UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

ANA CLAUDIA MACIEL

PADRÕES DE SOCIALIZAÇÃO DE NOVATOS EM PROJETOS DE SOFTWARE LIVRE

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO - PR

2013

ANA CLAUDIA MACIEL

PADRÕES DE SOCIALIZAÇÃO DE NOVATOS EM PROJETOS DE SOFTWARE LIVRE

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Tecnólogo em Tecnologia em Sistemas para Internet.

Orientador: Prof. Dr. Marco Aurélio Graciotto Silva

Co-orientador: Prof. Me. Igor Fábio Steinmacher

CAMPO MOURÃO - PR

2013

AGRADECIMENTOS

Aos meus pais, Zeni Fatima dos Santos Maciel e Luiz Ferreira Maciel, e ao meu irmão, Andre Luiz Maciel, que com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Aos professores Marco Aurélio Graciotto Silva e Igor Fábio Steinmacher pela paciência nas orientações e incentivo que tornaram possível a conclusão desta monografia.

Aos amigos e colegas de turma pelo incentivo e pelo apoio constantes.

RESUMO

MACIEL, Ana Claudia. PADRÕES DE SOCIALIZAÇÃO DE NOVATOS EM PROJETOS DE SOFTWARE LIVRE. 32 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2013.

Comunidades que mantêm projetos de software livre demandam a colaboração de voluntários e necessitam da entrada contínua de novatos para sua continuidade. No entanto, os novatos enfrentam dificuldades e obstáculos ao iniciar sua interação em um projeto. Este trabalho utiliza um método iterativo, dividido em etapas e baseado em mineração de repositórios de software e análise de redes sociais e tem o objetivo de encontrar padrões de socialização de novatos em projetos de software livre. O projeto analisado foi o Hadoop Common a partir de mensagens e tarefas realizadas até desde janeiro de 2006 até dezembro de 2012. Em geral 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.

Palavras-chave: novato, software livre, padrões de interação, análise de redes sociais, mineração de repositórios de software

ABSTRACT

MACIEL, Ana Claudia. SOCIALIZATION PATTERNS OF NEWCOMERS IN OPEN SOURCE SOFTWARE PROJECTS. 32 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Campo Mourão - PR, 2013.

Open source software projects are based on volunteers collaboration and require a continuous influx of newcomers for their continuity. However, newcomers face difficulties and obstacles when starting their contributions. Using an iterative method based upon mining of software repositories and social network analysis, we aim the detection of socialization patterns for newcomers in open source software projects. As research subject, we use the Apache project Hadoop Common. We analysed messages and issues through december 2012. 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 fruitfull interactions, we could not detect further socialization patterns.

Keywords: newcomers, open source software, interaction pattern, social network analysis, mining software repositories

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

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 dentro de uma rede social ao longo do tempo;
- Identificar padrões de interação entre os desenvolvedores, mas 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.

O restante deste trabalho organiza-se da seguinte forma. O referencial teórico é apresentado no Capítulo 2. O Capítulo 3 apresenta o método da pesquisa, detalhando cada um dos seus passos. No Capítulo 4 são apresentados os resultados obtidos a partir da realização deste trabalho. O resumo das contribuições deste estudo e suas limitações são descritas no Capítulo 5.

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

2 REVISÃO BIBLIOGRÁFICA

Inicialmente foi realizada uma revisão da literatura com o objetivo de encontrar estudos que se relacionam com o objetivo deste trabalho. Foi necessário conhecer conceitos relacionados à mineração de repositórios de software e análise de redes sociais. Tais conceitos estão apresentados na Seção 2.1. Foram também buscados e analisados trabalhos relacionados a padrões de entrada de novos voluntários em projetos de software livre, os quais são discutidos na Seção 4.5.

2.1 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os conceitos relevantes para o entendimento do presente trabalho. Serão abordados conceitos relacionados à Mineração de Repositórios de Software e à Análise de Redes Sociais.

2.1.1 MINERAÇÃO DE REPOSITÓRIOS DE SOFTWARE

A mineração de dados refere-se à extração de conhecimento útil e previamente desconhecido de grandes quantidades de dados, por meio da aplicação de algoritmos que extraem modelos e padrões representativos (FAYYAD et al., 1996). A mineração de repositórios de software (MSR, do inglês *Mining Software Repositories*) pode ser considerada um tipo específico de mineração de dados, que tem como fonte os dados relacionados ao processo de desenvolvimento de software. Isso inclui dados de sistemas de versão de código fonte, listas de e-mail, sistemas de gerenciamento de tarefas, fóruns e documentação do software.

De acordo com Godfrey et al. (2009), a mineração de repositórios de software é uma técnica utilizada na área de engenharia de software focada na análise e compreensão dos repositórios de dados relacionados a um projeto de desenvolvimento de software. O principal objetivo da MSR é fazer uso inteligente dos dados de repositórios de software para buscar resultados das interações diárias dos membros do projeto, mudanças evolutivas no código fonte, casos de teste,

relatórios de tarefas e documentos de requisitos (THOMAS, 2011).

Segundo Côrtes et al. (2002), a MSR é composta por várias etapas. O primeiro passo é o entendimento do objetivo que se deseja atingir com a análise dos dados. Em seguida, é necessário conhecer os dados, identificando quais são relevantes para o problema em questão. Posteriormente, os dados devem ser filtrados e preparados para a execução dos algoritmos de mineração. Em seguida, podem-se aplicar os mecanismos e algoritmos desejados para gerar novas informações ou conhecimentos, bem como pode-se utilizar ferramentas gráficas para a visualização e análise dos resultados. Estes três últimos passos estão apresentados na 1 sendo que o passo inicial se encontra na base da figura.

