projeto foi necessário solucionar o problema de identificação ambígua existente entre a lista de e-mails e Jira, considerando que no projeto o membro possui um identificador e na lista de e-mails ele possui um ou mais endereços de e-mail. Investigamos heurísticas que permitam determinar se um determinado membro do projeto encontra-se nas duas redes, verificando o nome do autor, endereço utilizado para enviar e-mails na lista de e-mails e o identificador utilizado no Jira. Caso não seja possível, uma análise manual é realizada para mesclar os dados. A união dos dados desses dois meios em uma única rede é importante para analisar a migração dos membros no projeto, inclusive a atuação nas diferentes redes.

2.2.1. Análise da rede social

Na rede social resultante foram analisadas as interações dos membros de acordo com o contexto em que a interação foi realizada. A representação dos membros dessa rede leva em consideração em qual dos meios o membro apareceu, seja de maneira isolada, seja concorrentemente em ambos os meios. Na Figura 1 temos uma possível representação de uma rede social contendo dados das interações realizadas por meio do Jira e da lista de e-mails. Os vértices representam o usuário, de acordo com o local/ferramenta que ele interage, e as arestas simbolizam as interações entre os usuários.

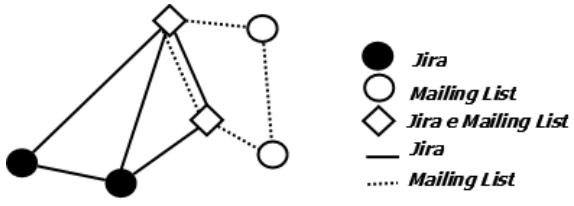


Figura 1. Representação de uma rede social com interação no Jira e lista de e-mails

Foram criadas diferentes redes sociais temporalmente, em diferentes intervalos, para analisar a migração dos membros. A proposta inicial de intervalos de criação das redes é apresentada na Figura 2. O primeiro intervalo (I1) agregará um período de 3 anos, do qual será extraída uma rede social inicial. Essa rede será considerada o ponto de partida:

os desenvolvedores que estiverem inclusos nesta rede serão considerados membros já existentes no projeto.

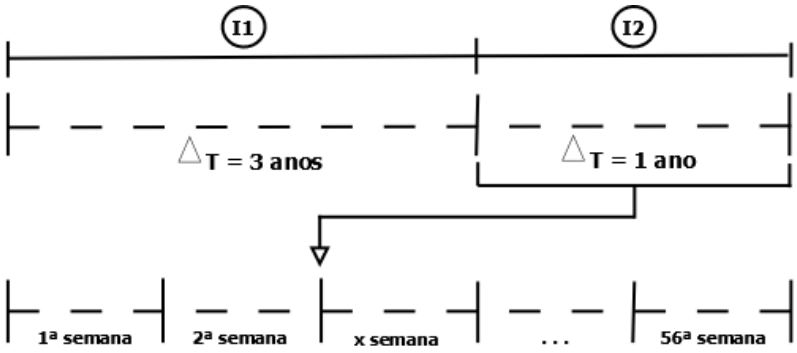


Figura 2. Linha do tempo para criação das redes sociais

O segundo grande intervalo (I2) contemplará o período de doze meses posteriores à data de criação da rede inicial. Esse intervalo será dividido em intervalos semanais a fim de conduzir a análise temporal. Os seis meses iniciais de I2 serão utilizados para identificar os novatos do projeto. Para isso, serão considerados novatos aqueles membros que aparecem nos primeiros seis meses de I2 e que não haviam aparecido em I1.

Para cada novato encontrado serão analisados os próximos 6 meses de interação a contar da data de sua primeira aparição. Para isso serão utilizadas as redes criadas semanalmente em I2. Um possível resultado a ser encontrado pode ser visualizado na Figura 3, em que o membro A começa em uma rede com poucos contatos, passa a aparecer nas duas redes com outros contatos e, por fim, apresenta-se mais central na rede, com contatos nos diferentes meios de interação. O estudo será elaborado para ser flexível quanto ao tempo, considerando que o intervalo semanal definido previamente pode não apresentar resultados satisfatórios, sendo necessário aumentar ou diminuir o período de tempo estabelecido.

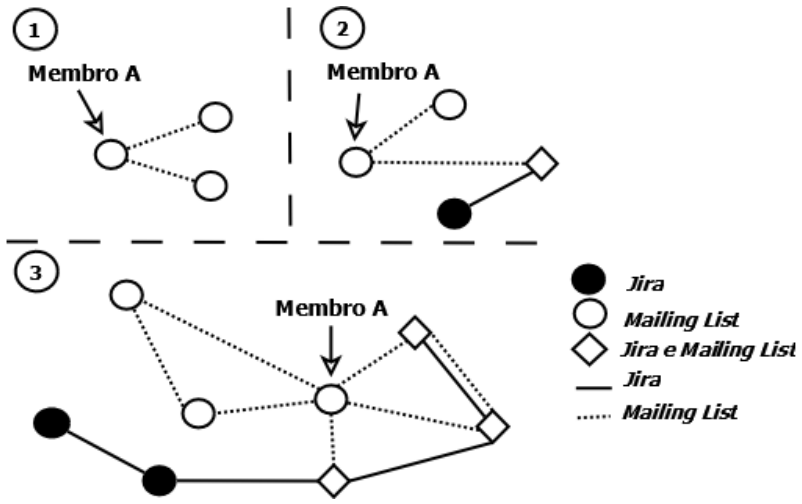


Figura 3. Possível migração temporal de um membro na rede social

Para a rede social, os membros serão classificados de acordo com o período de aparição e a participação (definida de acordo com a quantidade de mensagens enviadas), dividindo-os em três categorias:

- Membros centrais: apareceram no intervalo 1 e estão entre os 10% mais participativos;
- Novatos: não apareceu no intervalo 1 e apareceu no intervalo 2;
- Outros membros: apareceram no intervalo 1 e não estão entre os 10% mais participativos.

