

# Predição de falhas em projetos de software baseado em métricas de redes sociais de comunicação e cooperação

Vidal D. da Fontoura   Igor Fabio Steinmacher   Igor Scaliante Wiese   Marco Aurélio Gerosa  
UTFPR – Universidade   UTFPR – Universidade   UTFPR – Universidade   IME/USP – Universidade de  
Tecnológica Federal do Paraná   Tecnológica Federal do Paraná   Tecnológica Federal do Paraná   São Paulo  
– Campus Campo Mourão   – Campus Campo Mourão   – Campus Campo Mourão  
vidalfontoura@hotmail.com   igorfs@utfpr.edu.br   igor@utfpr.edu.br   gerosa@ime.usp.br

## ABSTRACT

This paper presents an approach to predict failures in open source projects based on social networks metrics. Social networks will be created by mining conversations from developers *mailing list* and from the comments provided on the issue manager. Cooperation social networks will be extracted from the log of versioning management system, enabling the identification of developers that worked on the same class. Matrices that represent the social networks will keep the degree of interaction among developers in each social network. Social Network Analysis (SNA) metrics will be used to calculate and identify patterns on these interactions. These values will be used as input to a Bayesian model which, after being trained, will enable build failure prediction.

## RESUMO

Este trabalho apresenta uma abordagem de predição de falhas em projetos de software livre baseado em métricas de redes sociais de comunicação e cooperação entre os desenvolvedores. Serão construídas redes sociais de comunicação extraídas a partir das conversas na *mailing list* de desenvolvedores e nos comentários realizados no sistema gerenciador de tarefas. Para a cooperação serão extraídos dados do *log* do sistema gerenciador de versão, possibilitando identificar quando desenvolvedores trabalharam em uma mesma classe. Matrizes que representem as redes sociais serão criadas para armazenar o grau de interação em cada rede social. Métricas de Análise de Redes Sociais (*Social Network Analysis*) serão utilizadas para calcular e identificar padrões existentes na interação. Estes valores serão utilizados como entrada no modelo bayesiano, que depois de treinado, possibilitará a predição de falhas em uma *build*.

## Keywords

predição de falhas, análise de redes sociais, cooperação e comunicação.

## 1. INTRODUÇÃO

Pesquisadores de engenharia de software têm uma variedade de abordagens para criar redes sociais e estudar fenômenos da área. Dentre elas, Meneely et al [5] menciona aquelas baseadas em

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.  
Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

mineração de *logs* de sistemas gerenciadores de versão [2],[7],[9],[11],[16],[8] e mineração de mensagens de diferentes meios de comunicação. [7],[16],[1].

Dentre os diversos fenômenos que podem ser estudados na engenharia de software, destacam-se aqueles relacionados à comunicação e cooperação entre os membros de uma ou mais equipes de software no desenvolvimento de suas tarefas. Para analisar aspectos de cooperação e comunicação pode-se utilizar recursos relacionados a criação e geração de redes sociais [5]. O uso de Análise de Redes Sociais (SNA-*Social Network Analysis*) possibilita descrever, caracterizar e entender redes sociais por meio da exploração de suas propriedades. [4]

Diante deste cenário este trabalho pretende estudar a comunicação e a cooperação, considerando as métricas e redes sociais utilizadas nos trabalhos relacionados, com o objetivo de verificar se é possível melhorar a predição de falhas em projetos de software livre. Os projetos selecionados para o estudo empírico serão minerados do repositório da *Apache Software Foundation* (ASF). Esta escolha foi feita a partir de uma investigação prévia, que permitiu identificar que é possível minerar informações sobre falhas, comunicação e cooperação.

Este trabalho está organizado da seguinte forma: a Seção 2 apresentará os conceitos de comunicação, cooperação e métricas de SNA comparando as abordagens dos trabalhos relacionados. A Seção 3 descreve a metodologia de coleta de dados para a realização do estudo empírico, as redes sociais que serão criadas e como será realizada a predição de falhas e sua validação. A Seção 4 apresenta os próximos passos da pesquisa e as conclusões do trabalho parcial já realizado.

## 2. CONTEXTO

Com o objetivo de prever falhas em projeto de software livre com base nas estruturas de comunicação e de cooperação dos desenvolvedores, foi realizada uma revisão bibliográfica. Nesta Seção serão abordados conceitos de cooperação, comunicação e métricas SNA.