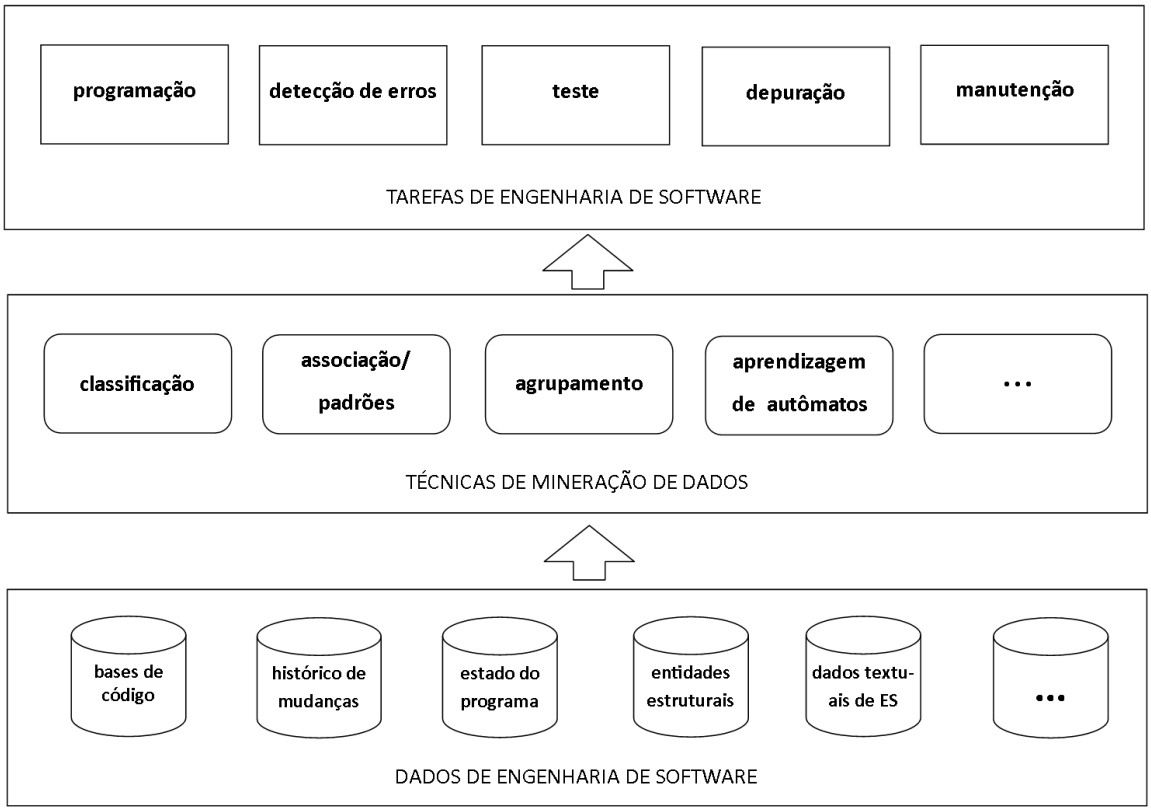


Figura 1: Visão geral de mineração de dados

A mineração de repositórios de software é útil para entender diferentes fatores do processo de desenvolvimento de software. Os estudos de mineração de repositórios de software têm contribuído na descoberta de informações importantes sobre o desenvolvimento de software e a sua evolução, considerando tanto os aspectos técnicos quanto sociais, tal como apresentado na Figura 1. Os estudos encontrados na literatura indicam o interesse por diversos tópicos, podendo-se destacar: estudos relacionados a propagações de alterações, erros no código fonte e previsão e identificação de erros (DAVIES et al., 2010; LAMKANFI et al., 2010); compreensão da dinâmica dos times de desenvolvimento e evolução de software (ROBLES, 2010; JURISTO;

VEGAS, 2010); mineração e extração de redes e métricas sociais a partir de repositórios de artefatos de software (COSTA et al., 2009; SOUSA et al., 2009; JUNIOR et al., 2010).

2.1.2 ANÁLISE DE REDES SOCIAIS

Uma rede social é definida por Wasserman e Faust (1994) como um conjunto finito de atores que compartilham algum tipo de relacionamento entre eles. Em uma representação de rede social na forma de grafo, os nós representam os atores e as arestas correspondem aos possíveis relacionamentos entre eles. A semântica do relacionamento depende da análise que se deseja conduzir nesta rede. A Figura 2 apresenta um exemplo de uma rede social genérica, representada como um grafo não direcionado de grau 14.

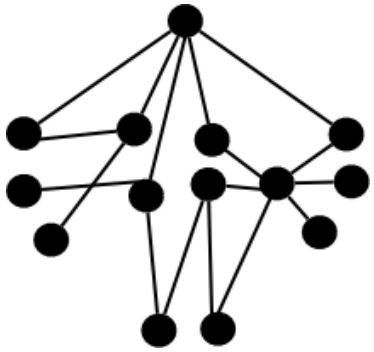


Figura 2: Exemplo de uma rede social.

Análise de Redes Sociais é a aplicação de técnicas matemáticas para estudar redes sociais, não se concentrando nos atributos de nós, mas sim em suas relações (arestas) (WASSERMAN; FAUST, 1994). Por exemplo, um nó da rede é considerado proeminente se os seus relacionamentos o tornam particularmente visível aos outros nós (MAGDALENO et al., 2010). Dessa forma, a utilização de técnicas de análise de redes sociais permite o estudo dos relacionamentos entre os desenvolvedores de uma comunidade e o entendimento de seus aspectos estruturais (SOUSA et al., 2008).

Magdaleno et al. (2010) explica o processo da análise de redes sociais da seguinte forma. O primeiro passo é definir o objetivo da análise e estabelecer a semântica dos nós e arestas da rede que se deseja analisar. O próximo passo é a coleta de dados para a construção da rede social. Nos projetos de desenvolvimento de software livre, é comum o uso de *parsing* nas páginas dos projetos e nas suas listas de e-mail e fóruns de discussão.

No desenvolvimento de software, utiliza-se a análise de redes sociais para entender a colaboração entre os membros da equipe de desenvolvimento. Essa análise pode ser feita visual ou analiticamente. Com a criação de representações visuais é possível explorar a sofisticada

capacidade visual do ser humano para facilitar a exploração e aquisição de informações úteis contidas nos dados. No caso de engenharia de software, é possível oferecer uma percepção sobre o que está acontecendo com as interações ou colaboração do grupo. Entretanto, sozinha, a visualização não permite a total compreensão da colaboração existente entre os atores da rede e precisa ser complementada para identificar características analíticas da rede. (MAGDALENO et al., 2010).

Analiticamente, três fatores são importantes: o nó, o grupo e o nível da rede. Propriedades relativas aos atores são extraídas dos nós. Nos grupos, agrupa-se os elementos de uma rede e as propriedades das sub-redes. O nível da rede está relacionado às propriedades da rede global, tais como centralidade e densidade (BRANDES; WAGNER, 2003).

A centralidade tenta descrever as propriedades da localização de um ator na rede. Estas medidas levam em consideração as diferentes maneiras em que um ator interage e se comunica com o restante da rede. A centralidade de um ator que está em uma rede social pode ser calculada de diferentes formas (MAGDALENO et al., 2010):

- Centralidade de grau: a centralidade de grau do nó está relacionada ao número de relações que este nó mantém na rede.
- Centralidade de proximidade: esta propriedade é inversamente relacionada com a distância, ou seja, quanto mais diminui a distância de um vértice para o restante da rede, maior sua centralidade de proximidade.
- Centralidade de intermediação: a centralidade de intermediação é medida pelo número de vezes que o nó aparece no caminho de outros nós (as interações entre dois nós não adjacentes dependem dos nós que se localizam no caminho entre eles).

A densidade da rede explora diretamente as propriedades da rede como um todo. Ela está relacionada à quantidade de arestas que mantém interligado um conjunto de nós. Quanto mais arestas existir numa rede, mais densa ela será (MARTINHO, 2003).

Além da densidade e centralidade, existem ainda outras métricas que podem ser utilizadas para analisar numericamente as redes sociais. Dentre elas, pode-se citar coesão, multiplexidade, diâmetro e eficiência global. Porém, não serão detalhadas aqui pois não se encaixam no escopo do presente trabalho.

2.2 TRABALHOS RELACIONADOS

Existem vários trabalhos na literatura que estudam a entrada de novatos e o processo de migração de colaboradores em projetos de software livre. Nesta seção analisaremos alguns desses trabalhos e sua relação com o presente estudo.

Hong et al. (2011) comparam redes sociais de desenvolvedores (DSN - Developers Social Network) com redes sociais gerais (no caso o Facebook, Twitter e Cyworld) e analisam como as DSNs evoluem ao longo do tempo diante de acontecimentos dentro de um projeto. Esses acontecimentos podem ser lançamento de novo software ou a saída de desenvolvedores proeminentes. Para a estrutura das redes foi considerado que quanto mais conexões existir numa estrutura, maior sua modularidade. Redes sociais de desenvolvedores foram criadas com dados extraídos do gerenciador de erros do projeto Mozilla[1] considerando que desenvolvedores que comentaram sobre o mesmo erro tinham alguma relação, desenvolvedores que possuíam menos de três interações eram excluídos da rede. Considerou-se membro do núcleo aquele que tem privilégio de enviar códigos, contribui com quantidades não triviais de código e permanece ativo no projeto por um período de anos. Concluiu-se que houve uma mudança brusca no número de desenvolvedores após a versão 1.0 do Mozilla, mas que depois do lançamento da primeira versão, esse número ficou estável. Apesar de Hong et al. (2011) analisarem como as redes evoluem ao longo do tempo, eles não estudam o comportamento dos membros isoladamente.