Quanto a centralidade de intermediação, duas medidas se destacam: *betweenness* e *closeness*. *Betweenness* é uma medida de papel central no interior de um vértice de um grafo. Os nós que estão nos caminhos mais curtos entre outros nós têm maior *betweenness* (WASSERMAN; FAUST, 1994). *Closeness* enfatiza a distância de um nó para todos os outros da rede centrando-se na distância geodésica de cada nó para todos os outros, pode ser considerada como uma medida de quanto tempo vai levar para as informações trafegarem a partir de um determinado nó para outros nós da rede (HE et al., 2012).

Por fim, serão analisados os relacionamentos dos novatos dentro do projeto a fim de verificar a existência de padrões de interação social e migração nos primeiros passos no projeto. A análise dos padrões levará em conta o meio de entrada do novato, a migração para outro meio e os tipos de interação dos novatos com outros membros. Por exemplo, com base no trabalho de He et al. (2012), as interações possíveis são: (i) entre novatos; (ii) entre novato e membro do núcleo; e (iii) entre novato e outros membros. Outros fatores poderão ser analisados baseando-se nas redes sociais obtidas, como, por exemplo, o comportamento da centralidade dos membros no decorrer do tempo. Entretanto, tais análises não são parte do escopo inicial deste trabalho.

Com esses passos, pretende-se identificar, se houver, padrões de socialização dos novatos em um projeto de software livre.

3. Resultados

O método definido no Seção 2 foi aplicado, escolhendo-se o projeto Hadoop Common. O projeto Hadoop Common foi escolhido por ser um projeto de sucesso, já consolidado, e com uma comunidade ativa e bem organizada (STEINMACHER et al., 2012). Além disso, os dados do gerenciador de tarefas e listas de e-mails estão disponíveis e podem ser coletados livremente. Nas próximas seções, são apresentados os resultados e as considerações sobre a aplicação de cada passo do método, desde a seleção do projeto até a detecção de padrões de socialização dos novatos no projeto selecionado.

3.1. Especificação e extração dos dados

A análise foi realizada com dados do gerenciador de tarefas (Jira) e da lista de e-mails do projeto. Para a coleta dos dados do gerenciador de tarefas, foram utilizados os serviços Web (REST) do Jira, que retornam arquivos no formato JSON. Foi utilizada uma ferramenta (`https://github.com/magsilva/SPA`) para utilizar tais serviços, fazer leitura dos arquivos JSON com as informações das tarefas e obter as informações detalhadas dos comentários atrelados a elas. Cada tarefa é um problema e cada comentário corresponde a uma solução.

Para extrair os dados da lista de e-mails, foram obtidos os arquivos no formato mbox, que contêm todos os e-mails, incluindo cabeçalho e corpo da mensagem. As informações

das mensagens contidas nos arquivos foram coletadas a partir do repositório do Apache, localizada em `http://mail-archives.apache.org/mod_mbox/`, com o auxílio de um script para automatizar a obtenção dos dados de cada mês.

As mensagens de e-mail foram processadas com a ferramenta Presley (TRINDADE et al., 2009) e armazenadas em um banco de dados relacional, contendo os desenvolvedores e as mensagens. As interações realizadas pela lista de e-mails foram classificadas em problemas e soluções. A primeira mensagem de uma *thread* é um problema e as mensagens restantes são classificadas como solução.

Para analisar a migração de determinado membro do projeto seria necessário solucionar o problema de identificação ambígua existente entre a lista de e-mails e Jira, considerando que no projeto o membro possui um identificador e na lista de e-mails ele possui um ou mais endereços de e-mail. Inicialmente, foi adotada a heurística de que quando tem-se a ocorrência de dois ou mais nomes de usuários iguais, eles são mesclados, podendo ser representados pelos e-mails cadastrados para estes usuários mesclados. Os e-mails da apache são considerados os e-mails principais caso exista mais de um e-mail para o usuário. No entanto, por questões de tempo e dificuldades para implementar tal heurística, optou-se por realizar a união manualmente caso fosse necessário (por exemplo, novatos com muitas e frequentes interações).

3.2. Análise dos dados

Os dados obtidos foram utilizados para criar redes sociais baseadas nas interações sociotécnicas dos membros em cada um dos meios analisados, mais precisamente a resolução de problemas (tarefas de desenvolvimento de software), tal como descrito na seção anterior.

As redes foram criadas com base em arquivos gml (Graph Modeling Language) de formato texto que suportam dados de redes. Posteriormente, a ferramenta Gephi foi utilizada para visualizar as redes contidas nos arquivos gml. O layout utilizado foi o Force Atlas 2, ele simula um sistema físico onde os nós se repelem como imãs, enquanto as bordas atraem os nós se conectam. Estas forças criam um movimento que converge para um estado de equilíbrio, buscando ajudar na interpretação dos dados (JACOMY et al., 2011).

Inicialmente, foram criados três grafos: um com os dados do gerenciador de tarefas (Figura 4), outro com dados da lista de e-mails (Figura 5) e o terceiro, com a união de ambos, representado na Figura 6.

A rede do Jira é composta por 176 nós e 244 arestas. Nem todos os novatos possuem uma ligação com outro nó. Por exemplo, vários novatos da periferia estão relacionados apenas consigo mesmo, ou seja, o próprio novato que criou a tarefa e realizou os comentários. Outro ponto a ser observado é que alguns novatos estão relacionados com um veterano, mas não estão associados à grande componente do grafo.

