

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

DIOGO VIEIRA FORTUNATO

**IDENTIFICANDO VALORES DE REFERÊNCIA PARA PREDIÇÃO DE SUCESSO
E FALHA EM *OPEN SOURCE SOFTWARE*: UM ESTUDO EMPÍRICO**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO

2011

DIOGO VIEIRA FORTUNATO

**IDENTIFICANDO VALORES DE REFERÊNCIA PARA PREDIÇÃO DE SUCESSO
E FRACASSO EM *OPEN SOURCE SOFTWARE*: UM ESTUDO EMPÍRICO**

Trabalho de Conclusão de Curso de graduação do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. MSc. Igor Scaliente Wiese

CAMPO MOURÃO

2011



Ministério da Educação
Universidade Tecnológica Federal do Paraná

Campus Campo Mourão



Coordenação do Curso Superior de Tecnologia em Sistemas para Internet

ATA DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

As **dezenove horas** do dia **vinte e cinco de novembro de dois mil e onze** foi realizada na sala F103 da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet do acadêmico **Diogo Vieira Fortunato** com o título **IDENTIFICANDO VALORES DE REFERÊNCIA PARA PREDIÇÃO DE SUCESSO E FRACASSO EM OPEN SOURCE SOFTWARE: UM ESTUDO EMPÍRICO**. Estavam presentes, além do acadêmico, os membros da banca examinadora composta pelo professor **Me. Igor Scaliante Wiese** (Orientador-Presidente), pelo professor **Dr. Reginaldo Ré** e pelo professor **Me. Gabriel Costa Silva**. Inicialmente, o aluno fez a apresentação do seu trabalho, sendo, em seguida, arguido pela banca examinadora. Após as arguições, sem a presença do acadêmico, a banca examinadora o considerou **Aprovado** na disciplina de Trabalho de Conclusão de Curso e atribuiu, em consenso, a nota _____. Este resultado foi comunicado ao acadêmico e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de **quinze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações: _____

Campo Mourão, 25 de novembro de 2011.

Prof. Dr. Reginaldo Ré

Membro

Prof. Me. Gabriel Costa Silva

Membro

Prof. Me. Igor Scaliante Wiese

Orientador

A folha de aprovação assinada encontra-se na coordenação do curso.

AGRADECIMENTOS

À Deus por acreditar em mim e tornar possível o que até mesmo eu muitas vezes duvido. “Entrega teu caminho ao Senhor, confia nele e Ele tudo fará” (Sl. 37:5). Ao meu orientador, Prof. MSc. Igor Wiese pela sua paciência e dedicação a este trabalho. Aos professores de estatística Prof. Me. Luiz Ledo Júnior e Prof. Me. Elton Celton de Oliveira pela contribuição com este trabalho. À minha família por me incentivar, ajudar e acreditar que sou capaz.

Aos meus amigos e colegas, principalmente Euclides Júnior, Kleiziane Nayara, Thays Córdova, Luiz Philipe Alves e Raul Benatti por me incentivarem e muitas vezes me alegrarem em momentos difíceis tornando esta jornada mais fácil.

RESUMO

FORTUNATO, Diogo Vieira. Identificando Valores de Referência para Predição de Sucesso/Fracasso em Open Source Software: Um Estudo Empírico. 2011. 60 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2011.

Diante de um contexto onde boa parte dos projetos de código fonte aberto são considerados falhos este trabalho investigou valores de referência para métricas encontradas na literatura e métricas do JIRA Issue Tracker da Apache para que estas métricas possam ser usadas como referência para inferir sobre o estado do projeto OSS (Open Source Software). Através do cálculo da média e desvio padrão dos valores encontrados, redes bayesianas foram treinadas para verificar se estes valores eram válidos. Este estudo ainda analisa algumas métricas propostas através da Análise de Componentes Principais (PCA) como forma de investigar o correlacionamento entre estas métricas.

Palavras-chave: Valor de Referência. Open Source Software. Apache Software Foundation. Métricas. Padrão de Comportamento.

ABSTRACT

FORTUNATO, Diogo Vieira. Identifying Reference Values for Prediction of Success / Failure to Open Source Software: An Empirical Study 2011. 60 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2011.