Costa et al. (2009) apresentam a ferramenta Transflow, que tem por objetivo analisar a evolução do software e dos desenvolvedores dentro dos projetos. Os projetos JEdit[2], MegaMek[3] e JBoss[4] são analisados para demonstrar diferentes funcionalidades da ferramenta. Uma das formas utilizadas para investigar a evolução dos desenvolvedores nos projetos é a medida do número de arquivos de código fonte adicionados ou modificados. A identificação de como os desenvolvedores começam a modificar arquivos centrais do projeto é tida como fundamental para a compreensão da evolução dos desenvolvedores no projeto. Os desenvolvedores podem optar por especialização, em que modificam os mesmos módulos ao longo do tempo, ou generalização, que é visto como um caminho para o desenvolvedor adquirir um melhor conhecimento da arquitetura, modificando vários módulos. No estudo de Costa et al. (2009), grupos do núcleo e da periferia são definidos de acordo com o padrão de interações entre os desenvolvedores: o grupo cujas interações da rede são densas e coesas é considerado o núcleo do projeto, enquanto o que é escasso e desconectado é a periferia.

[1] http://www.mozilla.org/
[2] http://www.jedit.org/
[3] http://megamek.info/
[4] http://www.jboss.org

Assim como neste trabalho, um aspecto importante a ser considerado na análise do desenvolvimento OSS é quando os desenvolvedores realizam as primeiras contribuições. Ao analisar como e onde os desenvolvedores começam a contribuir com o projeto, é possível identificar padrões de união que podem ser usados como uma referência para os novos desenvolvedores que querem saber que parte do software que pode começar a modificar. Entretanto, os padrões não são apresentados no estudo de Costa et al. (2009), devido ao objetivo do trabalho apresentar as funcionalidades da ferramenta Transflow e não a evolução dos membros no projeto.

O trabalho descrito por Sousa et al. (2009) tem como objetivo combinar múltiplas redes sociais para estudar a evolução de projetos de software livre, ou seja, para encontrar como diferentes redes sociais influenciam ou estão relacionadas umas com as outras. Para a visualização das redes sociais, foi utilizada a ferramenta Sargas. O projeto analisado para o estudo de caso deste trabalho foi o PMD[5], para o qual foram extraídos os dados para criar e analisar quatro diferentes redes sociais:

- rede social baseada na lista de discussão dos usuários;
- rede social baseada na lista de discussão dos desenvolvedores;
- rede social extraída de discussões sobre as tarefas;
- rede social extraída do código fonte.

Para a criação das redes sociais, foram extraídas informações para cada desenvolvedor que a produziu, gerando dados considerando desenvolvedor versus atividade exercida no projeto. Foram identificados seis grupos diferentes. O primeiro grupo representou atores que estavam em três ou quatro redes sociais ao mesmo tempo. O segundo grupo representou membros do projeto que estavam na lista de discussão dos usuários e redes de tarefas. O terceiro grupo representou os atores que atuavam como desenvolvedores e redes de discussão dos usuários. O quarto grupo representou o conjunto de atores que eram desenvolvedores e estavam na rede de tarefas, mas que não estavam presentes na rede de código fonte. O quinto grupo apresentou os atores que estavam na rede de código fonte e na rede de tarefas. Por fim, o último grupo é dos atores que apareceram em apenas uma das redes.

Embora identifique qual atividade o membro desenvolve dentro do projeto, não é feita a análise de como os membros evoluem ou permanecem no projeto durante um determinado período de tempo. O estudo limita-se a detectar padrões de comportamento de pessoas de acordo

[5]http://pmd.sourceforge.net/

com algumas características de cada rede social e combinações. De forma similar, poderiam ser detectados padrões de comportamento para novatos (observando outras características e combinações de redes sociais).

O estudo de Steinmacher et al. (2012) teve como objetivo se aprofundar nas razões pelas quais os novatos desistem. Os dados utilizados foram obtidos da lista de discussões de desenvolvedores e dos comentários provenientes do gerenciador de tarefas do Hadoop Common. O primeiro passo do estudo foi a coleta dos dados do gerenciador de tarefas, extração dos dados de e-mails e análise dos dados referentes aos novatos no projeto. Ao fim do estudo, foi concluído que menos de 20% dos novatos continuaram. A desistência é influenciada pelos autores das respostas e pelo tipo da resposta. A ausência de resposta não é fator relevante para a desistência. O trabalho de Steinmacher et al. (2012) analisa a evolução dos novatos dentro do projeto, entretanto não verifica se há um padrão de entrada ou migração para os novos membros.

No estudo apresentado por He et al. (2012) é analisado o comportamento dos desenvolvedores dentro de uma comunidade de software livre. Foram estudados projetos hospedados no SourceForge[6]. Em suma, foram analisados quatro tipos de padrões de colaboração sobre os desenvolvedores: (i) entre novos desenvolvedores; (ii) entre novo desenvolvedor e desenvolvedor existente; (iii) entre os desenvolvedores existentes que não tinham colaborado antes; (iv) entre os desenvolvedores existentes que já colaboraram antes.

O resultado mostrou que o número de novos desenvolvedores que colaboraram com outros novos membros é maior do que aqueles que colaboraram com desenvolvedores já existentes no projeto e que algumas colaborações foram desenvolvidas entre desenvolvedores já existentes no projeto com base na sua experiência em colaborações anteriores. Apesar de muito similar à proposta do presente trabalho, a forma utilizada na construção da rede dos desenvolvedores, considerando que todos os membros que participam de um mesmo projeto possuem relação, não é apropriada, pois eles podem estar no mesmo projeto e não realizar interação alguma, e também não é realizado um estudo para investigar a permanência ou não do membro no projeto.

2.3 CONSIDERAÇÕES FINAIS

A mineração de repositórios de software utiliza como fonte os dados relacionados ao processo de desenvolvimento de software. Os estudos de mineração de repositórios de software têm contribuído na descoberta de informações importantes sobre o desenvolvimento de software

[6]http://sourceforge.net/

e a sua evolução. A utilização de técnicas de análise de redes sociais, em combinação com MSR, permite entender a colaboração entre os membros da equipe de desenvolvimento com as informações obtidas com a mineração de repositórios de software.

Tabela 1: Relação dos trabalhos relacionados com os pontos de interesse desta monografia.

	desenvolvedores	novatos	padrões	temporal	SNA
Hong et al. (2011)	+			+	+
Costa et al. (2009)	+	+			
Sousa et al. (2009)	+				
Steinmacher et al. (2012)		+			
He et al. (2012)		+	+		

Os principais pontos trabalhados pelos trabalhos relacionados na seção anterior são resumidos na Tabela 1. Observa-se a carência de trabalhos relacionados com novatos e a detecção de padrões, focando-se nos desenvolvedores como um todo. No capítulo seguinte deste trabalho, analisaremos especificamente os novatos, identificando padrões de interação relevantes para projetos de software livre.

3 MÉTODO

Após a revisão da literatura, definiu-se o restante do método da pesquisa a ser conduzida para a conclusão deste trabalho, tal como apresentado na Figura 3. O primeiro passo consistiu na escolha do projeto de software livre a ser analisado. Posteriormente recuperamos os dados do projeto escolhido para que, na próxima etapa, fosse realizada a extração dos dados do repositório de software. Conforme apresentado na Figura 3 representamos as redes sociais baseadas nas interações dos membros no projeto escolhido anteriormente. Por fim, conduzimos a análise da rede social. Após o passo 6, o estudo pode ser finalizado ou pode ser feita a mineração novamente para gerar outra representação e análise da rede social em busca de novos resultados no mesmo projeto. Há também a opção de voltar ao passo 2, iniciando-se uma nova iteração para a análise de outro projeto de software livre, verificando-se se os padrões identificados também são válidos para o novo projeto; ou se o modelo precisa ser ajustado com novos padrões, ou ainda se modelos distintos devem ser extraídos para cada projeto. Neste trabalho, foi realizada apenas uma iteração, restringindo-se a um projeto.