A rede de e-mails, apresentada na Figura 5, é composta por 699 nós e 1273 arestas. Em relação à rede do Jira, ela é mais complexa. De modo a facilitar a análise e visualização, foram retirados os nós com grau inferior a 1.

Após a união dos dados das redes do Jira e dos e-mails, sem considerar a identificação

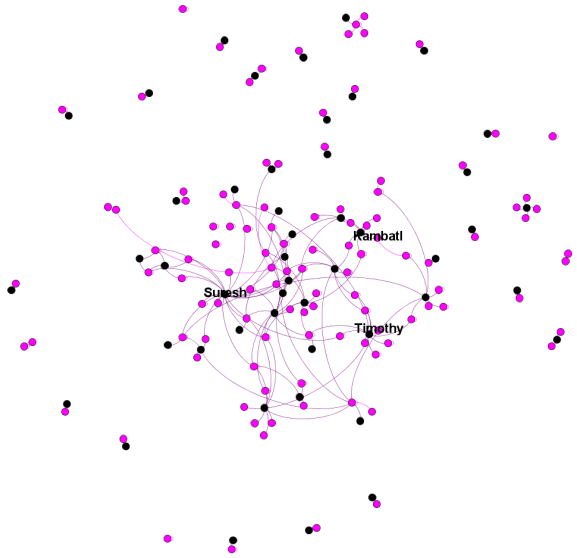


Figura 4. Rede social representando a comunicação por meio do gerenciador de tarefas. Os novatos são representados pelos nós de cor rosa (tom mais claro) e os veteranos pelos nós de cor preta.

de desenvolvedores duplicados entre as redes, obtivemos a rede apresentada na Figura 6. A rede possui 867 nós, dentre eles 635 são novatos. Foram excluídos todos os nós que tinham grau zero. As arestas que estão visíveis são aquelas que possuem novato em alguma extremidade da interação.

Observando-se a questão temporal dos dados, inicialmente planejava-se analisar os dados semana a semana. No entanto, como apresentamos nas próximas seções, a quantidade de interações não era o suficiente para essa granularidade. Dessa forma, optou-se por utilizar a periodicidade mensal para a análise temporal.

Para a identificação dos novatos, foram utilizados todos os meses anteriores à julho de 2012 (ou seja, a partir do mês 01 de 2006). Posteriormente, consideraram-se os seis meses seguintes, de julho a dezembro de 2006, para identificar e analisar os novatos. Originalmente, planejava-se identificar os novatos em um período de seis meses e analisá-los nos seis meses seguintes a esse. Entretanto, caso mantivéssemos tal estratégia, não poderíamos analisar a entrada dos novatos, que é um dos objetivos desse trabalho. Portanto, fizemos da seguinte forma: os veteranos foram os desenvolvedores que apareceram a partir de janeiro de 2006 à junho de 2012 e os novatos são os que tiveram interações no segundo semestre de 2012 e não haviam tido participações anteriores a esta data. A análise dos novatos foi feita no mesmo semestre que os identificamos.

3.3. Meio de entrada dos novatos

Analisando todos os novatos e suas interações no segundo semestre de 2012, cujos dados estão apresentados na Seção 3.3, detectamos que a maioria (81,5%) dos novatos utiliza a lista de e-mails como meio de entrada. Essa característica era esperada, dado que o conhecimento técnico necessário para a comunicação por e-mails é mais simples do que aquele requerido para a utilização do Jira.

No total, observa-se a entrada de 635 novatos no projeto no período analisado. Entre-