Faced with a context where much of the open source projects are considered failure this study investigated reference values for metrics in the literature and metrics from Apache JIRA Issue Tracker for these metrics can be used as reference to infer the state of OSS project (Open Source Software). By calculating the mean and standard deviation of the values found, Bayesian networks were trained to check if these values were valid. This study also examines some metrics proposed by Principal Component Analysis (PCA) in order to investigate the correlations between these metrics.

Keywords: Reference Value. Open Source Software. Apache Software Foundation. Metrics. Pattern of behavior.

LISTA DE TABELAS

TABELA 1 - VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE TIPO - <i>ISSUES</i> CRIADAS.....	34
TABELA 2 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE TIPO - <i>ISSUES</i> RESOLVIDAS.....	36
TABELA 3 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE <i>STATUS</i> - <i>ISSUES</i> CRIADAS.....	37
TABELA 4 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE <i>STATUS</i> - <i>ISSUES</i> RESOLVIDAS.....	38
TABELA 5 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE RESOLUÇÃO - <i>ISSUES</i> CRIADAS.....	38
TABELA 6 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE RESOLUÇÃO - <i>ISSUES</i> RESOLVIDAS.....	39
TABELA 7 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE PRIORIDADE - <i>ISSUES</i> CRIADAS.....	39
TABELA 8 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE PRIORIDADE - <i>ISSUES</i> RESOLVIDAS.....	40
TABELA 9 – RESULTADO DO TREINO DA REDE BAYESIANA PARA Q1.....	40
TABELA 10 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE TIPO PELO TOTAL - <i>ISSUES</i> CRIADAS.....	41
TABELA 11 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DAS MÉTRICAS DE RELAÇÃO DE <i>TIPO</i> - <i>ISSUES</i> CRIADAS.....	42
TABELA 12 – RESULTADO DO TREINO DA REDE BAYESIANA PARA TIPO CRIADA.....	42
TABELA 13 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE TIPO PELO TOTAL - <i>ISSUES</i> RESOLVIDAS.....	43
TABELA 14 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DAS MÉTRICAS DE RELAÇÃO DE TIPO - <i>ISSUES</i> RESOLVIDAS.....	43
TABELA 15 – RESULTADO DO TREINO DA REDE BAYESIANA PARA TIPO - <i>ISSUES</i> RESOLVIDAS.....	43
TABELA 16 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE <i>STATUS</i> PELO TOTAL - <i>ISSUES</i> CRIADAS.	44
TABELA 17 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DAS MÉTRICAS DE RELAÇÃO DE <i>STATUS</i> - <i>ISSUES</i> CRIADAS.	44
TABELA 18 – RESULTADO DO TREINO DA REDE BAYESIANA PARA VALORES DE <i>STATUS</i> - <i>ISSUES</i> CRIADAS.	45

TABELA 19 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE STATUS PELO TOTAL - <i>ISSUES</i> RESOLVIDAS.....	45
TABELA 20 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DAS MÉTRICAS DE RELAÇÃO DE <i>STATUS</i> - <i>ISSUES</i> RESOLVIDAS.....	46
TABELA 21 – RESULTADO DO TREINO DA REDE BAYESIANA PARA <i>STATUS</i> - <i>ISSUES</i> RESOLVIDAS.	46
TABELA 22 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE RESOLUÇÃO PELO TOTAL – <i>ISSUES</i> CRIADAS.....	46
TABELA 23 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DAS MÉTRICAS DE RELAÇÃO DE RESOLUÇÃO – <i>ISSUES</i> CRIADAS.....	46
TABELA 24 – RESULTADO DO TREINO DA REDE BAYESIANA PARA RESOLUÇÃO – <i>ISSUES</i> CRIADAS.....	47
TABELA 25 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE RESOLUÇÃO PELO TOTAL – <i>ISSUES</i> RESOLVIDAS.....	47
TABELA 26 – RESULTADO DO TREINO DA REDE BAYESIANA PARA MÉTRICAS DE RESOLUÇÃO.....	47
TABELA 27 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE PRIORIDADE PELO TOTAL – <i>ISSUES</i> CRIADAS.....	48
TABELA 28 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DA MÉTRICA DE RELAÇÃO DE PRIORIDADE – <i>ISSUES</i> CRIADAS.....	48
TABELA 29 – RESULTADO DO TREINO DA REDE BAYESIANA PARA PRIORIDADES – <i>ISSUES</i> CRIADAS.....	48
TABELA 30 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DAS MÉTRICAS DE PRIORIDADE PELO TOTAL – <i>ISSUES</i> RESOLVIDAS.....	49
TABELA 31 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO DA MÉTRICA DE RELAÇÃO DE PRIORIDADE – <i>ISSUES</i> RESOLVIDAS.....	49
TABELA 32 – RESULTADO DO TREINO DA REDE BAYESIANA PARA PRIORIDADE – <i>ISSUES</i> RESOLVIDAS.....	50
TABELA 33 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DA MÉTRICA DE DESENVOLVEDOR.....	50
TABELA 34 – RESULTADO DO TREINO DA REDE BAYESIANA PARA QUANTIDADE DE DESENVOLVEDOR.....	50
TABELA 35 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DA MÉTRICA DE RELAÇÃO DE DESENVOLVEDOR POR <i>ISSUES FIXED</i> – <i>ISSUES CRIADAS</i>	51
TABELA 36 – RESULTADO DO TREINO DA REDE BAYESIANA PARA MÉTRICA DE RELAÇÃO DE DESENVOLVEDOR POR <i>FIXED</i> – <i>ISSUES</i> CRIADAS.....	51
TABELA 37 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO	