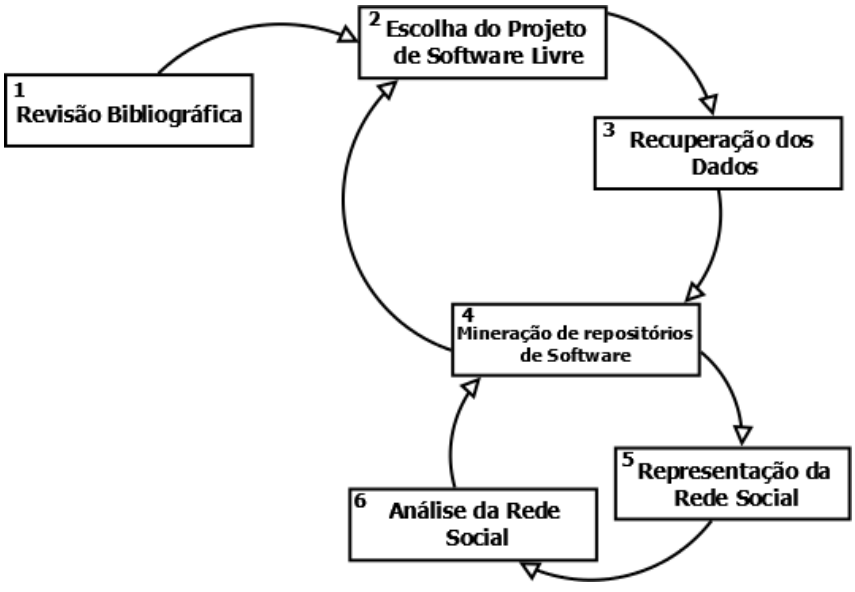


Figura 3: Método da pesquisa.

3.1 EXTRAÇÃO DOS DADOS

3.1.1 ESCOLHA DO PROJETO A SER ANALISADO

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/`.

3.1.2 ESPECIFICAÇÃO DOS DADOS A SEREM EXTRAÍDOS

Após a escolha do projeto, o próximo passo do método será a recuperação dos dados, conforme apresentado no Passo 2 na Figura 3. Para a análise serão 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, será 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). A Figura 4 apresenta o modelo dos dados recuperados e suas respectivas relações.

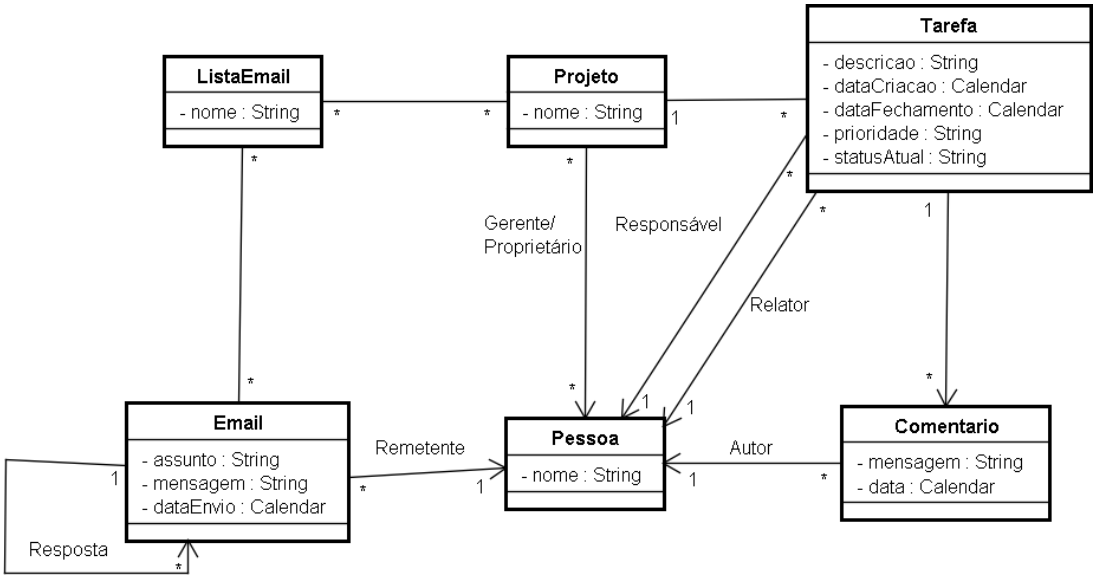


Figura 4: Modelo relacional dos dados recuperados de projeto OSS.

Para extrair os dados de e-mails, primeiramente serão obtidos os arquivos que contêm todos os e-mails, incluindo cabeçalho e corpo da mensagem. As informações das mensagens contidas nos arquivos serão 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 serão reconstruídas verificando o campo `In-reply-to` do cabeçalho bem como o assunto do e-mail (examinando 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 serão armazenados em um banco de dados local contendo os detalhes das mensagens extraídas.

Para a análise das mensagens de e-mail, serão 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.

3.2 ANÁLISE DOS DADOS

3.2.1 ESTRUTURAÇÃO DA REDE SOCIAL

Os dados obtidos serão 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 serão mesclados em uma única rede por meio dos autores das mensagens enviadas.

Para analisar a migração de determinado membro do projeto será 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.

3.2.2 ANÁLISE DA REDE SOCIAL

Na rede social resultante serão analisadas as interações dos membros de acordo com o contexto em que a interação foi realizada. A representação dos membros dessa rede levará em consideração em qual dos meios o membro apareceu, seja de maneira isolada, seja concorrentemente em ambos os meios. Na Figura 5 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.

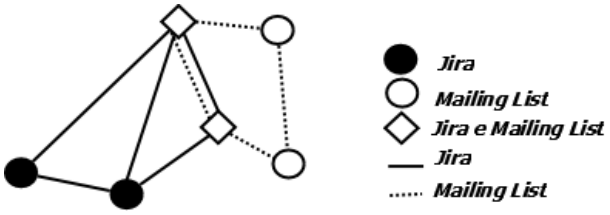


Figura 5: Representação de uma rede social com interação no Jira e lista de e-mails.

Serão 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 6. 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.

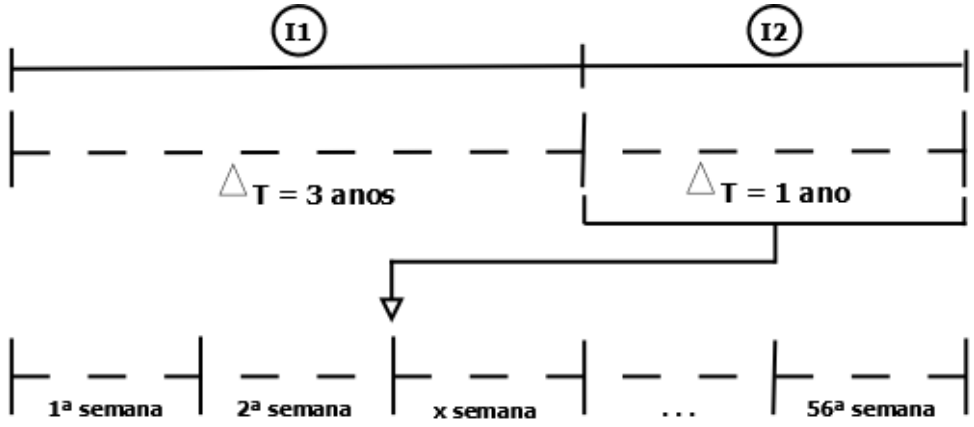


Figura 6: Linha do tempo para criação das redes sociais.

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 7, 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.

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.