### 2.1 Cooperação

Esta subseção apresentará como os trabalhos relacionados que utilizam dados de cooperação entre os desenvolvedores criam redes sociais e utilizam métricas de SNA para interpretar dados destas redes e prever falhas.

Fuks et al [3], define cooperação como membros de um grupo que operam em conjunto para manipular e organizar informações, construindo e refinando objetos. Segundo Steinmacher et al [12] identifica a cooperação quando um ou mais desenvolvedores interagem com artefatos compartilhados de maneira síncrona ou assíncrona.

Wolf et al [16] desenvolveu um estudo que tem por objetivo prever falhas utilizando SNA. No projeto de software estudado as tarefas e os objetivos de cada *release* são agrupados em itens de trabalho (*work items*) e atribuídos a equipes de desenvolvimento. Os itens de trabalho são desenvolvidos de forma paralela por mais de uma equipe sendo necessária a sua integração para a definição de uma *build* do projeto. Desta forma as equipes tem que cooperar entre si para concluírem os objetivos das *releases*. Assim, conexões são criadas em uma rede social considerando aqueles desenvolvedores que trabalharam em um mesmo item de trabalho.

Já Meneely et al [5] analisou o histórico dos arquivos de um projeto verificando os responsáveis pelas atualizações de cada arquivo. Identificando todos os desenvolvedores que atualizaram um determinado arquivo (classe), são criadas conexões entre esses desenvolvedores. Estas conexões indicam quais foram os desenvolvedores que cooperaram para o desenvolvimento de um artefato.

## 2.2 Comunicação

Esta subseção apresentará como os trabalhos relacionados consideram a comunicação no seu contexto, e como ela será considerada neste trabalho.

Wolf et. al [16] observou os comentários deixados pelos desenvolvedores nos *commits* dos itens de trabalho. No seu estudo ele considera somente esta forma de comunicação para criar as redes sociais.

Biçer et, al [1] minerou os comentários feitos por desenvolvedores em um gerenciador de tarefas (*issues*). Ele optou por esta comunicação porque considera que os desenvolvedores expressavam suas opiniões quando estes têm relações com tarefas.

## 2.3 Métricas de SNA

Esta seção apresentará os trabalhos relacionados que consideram o uso de SNA com intuito de prever falhas. Como visto nas subseções 2.1 e 2.2, a literatura indica que estudos podem ser realizados analisando a comunicação e cooperação para prever falhas. O Quadro 1 relaciona os estudos e as métricas de SNA utilizadas nos trabalhos descritos nas seções 2.1 e 2.2. A definição de cada uma das métricas pode ser encontrada nos trabalhos correlatos e também são apresentadas em [14].

Observa-se no Quadro 1 que diferentes métricas foram utilizadas para medir as propriedades extraídas das redes sociais dos trabalhos relacionados identificados. Em todos os trabalhos estes valores calculados a partir da SNA foram utilizados como entrada para o modelo de predição.

Métricas SNA	[16]	[5]	[1]
Centralidade	X	X	X
Densidade	X		X
<i>Structural Hole</i>	X		
Diametro			X
Coefficiente de <i>Clustering</i>			X
<i>Bridge</i>			X
<i>Characteristic Path Length</i>			X

**Quadro 1: Comparação das métricas de SNA**

## 2.4 Considerações finais

Este trabalho pretende utilizar a combinação das métricas identificadas nos trabalhos relacionados para verificar a possibilidade de melhoria da predição de falhas utilizando SNA. Outro diferencial encontra-se nos dados que serão minerados a partir da comunicação e cooperação, oferecendo uma maior quantidade de dados para criação das redes sociais. O Quadro 2 apresenta as redes sociais criadas nos trabalhos relacionados e quais serão construídas nesta proposta.

Redes Sociais	[16]	[5]	[1]	Esta proposta
Comunicação em <i>Issues</i>			X	X
Comunicação em <i>Mailing list</i>				X
Comunicação em <i>Commits</i>	X			X
Mineração de Cooperação em SVN	X	X	X	X

**Quadro 2: Comparação das redes sociais utilizadas para cálculo das métricas de SNA.**

## 3. Modelo de Predição de Falhas baseado em métricas de SNA para redes sociais de comunicação e cooperação

Três questões de pesquisa serão utilizadas para guiar a metodologia deste trabalho.