DA MÉTRICA DE RELAÇÃO DE DESENVOLVEDOR POR <i>ISSUES FIXED</i> – <i>ISSUES</i> RESOLVIDAS.....	51
TABELA 38 – RESULTADO DA REDE BAYESIANA PARA MÉTRICA DE RELAÇÃO DE DESENVOLVEDOR POR <i>FIXED</i> – <i>ISSUES</i> RESOLVIDAS.....	52
TABELA 39 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DA MÉTRICA DE COMUNICAÇÃO.....	52
TABELA 40 – RESULTADO DO TREINO DA REDE BAYESIANA PARA QUANTIDADE DE MENSAGENS.....	52
TABELA 41 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DA MÉTRICA DE RELAÇÃO DE COMUNICAÇÃO POR <i>ISSUES FIXED</i> – <i>ISSUES</i> CRIADAS.....	53
TABELA 42 – RESULTADO DO TREINO DA REDE BAYESIANA PARA MÉTRICA DE RELAÇÃO DE MENSAGEM POR <i>FIXED</i> – <i>ISSUES</i> CRIADAS.....	53
TABELA 43 – VALORES DE MÍNIMO, MÁXIMO, MÉDIA, DESVIO PADRÃO DA MÉTRICA DE RELAÇÃO DE COMUNICAÇÃO POR <i>ISSUES FIXED</i> – <i>ISSUES</i> RESOLVIDAS.....	53
TABELA 44 – RESULTADO DO TREINO DA REDE BAYESIANA PARA MÉTRICA DE MENSAGEM – <i>ISSUES</i> RESOLVIDA.....	54
TABELA 45 - VALORES DE PRECISION, RECALL E F-MEASURE DAS MÉTRICAS DE Q2.....	55

SUMÁRIO

1	INTRODUÇÃO	11
2	REVISÃO BIBLIOGRÁFICA.....	13
2.2	Apache Software Foundation.....	16
2.3	Modelos de Predição Baseados em BBN (<i>Bayesian Belief Networks</i>).....	19
3	METODOLOGIA.....	22
3.1	Apache Felix Gogo.....	22
3.2	Gerônimo Miner.....	22
4	RESULTADOS.....	27
4.1	Classificação e Identificação das Métricas.....	27
4.1.2.	Q1: É Possível Estabelecer um Valor de Referência para Períodos de Sucesso e Períodos de Falha?.....	32
4.1.2.1.	Considerações dos Resultados de Q1.....	39
4.1.4.	Q2: Com Base em um Padrão de Comportamento, é Possível Encontrar Valores de Referência para as Métricas?.....	41
4.1.3.1.	Considerações dos Resultados de Q2.....	50
4.1.4.	Q3: Qual a Influência de uma Métrica Sobre as Demais Métricas para o Período de Sucesso e o Período de Falha?.....	51
4.1.4.1.	Considerações dos Resultados de Q3.....	52
5	AMEAÇAS DE VALIDAÇÃO.....	59
6	CONCLUSÃO.....	60
7	TRABALHOS FUTUROS.....	62
8	REFERÊNCIAS.....	63