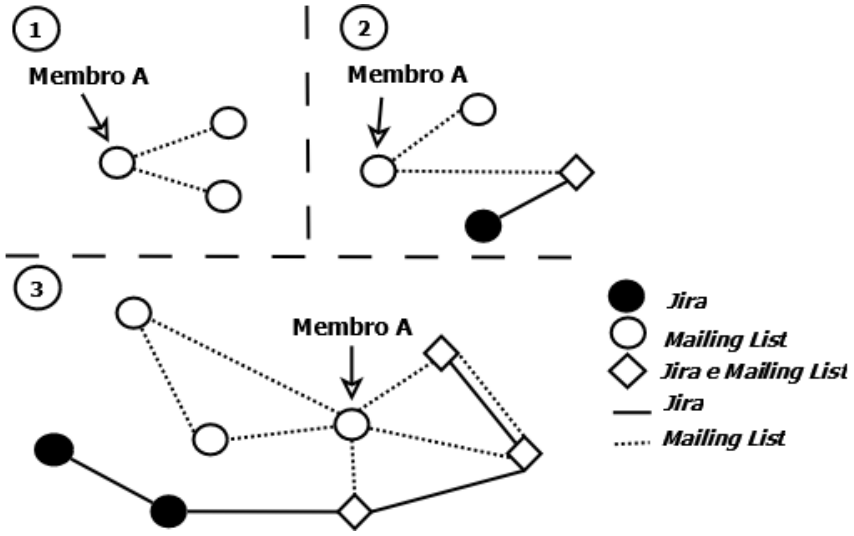


Figura 7: Possível migração temporal de um membro na rede social.

Quanto a centralidade de intermediação, duas medida 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.

4 RESULTADOS

O método definido no Capítulo 3 foi aplicado, escolhendo-se o projeto Hadoop Common. 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.

4.1 ESCOLHA DO PROJETO A SER ANALISADO

Considerando as observações feitas na subseção 3.1.1, o projeto Hadoop Common, um dos subprojetos do Hadoop, foi escolhido. 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.

O Apache Hadoop é um arcabouço para o armazenamento e processamento de dados em larga escala (GOLDMAN et al., 2012). A eficácia obtida pelo Hadoop pode ser constatada ao verificar a quantidade de importantes empresas, de diferentes ramos, que o utilizam, a citar *Yahoo!*, *IBM*, *Oracle* e *Facebook*. Esse sucesso está associado à comunidade de desenvolvimento, apoiada pela Apache Foundation, com amplo reconhecimento no meio de software livre.

O Hadoop oferece como ferramentas principais o MapReduce, responsável pelo processamento distribuído, e o Hadoop Distributed File System (HDFS), para armazenamento de grandes conjuntos de dados, também de forma distribuída. Em comum a estas duas ferramentas, encontra-se o Hadoop Common, que contém um conjunto de utilitários e a estrutura base que dá suporte aos demais subprojetos do Hadoop. Este último projeto é um bom candidato a receber tanto novatos advindos da comunidade de usuários do Hadoop quanto desenvolvedores dos outros projetos associados.

4.2 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.

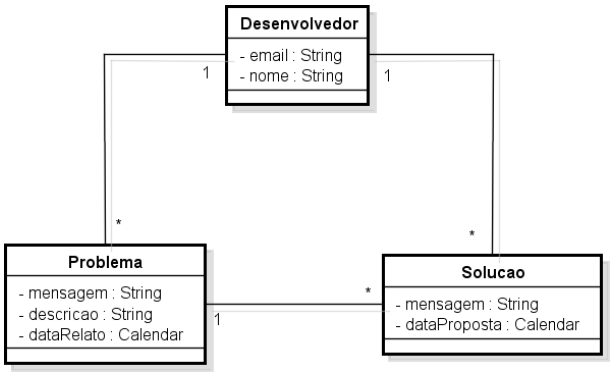


Figura 8: Modelo relacional dos dados recuperados da lista de e-mails.

Na Figura 8 apresentamos a estrutura das informações que obtivemos do projeto. Um desenvolvedor pode ser o autor de uma ou mais soluções ou problemas. Toda solução está atrelada a um problema e um problema pode não ter solução ou ter uma ou mais soluções.

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).

4.3 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 9), outro com dados da lista de e-mails (Figura 10) e o terceiro, com a união de ambos, representado na Figura 11.

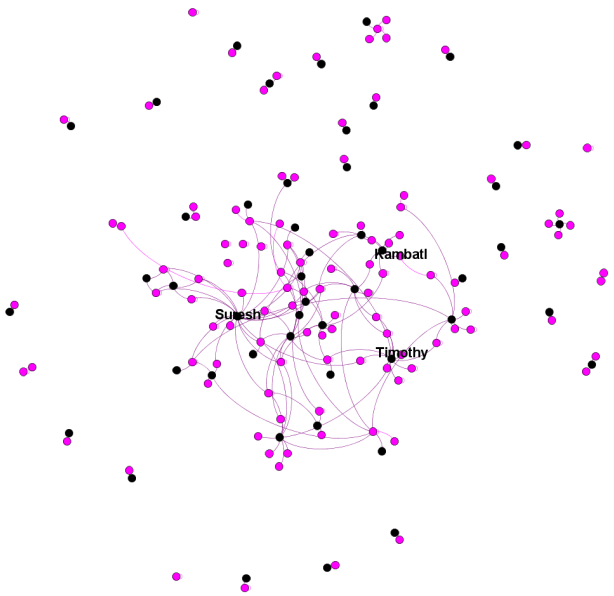


Figura 9: Rede social representando as comunicações do gerenciador de tarefas. Os nós representados pela cor rosa (tom mais claro) são os novatos e os de cor preta são os veteranos.

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 10, 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.