- (Q1) Métricas de SNA aplicadas a redes sociais extraídas de mecanismos de comunicação dos desenvolvedores podem auxiliar a predição de falha em projetos de software?
- (Q2) Métricas de SNA aplicadas a redes sociais extraídas de mecanismos de cooperação dos desenvolvedores podem auxiliar a predição de falha em projetos de software?
- (Q3) Resultados combinados a partir da análise das redes sociais criadas podem melhorar os resultados de predição?

Para responder estas questões um experimento será realizado. A subseção 3.1 apresentará como se pretende realizar a coleta de dados. A subseção 3.2 apresenta os passos para criação das redes sociais de comunicação e cooperação. A subseção 3.3 explica como será realizado o treinamento da rede bayesiana utilizada no modelo de predição e por fim, a subseção 3.4 discute as métricas para validação da predição realizada.

### 3.1 Coleta de dados

Será realizada a coleta de dados de projetos armazenados nos repositórios da ASF.

Neste repositório a comunicação será minerada de diferentes fontes de discussão, tais como: comentários em tarefas (*issues*), comentários feitos por um desenvolvedor no instante em que ele realiza *commits* e das discussões realizadas nos *mailing lists* de desenvolvedores. A cooperação será minerada a partir do conjunto de arquivos modificados por um desenvolvedor (*commits*).

A mineração de dados referentes à comunicação e dados referentes à cooperação nos garantirá uma diversidade maior de

redes sociais em relação aos trabalhos relacionados que serão apresentados na subseção 2.2 e 2.3.

Uma vez mineradas estas informações serão geradas redes sociais, que analisadas com SNA resultarão em valores para as métricas, que serão utilizadas como entrada para o modelo de predição de falhas.

Os projetos escolhidos serão aqueles que possibilitem a identificação de *builds* que obtiveram sucesso ou falha e que utilizem *mailing lists* e gerenciadores de tarefas (*issues*).

Para a coleta de dados será construído um minerador capaz de extrair as informações dos projetos e salvá-las em um banco de dados local.

### 3.2 Construção das redes sociais

A partir dos dados minerados e armazenados no banco de dados local, será gerada uma matriz desenvolvedor versus desenvolvedor com o grau da interação capturada. O grau de interação representa a quantidade de trocas de mensagem existente entre dois desenvolvedores, ou a quantidade de vezes que eles trabalharam em um arquivo compartilhado. Este último será obtido a partir do número de *commits* realizados por dois desenvolvedores que alteraram uma mesma classe.

Para cada *build* de um projeto ASF serão criadas quatro redes sociais:

- Rede social de *mailing list*: As conversas realizadas dentro do intervalo de uma *build* serão mineradas e uma matriz com a quantidade de troca de mensagens entre os desenvolvedores será criada.
- Rede social de comunicação de tarefas: Serão identificadas as tarefas que compõe uma *build*. A partir das tarefas serão minerados todos os comentários realizados pelos desenvolvedores, formando uma nova matriz.
- Rede social de cooperação a partir dos arquivos (classes) compartilhados: Serão minerados a partir do *log* do sistema gerenciador de versão, todos os *commits* realizados pelos desenvolvedores para implementar as tarefas que compõe uma *build*.
- União das redes sociais anteriores: é a soma das matrizes anteriores, representa o grau total de comunicação versus cooperação que inclui todos os relacionamentos entre os desenvolvedores.

Pretende-se com estas redes sociais, melhorar a qualidade do conjunto de dados que será utilizado para treinamento da rede bayesiana no modelo de predição. A utilização de dados mais abrangentes e da união de todas as métricas encontradas nos trabalhos relacionados pode proporcionar uma melhora na predição, uma vez que a melhoria da performance dos modelos de predição não está associado a implementação de novos algoritmos, mas sim, da qualidade do conjunto de dados minerados ou até mesmo, de métricas que ainda não foram utilizadas para predição. [1]

Para criação das redes sociais será utilizada a ferramenta UCINet [13], que permite importar as matrizes criadas e calcular as métricas apresentadas no Quadro 1.

### 3.3 Predição de Falhas

No experimento deste estudo também serão identificadas falhas baseando-se no resultado de *builds*. Considera-se como premissa que as *builds* podem resultar em dois estados: o de falha ou de sucesso. Obtêm-se os valores das métricas de SNA para cada rede social criada a partir de uma *build* conforme subseção 3.2. os valores obtidos com as métricas de SNA serão utilizados como entrada de dados para o treino do classificador bayesiano.