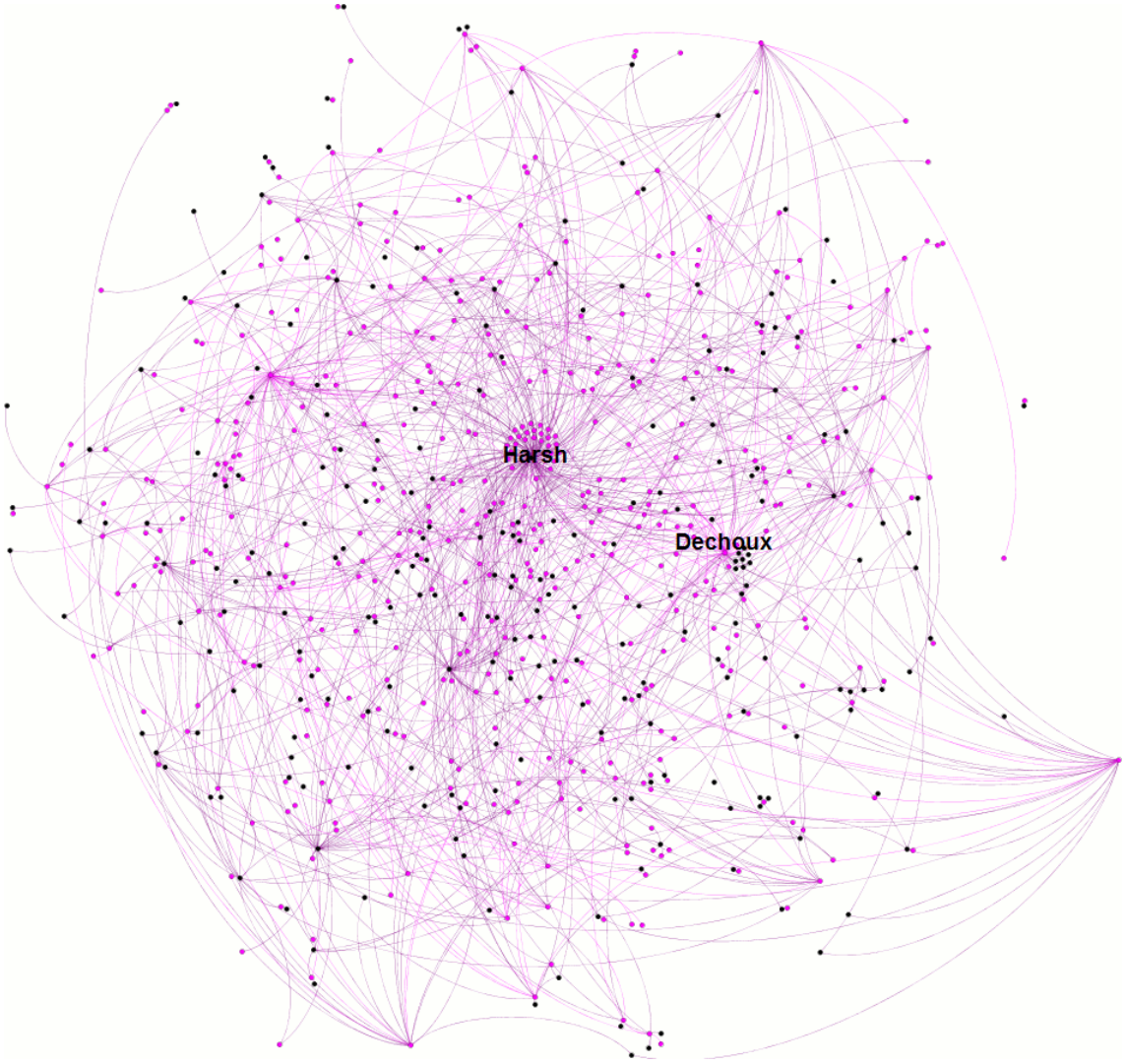


Figura 5. Rede social representando a comunicação da lista de e-mails. Os nós representados pela cor rosa (tom mais claro) são os novatos e os de cor preta são os veteranos.

tanto, resta analisar se tais números afetam a participação desses no projeto. Na Seção 3.3, apresentamos a quantidade de meses que um novato interagiu com o projeto e a porcentagem que essa quantidade representou em relação ao total de novatos daquele meio.

Em geral, os novatos possuem interações pontuais em apenas um dos meios de entrada. Por exemplo, Ashwin, novato com meio de entrada na lista de e-mails, possui alto *closeness*, porém possui uma participação pontual, não permanecendo no projeto nos próximos meses. Em outras palavras, embora não tenha muitas interações, elas aconteceram com um desenvolvedor-veterano com alto *betweenness*. Isto está de acordo com o afirmado por STEINMACHER et al. (2013): a maior parte das perguntas enviadas pelos novatos são respondidas por membros do núcleo (veteranos) e que a maior parte das discussões iniciadas pelos novatos que deixam o projeto recebem, também, respostas de veteranos.