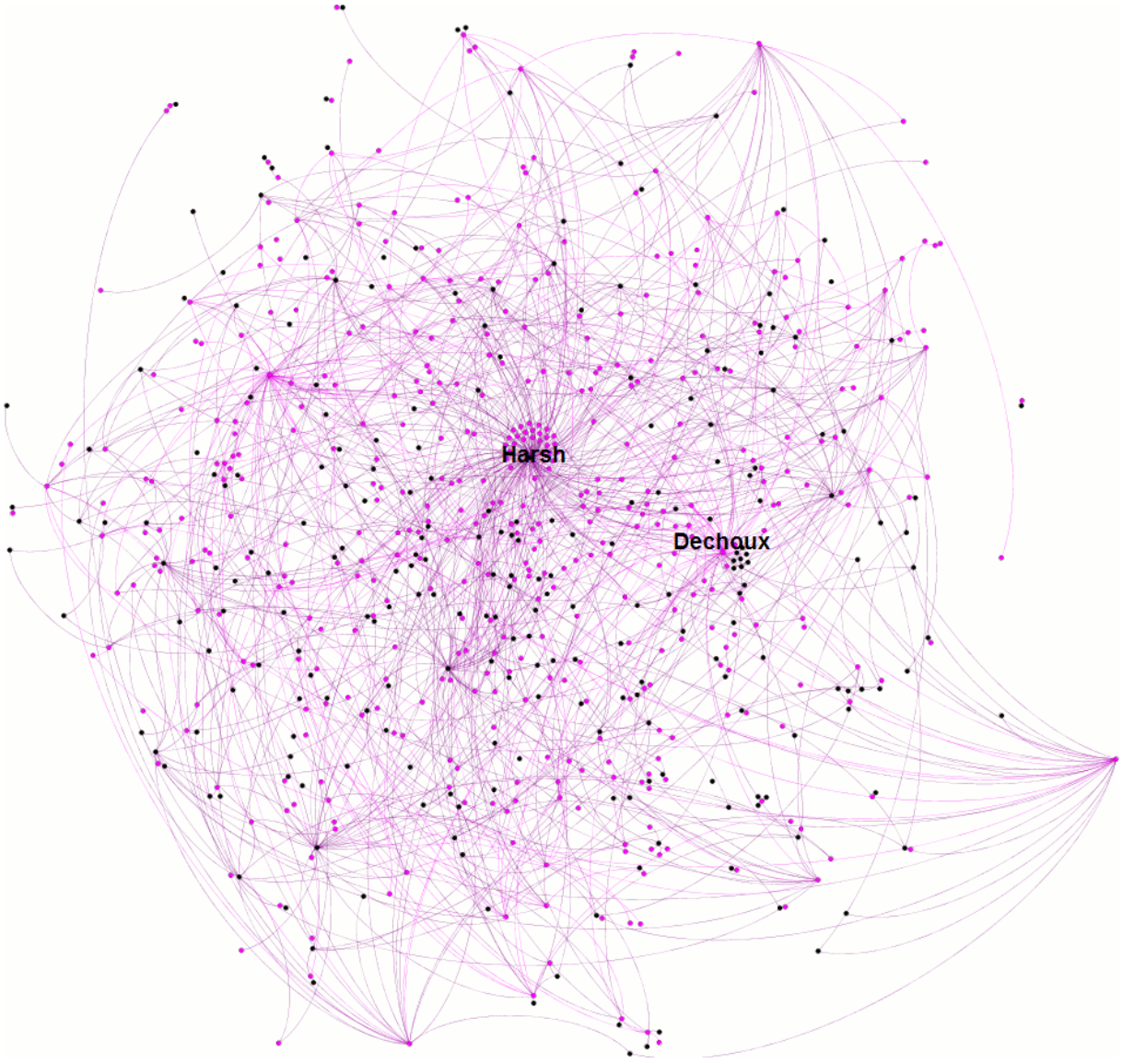


Figura 10: Rede social representando as comunicações da lista de e-mails. Os nós representados pela cor rosa (tom mais claro) são os novatos e os de cor preta são os veteranos.

Após a união dos dados das redes do Jira e dos e-mails, sem considerar a identificação de desenvolvedores duplicados entre as redes, obtivemos a rede apresentada na Figura 11. 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.

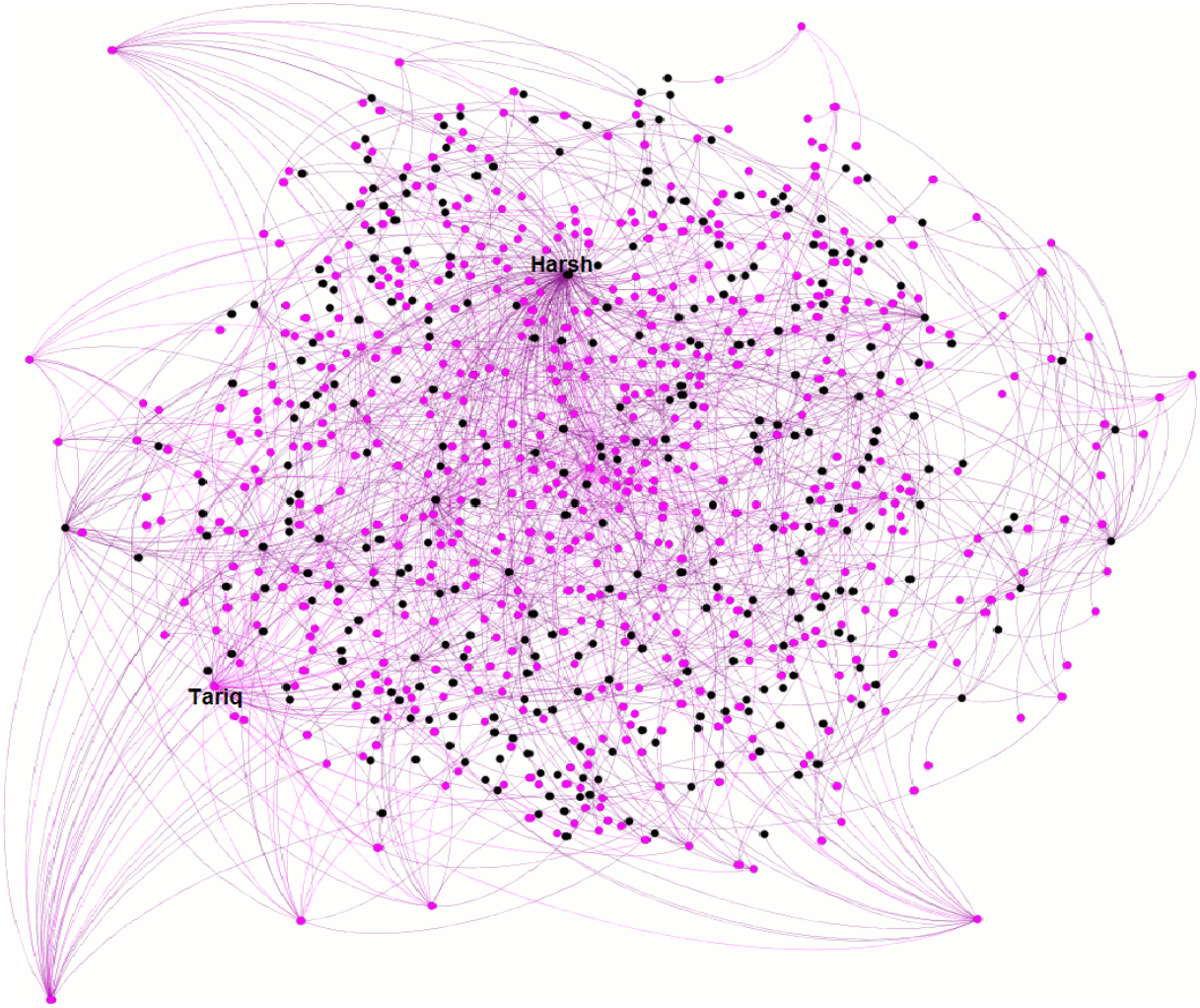


Figura 11: Rede social representando a união da lista de e-mails e gerenciador de tarefas. Os nós representados pela cor rosa (tom mais claro) são os novatos e os de cor preta são os veteranos.

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.

4.4 MEIO DE ENTRADA DOS NOVATOS

Analisando todos os novatos e suas interações no segundo semestre de 2012, cujos dados estão apresentados na Seção 4.4, 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.

Tabela 2: Quantidade de novatos por meio de entrada no segundo semestre de 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

No total, observa-se a entrada de 635 novatos no projeto no período analisado. Entretanto, resta analisar se tais números afetam a participação desses no projeto. Na Tabela 2, 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.

Tabela 3: Quantidade de novatos que interagiram entre 1 e 6 meses no segundo semestre de 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%)

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-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 suas comunicações 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 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.

4.5 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 11, 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 12 (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 12 é o desenvolvedor Dechoux, da lista de e-mails, com alto *closeness* e *betweenness*: ele interagiu com novatos e veteranos de alto *betweenness*.