1. INTRODUÇÃO

Desenvolvedores de software têm procurado participar de projetos OSS (Open Source Software) para enriquecer seu currículo, adquirir novos conhecimentos e aprender com outros desenvolvedores mais experientes (COMINO; MANENTI; PARISI, 2007). O motivo deste interesse se deve a qualidade e estabilidade de projetos de software livre como o servidor Apache, as diferentes distribuições do Linux e o navegador web Mozilla Firefox (SAMOLADAS; ANGELIS; STAMELOS, 2010). Este fenômeno tem despertado interesse de universidades e empresas uma vez que ele traz vantagens a todos os seus participantes, sejam eles com o intuito de desenvolver seus estudos ou obter alguma vantagem comercial (SAMOLADAS; ANGELIS; STAMELOS, 2010) (CAPRA et al., 2010).

Diante das possibilidades de pesquisas que podem ser realizadas com os dados existentes nos repositórios de software livre, diversas análises têm sido feitas para descobrir diferentes aspectos do desenvolvimento de projetos OSS (VEM; VERELST; MANNAERT, 2008). Dentre as quais se destacam: análise e visualização da evolução de projetos OSS (STEFHANY; MENS; GÎRBA, 2009) (GONZALES-BARAHONA; ROBLES; DUEÑAS, 2010), modelos de predição de risco de software com base na mineração de repositórios de softwares livres (WOLF et al., 2009) (HU et al., 2007) (HU et al., 2009), métricas para sucesso e fracasso de software (WU; GOH, 2009) (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001) entre outros.

Rainer e Gale (2005) verificaram em seu estudo realizado no repositório de software livre Sourceforge.net que de 50000 projetos analisados, apenas 0,9% poderiam ser considerados os “mais ativos”, aproximadamente 11,6% considerados com código de desenvolvimento ativo e apontam que a maior parte dos projetos hospedados são projetos “impulsivos” e acabam não tendo sucesso. Diversos fatores podem influenciar no sucesso ou fracasso de um projeto OSS, tais como: atratividade; número de desenvolvedores; tamanho total da comunidade; público alvo; linguagem de programação de desenvolvimento; tipo de licença; tempo gasto para correção de *bugs*; tempo para lançamento de novos *releases*, entre outros (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001) (SUBRAMANIAM, SEN, &

NELSON, 2009) (WANG, GUO, & SHI, 2007) (GONZALEZ-BARAHONA, ROBLES, & DUEÑAS, 2010) (COMINO, MANENTI, & PARISI, 2007).

A literatura indica diferentes definições de sucesso e fracasso para projetos de software. Cada definição normalmente esta associada a fatores observados por cada pesquisador. Estes trabalhos, normalmente transformam um fator observado em uma única métrica que não considera a evolução do seu valor ao longo da evolução do desenvolvimento do projeto.

Diante deste contexto, o objetivo deste estudo é realizar um estudo empírico que observará o projeto Felix Gogo com o objetivo de obter valores de referência de um projeto OSS para que o mantenedor, lider de projeto e/ou desenvolvedores envolvidos possam saber informações sobre o estado do projeto. A escolha do Felix Gogo se deu pelo fato dele possuir o histórico de builds deste o início da geração destas, além de contar com fases bem definidas de sucesso e fracasso.

Para este estudo forão utilizadas métricas já existentes na literatura, tais como: quantidade de desenvolvedores; quantidade de troca de mensagem; quantidade de *bugs* identificados e que foram apontadas por outros autores como influenciadores de sucesso ou fracasso em um projeto de software métricas estas que possam ser extraídas do repositório Apache, além de outras métricas do repositório que foram coletadas. Os valores de referência serão utilizados como entrada para o modelo de predição baseado em bayes.