Em relação à migração entre meio de interação, observando-se ainda a lista de e-

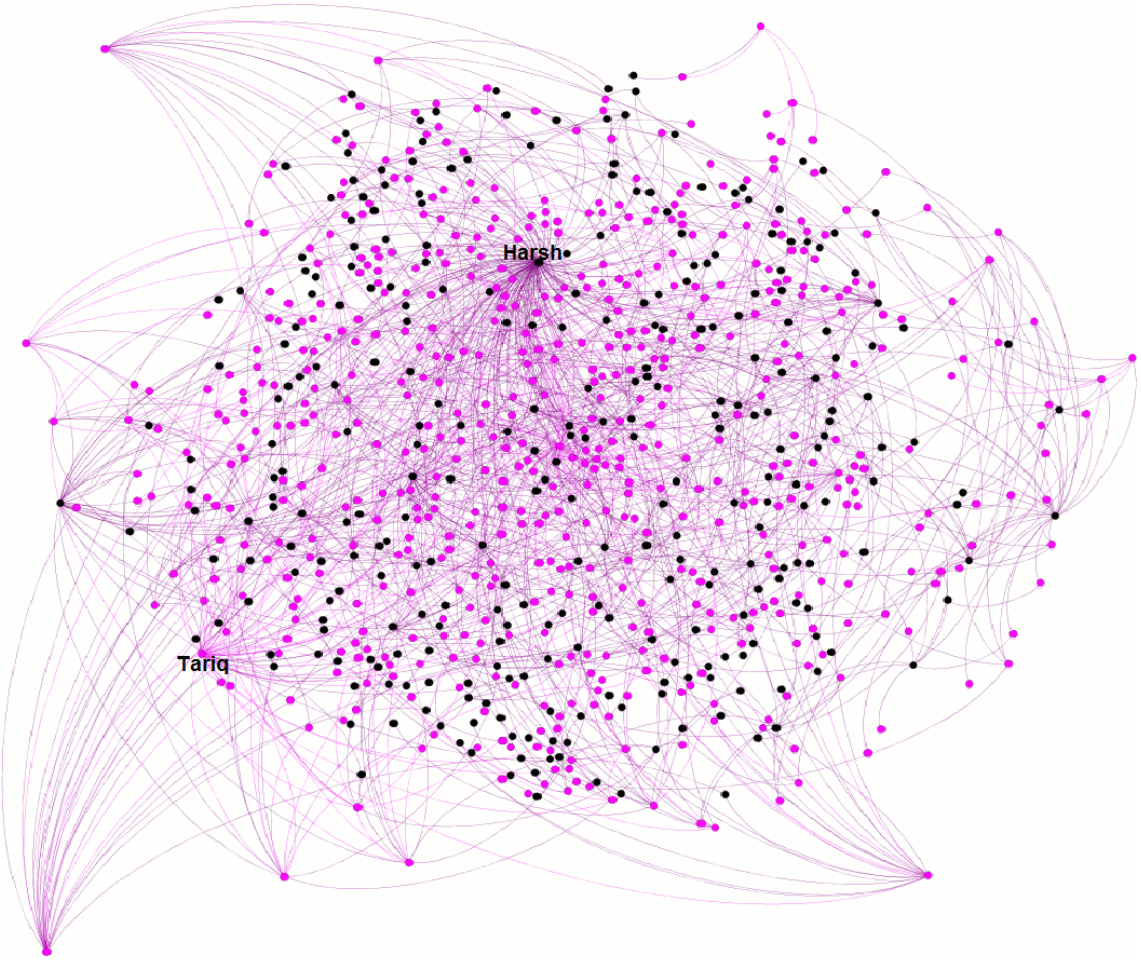


Figura 6. Rede social representando a união da lista de e-mails e gerenciador de tarefas. Os nós representados pela cor rosa (tom mais claro) são os novatos e os de cor preta são os veteranos.

mails, Tariq permaneceu no projeto com interações em cinco dos seis dos meses analisados, obteve 19 na rede, mas não migrou para o Jira. Tal padrão também pode ser observado para quem iniciou no Jira. Por exemplo, Parker, novato do Jira, não migrou para a lista de e-mails no decorrer do tempo analisado e abriu tarefas duas tarefas para correção de bugs e duas tarefas de melhoria.

Alguns novatos, diferente do esperado para os mesmos, só interagiram no Jira e permaneceram com suas participações em todos os meses da análise, como o Kambatla.

Quinze novatos migraram de um meio para outro. Por exemplo, Beech, começou sua interação na lista de e-mails e passou a ter pequenas participações no Jira. Entretanto, tais novatos, que migraram da lista de e-mails, não se mantiveram no projeto por muito tempo.

Quatro novatos começaram pelo Jira e depois migraram para a lista de e-mails, Ozawa é um exemplo que começou abrindo uma tarefa de melhoria e depois iniciou sua participação na lista de e-mails. Diferente dos padrões esperados, foram encontrados novatos que começam pelo Jira e depois migram para a lista de e-mails, outros nem aparecem na lista de e-mails e só interagem no Jira. Tais padrões não eram esperados

Tabela 1. Quantidade de novatos por meio de entrada no semestre de 2/2012

	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
E-mail	48 (1,24%)	183 (4,73%)	84 (2,17%)	78 (2,01%)	71 (1,83%)	53 (1,37%)
Jira	21 (1,56%)	26 (1,93%)	25 (1,86%)	22 (1,63%)	16 (1,19%)	8 (0,59%)
Total	69	209	109	100	87	61

Tabela 2. Quantidade de novatos que interagiram entre 1 e 6 meses no semestre 2/2012

	1	2	3	4	5	6
E-mail	373 (9,65%)	90 (2,32%)	28 (0,72%)	11 (0,28%)	4 (0,103%)	1 (0,025%)
Jira	95 (7,07%)	16 (1,19%)	4 (0,30%)	1 (0,075%)	1 (0,075%)	1 (0,075%)

devido ao volume de mensagens e desenvolvedores da lista de e-mail ser maior se comparado ao Jira. Somado a esta razão existe o fato de o e-mail ser um meio de entrada mais acessível tecnicamente ao novato, inclusive com recomendações de gerentes de projetos de software livre para entrar e discutir por e-mail antes de abrir uma tarefa no Jira.

3.4. Padrões de socialização de novatos

A análise da rede foi feita de acordo com os valores de *closeness* e *betweenness*. No contexto de engenharia de software, os desenvolvedores novatos com alto *betweenness* são aqueles que possuem os caminhos mais curtos entre os demais desenvolvedores e os novatos com alto *closeness* são os que têm relação com os desenvolvedores mais influentes na rede.

Os dez desenvolvedores novatos de maior *closeness* das redes do segundo semestre de 2012 da lista de e-mails e do Jira foram separados e analisados. Com isto, buscou-se a identificação do comportamento desses novatos que são considerados bem sucedidos pelo valor de *closeness*. A partir desses dados, observamos que os novatos que possuem alto *closeness* não permanecem no projeto: suas participações são pontuais.

A partir do grafo apresentado na Figura 6, percebemos que existe uma grande quantidade de novatos com comportamentos diferentes dentro da rede. Identificou-se novatos que encontram-se ao centro do grafo, com alto *closeness* e *betweenness*. No entanto, também foram identificados novatos com poucas interações e consequentemente baixo *closeness* e *betweenness*, aparentemente não obtendo sucesso na comunidade.

Considere o grafo da Figura 7 (lista de e-mails), no qual são apresentados somente os desenvolvedores com grau maior que dez. Observa-se que existem alguns novatos com uma centralidade significativa (alto *closeness* e *betweenness*), tal como aquele destacado com um nó com maior diâmetro. A situação desses novatos demonstra um padrão interessante, relacionando-se com desenvolvedores experientes. O nó representado com o diâmetro maior na Figura 7 é o desenvolvedor Dechoux, da lista de e-mails, com alto *closeness* e *betweenness*: ele interagiu com novatos e veteranos de alto *betweenness*.