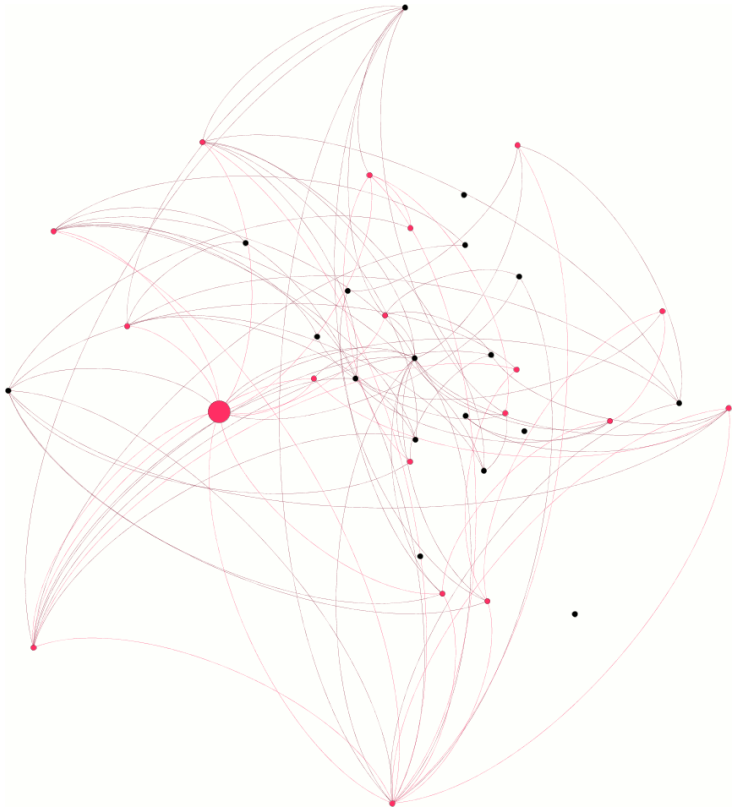
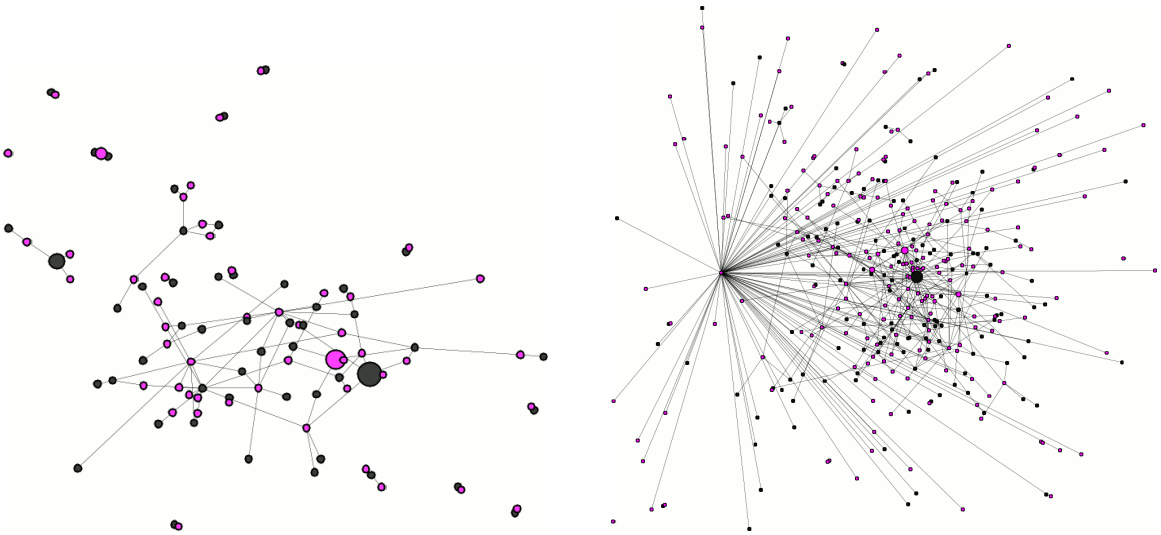


Figura 12: Rede social com nós de grau maior que dez.

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 11, 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 13. 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 14 e 15, 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.



(a) Comunicações do mês de julho/2012. (b) Comunicações do mês de agosto/2012.

Figura 13: Rede representando as comunicações dos meses de julho e agosto de 2012.

Diante dos grafos apresentados nas Figuras 14 e 15, 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 14, temos o novato Tariq, também presente na Figura 15 em conjunto de Embree, ambos da lista de e-mails.

Depois de selecionar os novatos com alto *closeness* e *betweenness*, apresentados na Tabela 4, foi feita uma análise individual para eles em busca de identificar padrões. Porém, 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.

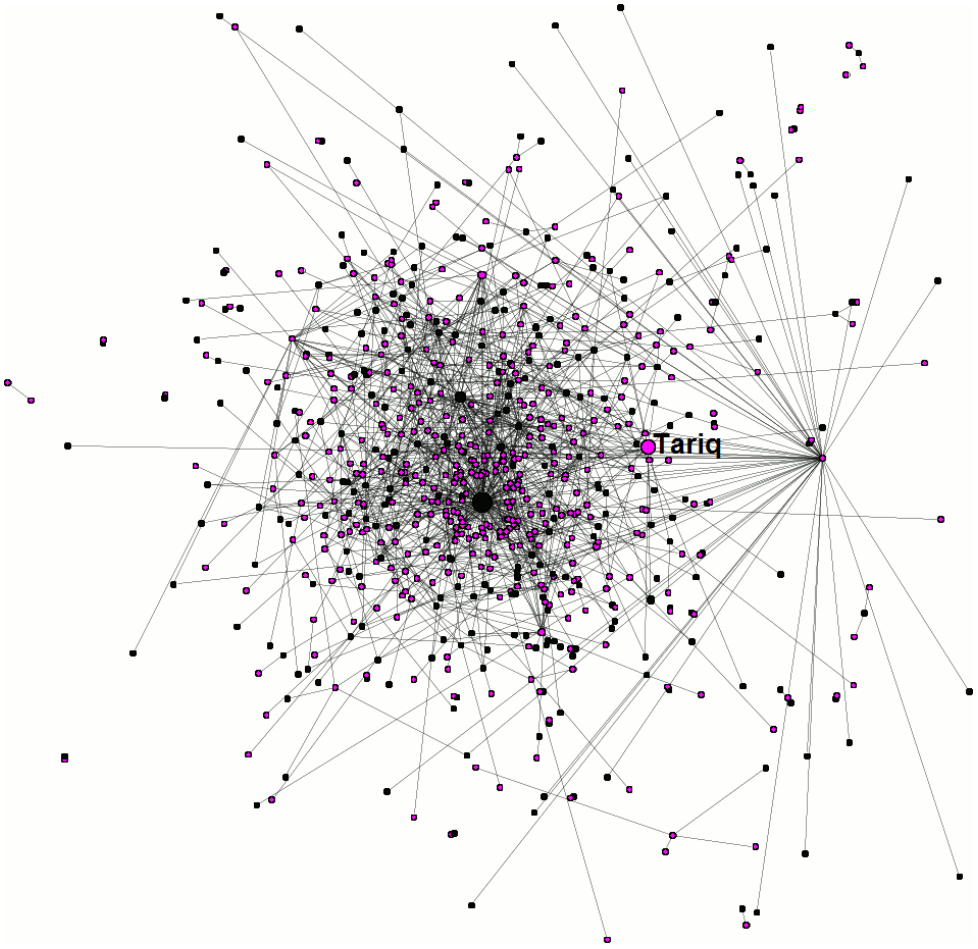


Figura 14: Rede representando as comunicações do segundo semestre de 2012.

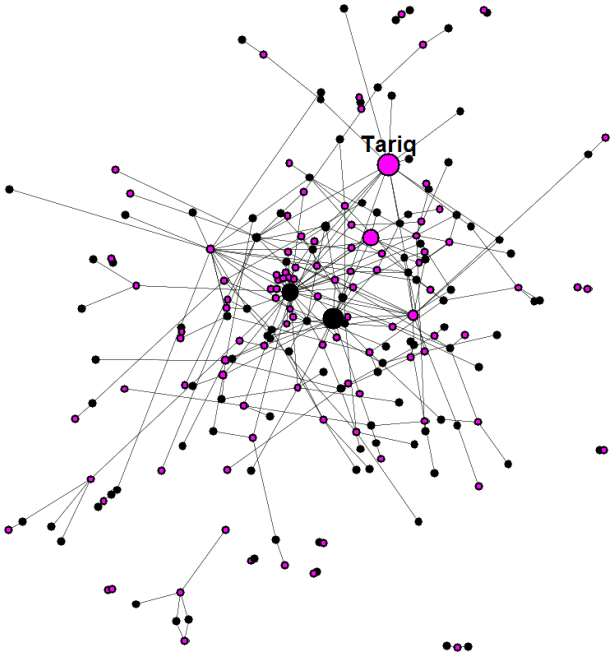


Figura 15: Rede representando as comunicações do primeiro semestre de 2013.