Para o treinamento serão definidos quatro conjuntos de dados. O primeiro é referente às métricas extraídas das redes sociais de comunicação de *mailing list* e comentários feitas sobre as tarefas, outro referente a rede social de cooperação extraída a partir dos *commits* e um último conjunto referente à união das redes sociais.

Para avaliar a predição outro conjunto de validação será utilizado. Este conjunto ficará fora da etapa de treinamento. Para executar a predição de falhas será utilizada a ferramenta WEKA Software[15].

O WEKA (“*Waikato Environment for Knowledge Analysis*”) é uma ferramenta que implementa uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Ela oferece suporte para pré-processamento de dados, treinamento e classificação de redes bayesianas, oferecendo opções de escolha de diferentes algoritmos.

Para este trabalho serão comparados os treinamentos com proporções de 80/20 e 66/34 utilizando o algoritmo *naive bayes* que também foi utilizado nos trabalhos relacionados. A comparação das proporções de treinamento será feita porque não houve consenso nos trabalhos relacionados da proporção de *builds* que foram utilizadas para treinamento e validação, assim, pretende-se testar esta abordagem com as duas medidas.

### 3.4 Avaliação do modelo de predição

Para avaliar as predições serão utilizadas medidas de *Recall* e *Precision*. No entanto, será adicionada uma nova medida de *Accuracy*. Estas métricas definem a qualidade de dados recuperados.

*Precision* é definida como o número de dados relevantes recuperados, dividido pelo número total de dados recuperados [10], por exemplo, suponha que existam 50 *builds* que falharam realmente, e o classificador bayesiano respondeu (predisse) que 30 *builds* falharam, e apenas 20 destas *builds* realmente falharam. A *Precision* é calculada da seguinte forma: 30 *builds* que o classificador respondeu dividido pelas 20 *builds* que o classificador respondeu corretamente, resultando em um coeficiente de *Precision* de 1,5% e 66% expresso em porcentagem. *Recall* é definido como o número de dados relevantes recuperados divididos pelo número total de dados relevantes [10], por exemplo, suponha que 50 *builds* falharam realmente, e o classificador respondeu (predisse) que 38 *builds* falharam, e apenas 20 destas *builds* realmente falharam, o *Recall* é calculado da seguinte forma: 20 *builds* que falharam realmente divididos por 38 *builds* que o classificador respondeu corretamente, resultando em um coeficiente de *Recall* de 2,5% e 40% expresso em porcentagem.

Os trabalhos relacionados [16],[5],[1] utilizaram *Recall* e *Precision* para avaliar o modelo de predição proposto. Este trabalho além de utilizar *Recall* e *Precision* também utilizará a métrica de *Accuracy* para avaliação do modelo de predição. Esta métrica é medida pelo resultado da taxa de acerto positivo calculado com base nas classificações corretas e taxa de acerto

negativo calculado com base nas classificações incorretas. Para calcular a taxa de acerto positivo (TAP) e a taxa de acerto negativo (TAN) utilizam-se as seguintes formulas:

$$\text{TAP} = \text{SP} / (\text{SP} + \text{FN}); \text{TAN} = \text{FP} / (\text{FP} + \text{SN}).$$

- SP é o numero de *builds* que obtiveram sucesso, preditas corretamente;
- FN o numero de *builds* que falharam realmente;
- FP é o numero de *builds* que obtiveram falha, preditas corretamente;
- TN é o numero de *builds* que obtiveram sucesso realmente.

Ao final do experimento, será verificado, analisando os valores resultantes para *Recall*, *Precision* e *Accuracy*, se esta abordagem poderá prever com maior taxa de acerto quando um *build* terá sucesso ou falha, utilizando matrizes de comunicação e cooperação.

#### 4. Conclusão e Próximos Passos

Espera-se com este trabalho oferecer um modelo de predição mais preciso para prever falhas em projetos de software livre. O uso de mais métricas e de uma quantidade maior de informações para criação de redes sociais pode possibilitar esta melhora.

Um primeiro experimento com o projeto Geronimo da ASF esta sendo conduzido para mineração dos dados referentes a comunicação e cooperação.