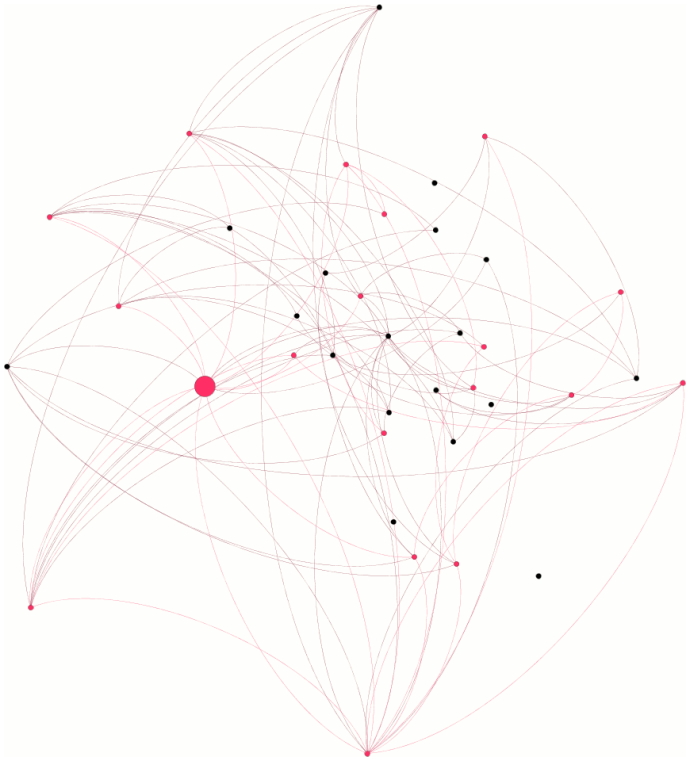


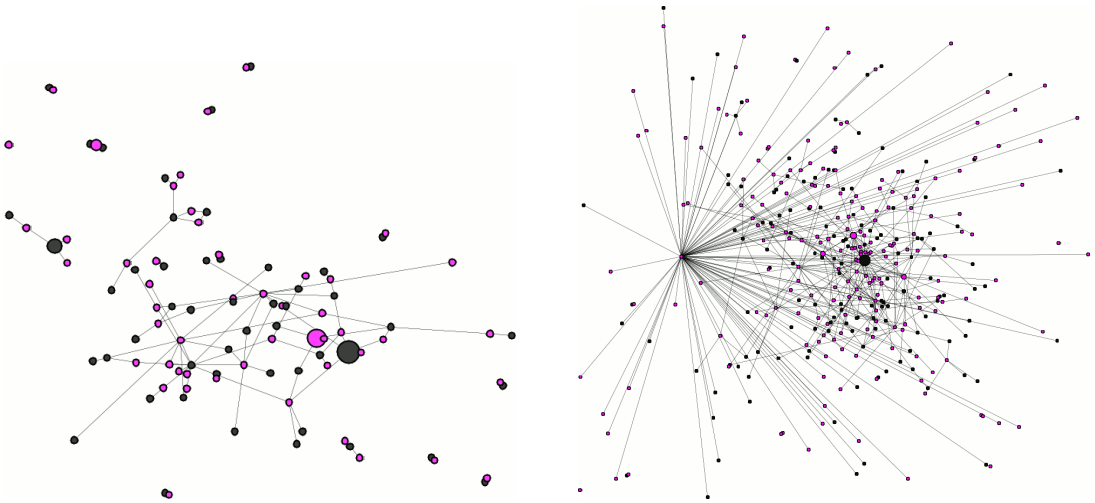
Figura 7. Rede social com nós de grau maior que dez.

Em oposição a esse cenário positivo, temos a situação dos novatos que não foram bem sucedidos e deixaram de interagir no projeto. Por exemplo, para a ilustração da rede apresentada na Figura 6, foram excluídos os desenvolvedores que tiveram grau zero no período de análise. Além disso, muitos nós possuem grau 1. Tais novatos, com graus baixos, desistiram de dar continuidade à participação no projeto, talvez por não conseguirem uma boa comunicação. Sendo assim, esses casos também são importantes para análise e identificação dos padrões.

Para uma análise mais detalhada, foram criadas redes de cada mês do segundo semestre de 2012 ao primeiro semestre de 2013. Porém, a cada nova rede gerada mês a mês, os nós mudam de posição, dificultando assim, a análise de padrões, um exemplo das redes mês a mês pode ser observado na Figura 8. Diante deste problema, foram criadas redes semestrais, em que uma representou o segundo semestre de 2012 e a outra o primeiro semestre de 2013, representadas nas Figuras 9 e 10, onde os novatos estão representados pela cor clara, enquanto os veteranos são os de cor cinza. A diferença de diâmetro é determinada pela centralidade, quanto maior a centralidade, maior o diâmetro.

Diante dos grafos apresentados nas Figuras 9 e 10, foram observados alguns novatos com alto *closeness* e *betweenness*, conforme pode ser visto pelo tamanho dos nós (quanto maior o nó maior a centralidade). A análise destes novatos foi feita de forma visual, com base nos grafos semestrais. Em destaque, na Figura 9, temos o novato Tariq, também presente na Figura 10 em conjunto de Embree, ambos da lista de e-mails.

Depois de selecionar os novatos com alto *closeness* e *betweenness*, apresentados na Tabela 3, foi feita uma análise individual para eles em busca de identificar padrões. Porém,



(a) Comunicações do mês de julho/2012.

(b) Comunicações do mês de agosto/2012.

Figura 8. Rede representando a comunicação dos meses de julho e agosto de 2012.

a maioria dos novatos selecionados não permaneceram, tendo apenas interações pontuais e deixaram o projeto. Por exemplo, o novato de maior centralidade é Tariq (representado com maior diâmetro na rede). Ele interagiu na lista de e-mails por 5 meses, possui *betweenness* de valor 1991,217 e *closeness* de valor 1,647. Todos os 19 desenvolvedores com quem se relacionou são veteranos.