Tabela 4: 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.

4.6 CONSIDERAÇÕES FINAIS

As influências que os membros têm em um projeto e para a comunidade são diferentes, dependendo dos papéis que desempenham. Tomando como base o *onion model*, o desenvolvedor mais próximo do centro tem maior influência e aqueles que estão na periferia, podem, com o decorrer do tempo, migrar para outros meios através de suas contribuições na comunidade.

No entanto, nem todos os membros querem evoluir ao ponto de tornarem-se membros do núcleo. Alguns sempre serão usuários passivos, e alguns vão desistir da tarefa antes de alcançar seus objetivos.

Neste trabalho, analisamos os novatos da lista de e-mails e do gerenciador de tarefas do Hadoop Common. Apenas um desenvolvedor permaneceu nos 6 meses analisados em cada um dos meios. Poucos novatos migraram de um meio para outro.

Os desenvolvedores de alto *closeness* das redes dos dois meios não permaneceram no projeto e relacionaram-se apenas com veteranos. Observa-se que a baixa frequência de interações impede a identificação de padrões bem sucedidos de socialização dos novatos.

Dentre os fatores da rede social, as principais centralidades de intermediação foram *closeness* e *betweenness* para identificar os padrões de socialização dos novatos, bem como suas formas de comunicação com os demais membros da rede social. A análise visual, apesar de limitada, também foi utilizada neste trabalho, a fim de observar as interações entre os desenvolvedores de forma simplificada.

5 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 buscou a identificação de padrões de socialização dos novatos de modo a auxiliar tais projetos a compreender esta dinâmica e a melhorar os mecanismos e práticas utilizados. Em relação aos objetivos definidos para este trabalho, obtivemos os seguintes resultados.

Os novatos utilizam como meio de entrada a lista de e-mails, o que era o esperado, pois é um meio para facilitar o ingresso dos desenvolvedores no projeto. Foram poucos os que usaram o gerenciador de tarefas para ingresso no projeto.

Em relação à permanência no projeto, em geral, os novatos permanecem apenas poucos meses, independentemente do meio de entrada utilizado. Quanto a aqueles 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 entre os desenvolvedores, mais especificamente dos veteranos com novatos e entre os novatos, observou-se que a maioria das interações dos novatos, quando elas existem, são com veteranos. No entanto, muitos novatos sequer conseguem uma resposta à primeira interação no projeto e abandonam-no. Também não se observam 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 consistente.

5.1 LIMITAÇÕES

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 para obtenção de padrões para novatos.

5.2 TRABALHOS FUTUROS

Os trabalhos futuros incluem analisar projetos mais populares (e não apenas projetos ativos), explorar melhor a parte temporal e unificar os desenvolvedores de ambos os meios, pois na lista de e-mails o desenvolvedor utiliza como identificador o e-mail e no Jira um nome. Outro ponto a ser investigado é a representação das formas de interações em busca de um modelo alternativo utilizado neste trabalho (problema-solução).

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: . São Paulo, SP, Brasil: Experimental Software Engineering Latin America Workshop, 2007. p. 33–44.

BRANDES, U.; WAGNER, D. Visone – analysis and visualization of social networks. In: **GRAPH DRAWING SOFTWARE**. Berlim, Alemanha: Springer-Verlag, 2003. p. 321–340.

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`.

CÔRTES, S. C.; PORCARO, R.; LIFSCHITZ, S. **Mineração de dados - funcionalidades, técnicas e abordagens**. PUC, 2002. 10-15 p. (Monografias em Ciência da Computação). Disponível em: `http://books.google.com.br/books?id=uSYktwAACAAJ`.

COSTA, J. M. R.; SANTANA, F. W.; SOUZA, C. R. B. D. Understanding open source developers' evolution using TransFlow. In: **15th international conference on Groupware: design, implementation, and use**. Berlin, Alemanha: Springer-Verlag, 2009. p. 65–78. ISBN 978-3-642-04215-7.

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.

DAVIES, J.; ZHANG, H.; NUSSBAUM, L.; GERMáN, D. M. Perspectives on bugs in the debian bug tracking system. In: WHITEHEAD, J.; ZIMMERMANN, T. (Ed.). **7th International Working Conference on Mining Software Repositories**. Cape Town, África do Sul: IEEE, 2010. p. 86–89. ISBN 978-1-4244-6803-4.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, Association for the Advancement of Artificial Intelligence, Palo Alto, CA,, v. 17, n. 3, p. 37–54, set.–nov. 1996. ISSN 0738-4602.

GODFREY, M. W.; HASSAN, A. E.; HERBSLEB, J.; MURPHY, G. C.; ROBILLARD, M.; DEVANBU, P.; MOCKUS, A.; PERRY, D. E.; NOTKIN, D. Future of mining software archives: A roundtable. **IEEE Software**, IEEE Computer Society, Los Alamitos, CA, EUA, v. 26, n. 1, p. 67–70, jan. 2009. ISSN 0740-7459.

GOLDMAN, A.; KON, F.; JUNIOR, F. P.; POLATO, I.; PEREIRA, R. de F. JAI 03: Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. In: **XXXII Congresso**

da Sociedade Brasileira de Computação, XXXI Jornadas de Atualização em Informática (JAI). Curitiba, Paraná, Brasil: 1ed., 2012.

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.

HONG, Q.; KIM, S.; CHEUNG, S.; BIRD, C. Understanding a developer social network and its evolution. In: **27th IEEE International Conference on Software Maintenance**. Washington, DC, EUA: IEEE Computer Society, 2011. p. 323–332. ISSN 1063-6773.

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.

JUNIOR, M.; MENDONCA, M.; FARIAS, M.; HENRIQUE, P. OSS developers context-specific Preferred Representational systems: A initial Neurolinguistic text analysis of the Apache mailing list. In: **7th IEEE Working Conference on Mining Software Repositories**. Cape Town, África do Sul: IEEE, 2010. p. 126–129.

JURISTO, N.; VEGAS, S. Using differences among replications of software engineering experiments to gain knowledge. In: **7th IEEE Working Conference on Mining Software Repositories**. Washington, DC, EUA: IEEE Computer Society, 2010. p. 1–10.

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.

LAMKANFI, A.; DEMEYER, S.; GIGER, E.; GOETHALS, B. Predicting the severity of a reported bug. In: **7th IEEE Working Conference on Mining Software Repositories**. Cape Town, África do Sul: IEEE, 2010. p. 1–10.

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.

MARTINHO, C. **Redes - uma introduçao às dinâmicas da conectividade e da auto-organização**. Brasília: WWF - Brasil, 2003. Disponível em: `http://www.wwf.org.br/informacoes/bliblioteca/?3960`.

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.

ROBLES, G. Replicating msr: A study of the potential replicability of papers published in the mining software repositories proceedings. In: **7th IEEE Working Conference on Mining Software Repositories**. Cape Town, África do Sul: IEEE, 2010. p. 171–180.

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. ISSN 1462-5970.

SOUSA, S.; BALIEIRO, M.; COSTA, J. R.; SOUZA, C. Multiple social networks analysis of floss projects using sargas. In: **42nd Hawaii International Conference on System Sciences**. Big Island, HI, EUA: IEEE Computer Society, 2009. p. 1–10. ISSN 1530-1605.

SOUSA, S.; BALIEIRO, M. A.; SOUZA, C. R. B. Análise multidimensional de redes sociais de projetos de software livre. In: **5th Brazilian Symposium on Collaborative Systems**. Los Alamitos, CA, EUA: IEEE Computer Society, 2008. p. 23–33. ISBN 978-0-7695-3500-5.

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.

THOMAS, S. W. Mining software repositories using topic models. In: **33rd International Conference on Software Engineering**. New York, NY, EUA: ACM, 2011. p. 1138–1139. ISBN 978-1-4503-0445-0.

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.