Testes preliminares de criação de matrizes que representam as redes sociais já foram realizados. Os testes mostraram que é possível importar as matrizes que representam as redes sociais para a ferramenta *UCINet* para cálculo das métricas de SNA.

Os próximos passos compreendem a criação de redes sociais para um conjunto de *builds* que permita testar o uso da ferramenta Weka para treinamento e predição das *builds*.

Por fim será replicado o experimento para outros projetos da ASF para testar a validade da abordagem e assim comparar os resultados obtidos com os trabalhos relacionados.

#### 5. REFERÊNCIAS

- [1] BIÇER, Serdar; BENER, Ayse; CAGLAYAN, Bora. Defect Prediction Using Social Network Analysis on Issue Repositories. In ICSSP '11 Proceedings of the 2011 International Conference on on Software and Systems Process. New York, NY, USA, 2011.
- [2] BIRD, Christian; PATTISON, David; D'SOUZA, Raissa; FILKOV, Vladimir; DEVANBU, Premkumar. Latent Social Structures in Open Source Projects. In SE, Atlanta, GA, 2008, p.p24-36.
- [3] FUKS, Hugo; RAPOSO, Alberto; GEROSA, Marco. Do Modelo de Colaboração 3C à Engenharia de Groupware. In WEBMIDIA 2003 -Simpósio Brasileiro de Sistemas Multimídia e Web, Trilha especial de Trabalho Cooperativo Assistido por Computador, Novembro 2003, Salvador-BA, pp. 445-452
- [4] MAGDALENO, Andréa; WERNER, Cláudia; Araujo, Renata. Estudo de Ferramentas de Mineração, Visualização e Análise de Redes Sociais. Rio de Janeiro, ES-735/10, 2010.
- [5] MENEELY, Andrew; WILLIAMS, Laurie; SNIPES, Will; Osborne, Jason. Predicting failures with developer networks and social network analysis. In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (SIGSOFT '08/FSE-16). ACM, New York, NY, USA, 2008.
- [6] MENEELY, Andrew; WILLIAMS, Laurie. Socio-technical developer networks: should we trust our measurements?. In Proceeding of the 33rd international conference on Software engineering (ICSE'11). ACM, New York, NY, USA, 281-290. DOI=10.1145/1985793.1985832, 2011.
- [7] MENEELY, Andrew; WILLIAMS, Laurie; CORCORAN, Mackenzie. Improving Developer Activity Metrics with Issue Tracking Annotations. In Workshop on Emerging Trends in Software Metrics (WETSoM), Cape Town, South Africa, 2010.
- [8] MENZIES, Tim; GREENWALD, Jeremy; FRANK, Art. Data Mining Static Code Attributes to Learn Defect Predictors. In Ieee Transactions On Software Engineering, West Virginia Univ., Morgantown, WV, Jan. 2007.
- [9] NIA, Roozbeh; BIRD, Christian; DEVANBU, Premkumar; FILKOV, Vladimir. Validity of Network Analyses in Open Source Projects. In Mining Software Repositories, Cape Town, South Africa, 2010.
- [10] SARR M. Improving precision and recall using a spellchecker in a search engine. Royal Institute of Technology, Sweden, 2003.
- [11] SHIN, Yonghee; MENEELY, Andrew; WILLIAMS, Laurie; OSBORNE Jason. Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities. In IEEE Transactions in Software Engineering (TSE), vol. to appear, no. p. 2010.
- [12] STEINMACHER, Igor; CHAVES, Ana; GEROSA, Marco. Awareness Support in Global Software Development: A Systematic Review Based on the 3C Collaboration Model. In: 16th CRIWG Conference on Collaboration and Technology, 2010, Maastricht. Lecture Notes in Computer Science, 2010. v. 6257. p. 185-201
- [13] Ucinet is a social network analysis program. Disponível em <http://www.analytictech.com/ucinet/>
- [14] WASSERMAN, Stanley; FAUST, Katherine; Social Network Analysis: Methods and Applications. 1 ed. Cambridge, United Kingdom, Cambridge University Press. 1994.
- [15] Weka Software is a collection of machine learning algorithms for solving real-world data mining problems. Disponível em <http://sourceforge.net/projects/weka/>
- [16] WOLF, Timo; SCHROTER, Adrian; DAMIAN, Daniela; NGUYEN, Thanh. Predicting build failures using social network analysis on developer communication. In Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, 2009.