Tabela 3. Dados dos novatos de alto *closeness* e *betweenness*

Novato	Grau	*Closeness*	*Betweenness*
Spaggiari	5	3,477	130
Kumar	3	2,778	22,667
Dechoux	22	2,556	338,7
Embree	5	2,5	210,5
Sasha	8	2,419	214,333
Sadak	20	1,774	502,467
Tariq	19	1,647	1991,217
Verwilst	3	1	216,5

Observa-se também que a comunicação no gerenciador de tarefas é menos intensa do que na lista de e-mail, levando-nos a um resultado de mais novatos com alto *closeness* e *betweenness* na lista de e-mails que no Jira.

4. Conclusões

A grande base de desenvolvedores contribuindo voluntariamente é um dos mais importantes fatores de sucesso dos projetos de software livre. Qualquer modificação ou melhoria feitas em um projeto, redefine o papel dos membros que contribuem, alterando assim, a dinâmica social da comunidade (YE; KISHIDA, 2003).

Este trabalho teve como objetivo identificar padrões de socialização dos novatos de modo a auxiliar tais projetos a compreender esta dinâmica e a melhorar os mecanismos

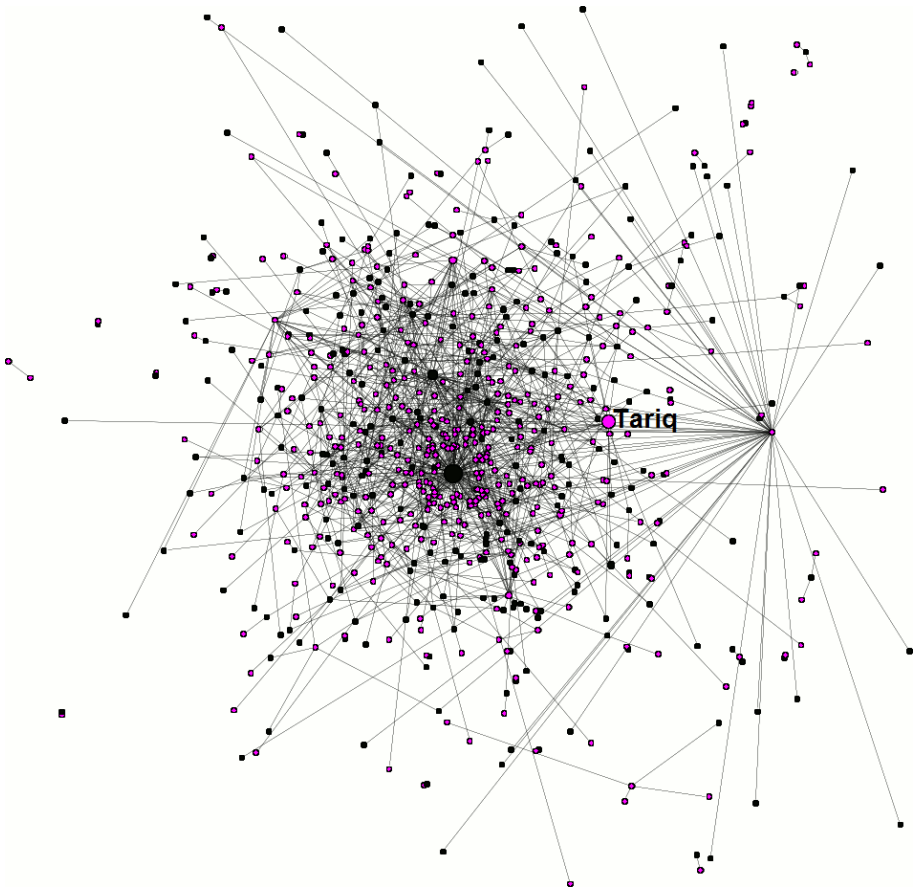


Figura 9. Rede representando a comunicação do segundo semestre de 2012.

e práticas utilizados. Pudemos observar que os novatos geralmente utilizam a lista de e-mails como meio de entrada, o que era o esperado e está de acordo com o *onion model*. Encontramos poucos novatos que iniciaram suas ações por meio do gerenciador de tarefas.

Com relação à permanência no projeto, em geral, os novatos observados permaneceram por poucos meses, independentemente do meio de entrada utilizado. Quanto àqueles que permanecem no projeto, não se observou a migração de um meio para outro ao longo do tempo. Uma consequência da reduzida quantidade de novatos que migraram de um meio para outro, é que não foi possível detectar um padrão de migração para os mesmos.

Quanto à identificação de padrões de interação dos novatos com a comunidade, observou-se que a maioria das interações são com veteranos. No entanto, muitos novatos sequer conseguem uma resposta, e abandonam-no, corroborando com os achados de (STEINMACHER et al., 2013). Como esperado também, não observamos interações significativas entre novatos.

Quanto à identificação de padrões de migração de novatos dentro de uma rede social ao longo do tempo, a quantidade de desenvolvedores que tiveram interações frequentes não foi o suficiente para chegarmos a um padrão e a conclusões consistentes.

Talvez o projeto Hadoop Common não tenha um perfil que atraia novatos que queiram contribuir por períodos longos. A caracterização das interações em problemas e soluções, talvez não tenha sido a melhor maneira para conseguir captar as características essenciais

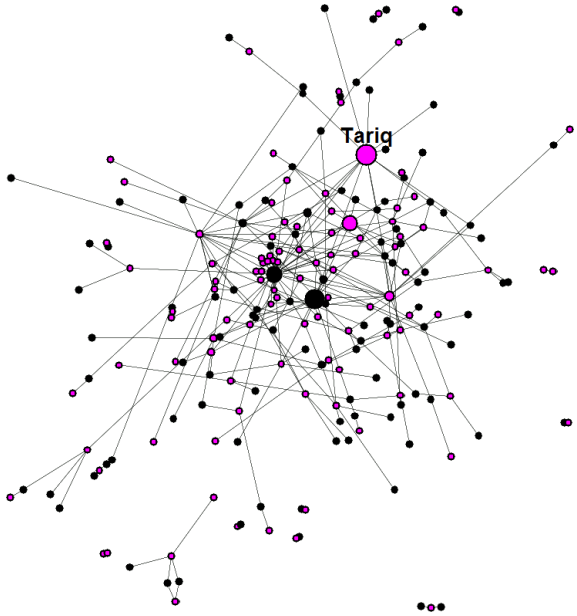


Figura 10. Rede representando a comunicação do primeiro semestre de 2013.

para obtenção de padrões para novatos. Como trabalhos futuros visamos analisar projetos mais populares, com o objetivo de melhor explorar a temporalidade dos eventos.

Referências

BALIEIRO, M. A.; SOUSA, S. de; PEREIRA, L.; SOUZA, C. R. B. de. Ossnetwork: Um ambiente para estudo de comunidades de software livre usando redes sociais. In: **Experimental Software Engineering Latin America Workshop**. São Paulo, SP, Brasil: [s.n.], 2007. p. 33–44.

CAMPOS, A. O que é software livre. **Fundação para o software livre**, v. 11, n. 09, 2006. Disponível em: `http://www.gnu.org/philosophy/free-sw.pt.html`.

CUBRANIC, D.; MURPHY, G. C.; SINGER, J.; BOOTH, K. S. Hipikat: A project memory for software development. **IEEE Transactions on Software Engineering**, IEEE, Piscataway, NJ, EUA, v. 31, n. 6, p. 446–465, jun. 2005. ISSN 0098-5589.

DAFFARA, C. Business models in floss-based companies. May 2007.

DAGENAIS, B.; OSSHER, H.; BELLAMY, R. K. E.; ROBILLARD, M.; VRIES, J. Moving into a new software project landscape. In: **32nd International Conference on Software Engineering**. New York, NY, EUA: ACM, 2010. v. 1, p. 275–284. ISSN 0270-5257.

HE, P.; LI, B.; HUANG, Y. Applying centrality measures to the behavior analysis of developers in open source software community. In: **2nd International Conference on Cloud and Green Computing**. Xiangtan, Hunan, China: IEEE Computer Society, 2012. p. 418–423.

JACOMY, M.; HEYMANN, S.; VENTURINI, T.; BASTIAN, M. Forceatlas2, a graph layout algorithm for handy network visualization. **Paris http://www. medialab. sciences-po. fr/fr/publications-fr**, 2011.

KOLASSA, C.; RIEHLE, D.; SALIM, M. The empirical commit frequency distribution of open source projects. In: **2013 International Symposium on Open Collaboration**. Hong Kong, China: 2013 International Symposium on Open Collaboration.

MADEY, G.; FREEH, V.; TYNAN, R. The open source software development phenomenon: An analysis based on social network theory. In: **Americas Conference on Information Systems (AMCIS2002)**. Dallas, TX, EUA: Idea Group Publishing, 2002. p. 1806–1813.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M. Estudo de ferramentas de mineração, visualização e análise de redes sociais. **COPPE/UFRJ**, Rio de Janeiro, RJ, Brasil, p. 49, 2010.

PARK, Y.; JENSEN, C. Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers. In: **5th IEEE International Workshop on Visualizing Software for Understanding and Analysis**. Corvallis, OR, EUA: IEEE Computer Society, 2009. p. 3–10.

SCACCHI, W. Understanding the requirements for developing open source software systems. **IEE Proceedings Software**, IEEE, Irvine, CA, EUA, v. 149, n. 1, p. 24–39, 2002.

STEINMACHER, I.; WIESE, I. S.; CHAVES, A. P.; GEROSA, M. A. Newcomers withdrawal in open source software projects: analysis of Hadoop Common project. In: **9th Brazilian Symposium on Collaborative Systems (SBSC)**. Sao Paulo, SP, BRA: IEEE Computer Society, 2012. p. 65–74.

STEINMACHER, I.; WIESE, I. S.; CHAVES, A. P.; GEROSA, M. A. Why Do Newcomers Abandon Open Source Software Projects? In: **Workshop on Cooperative and Human Aspects of Software Development**. San Francisco, CA, EUA: IEEE Computer Society, 2013. p. 1–8.

TRINDADE, C. d.; BARBOSA, Y. A. M.; MORAES, A. K. O.; ALBUQUERQUE, J. O. d.; MEIRA, S. R. d. L. An expert recommender system to distributed software development: Requirements, project and preliminary results. In: **Proceedings of the 2009 Simpósio Brasileiro de Sistemas Colaborativos**. Washington, DC, EUA: IEEE Computer Society, 2009. p. 161–168. ISBN 978-0-7695-3918-8.

WASSERMAN, S.; FAUST, K. **Social Network Analysis: Methods and Applications**. Cambridge University Press, 1994. (Structural Analysis in the Social Sciences). ISBN 9780521387071. Disponível em: `http://books.google.com.br/books?id=CAm2DpIqRUIC`.

YE, Y.; KISHIDA, K. Toward an understanding of the motivation of open source software developers. In: **25th International Conference on Software Engineering**. Washington, DC, EUA: IEEE Computer Society, 2003. p. 419–429. ISSN 0270-5257.