Este trabalho também classificou as métricas encontradas na literatura como endógenas ou exógenas. Para este trabalho, uma métrica endógena é todo o trabalho percebido por um desenvolvedor (BEECHER, CAPILUPPI, & BOLDYREFF, 2009), como por exemplo, um *commit* realizado, uma tarefa recebida, uma comunicação efetuada entre dois ou mais desenvolvedores. Já uma métrica exógena (BEECHER, CAPILUPPI, & BOLDYREFF, 2009) indica os fatores que são externos ao projeto, mas que também podem influenciar as escolhas de um desenvolvedor, tais como, a linguagem de programação, a licença do *software*, o domínio do problema.

O conceito de sucesso/fracasso deste trabalho está diretamente relacionado com a descoberta dos valores de referências das métricas identificadas na literatura.

Uma vez que foi possível utilizar as métricas para obter valores de referência, observar instantes diferentes do desenvolvimento em que o software teve sucesso e/ou falhou, ou seja, observar estes valores de referência no instante em que um *build* falha ou obtêm sucesso, foi possível prever como o desenvolvimento se dará.

Este trabalho está organizado da seguinte forma: O capítulo 2 apresenta os conceitos e trabalhos relacionados, bem como lista das métricas encontradas na literatura. Também será feita uma descrição do funcionamento da ASF (Apache Software Foundation) e a apresentação da ferramenta de classificação bayesiana utilizada para o modelo de predição. O capítulo 3 descreve a metodologia de coleta de dados para a realização do estudo empírico, bem como, as questões de pesquisa que guiarão o estudo. Já o capítulo 4 apresenta os resultados obtidos durante a realização desta pesquisa. Por fim, são apresentadas as ameaças de validação, a conclusão e os trabalhos futuros que podem ser realizados a partir deste estudo.

2. REVISÃO BIBLIOGRÁFICA

Fornecer um meio de inferir sucesso ou fracasso de um software é cada vez mais necessário num contexto onde boa parte dos projetos são considerados falhos (BEECHER, CAPILUPPI, & BOLDYREFF, 2009). Saber onde pode estar o principal motivo que leva um OSS a falhar, ou promover meios de evitar falhas pode ser importante na tomada de decisão para gerentes e desenvolvedores. No entanto, uma revisão na literatura apontou que pode haver divergência sobre o conceito de sucesso, e como obter uma medida de referência.

A definição de sucesso ou fracasso é variável, porque o contexto de estudo também pode ser. Assim, estudos têm se interessado pelos dados que repositórios de projetos OSS podem oferecer, procurando estabelecer formas de medir sucesso e fracasso.

Segundo Beecher, Capiluppi e Boldyreff (2009) forças exógenas podem atuar sobre o projeto de acordo com o ambiente no qual ele está hospedado influenciando sobre seu sucesso. Estas forças exógenas atuam sobre métricas específicas. Portanto, métricas exógenas, são métricas que estão fora do controle dos desenvolvedores ou do mantenedor do projeto, mas que influenciam no seu trabalho diretamente. Ao contrário das exógenas, as forças endógenas atuam diretamente sobre os valores intrinsecamente relacionados ao código-fonte e ao desenvolvimento.

O quadro abaixo mostra as métricas encontradas na literatura e os autores que a elas fazem referências e o exemplo de possíveis valores para cada métrica:

Autor(es)	Métrica	Exemplo de Valores
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO,	Tipo de licença	Altamente restritiva; restritiva; não restritiva.

MANENTI, & PARISI, 2007)		
(MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (COMINO, MANENTI, & PARISI, 2007); (SANTOS JR., PEARSON, & KON, 2010)	Público Alvo	Desenvolvedores; usuários finais.
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Tipo de Projeto	Jogos; Office; Internet; Comunicação.
(MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007); (SANTOS JR., PEARSON, & KON, 2010); (RAINER & GALE, 2005)	Status do Desenvolvimento	Planejamento; Alfa; Beta.
(XU, ZHENG, & GUO, 2006); (HU, ZHANG, SUN, LIU, & DU, 2009); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (WANG, GUO, & SHI, 2007); (GIANG, KANG, & BAE, 2010)	Métricas de código fonte	LOC; KLOC; DIT.
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001)	Tamanho da comunidade de usuários ou número de downloads	30000; 80000; 200000.
(SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Sistema Operacional	Windows; Linux; MacOs.
(SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Linguagem de Programação	C#; Java; .Net; PHP.
(COMINO, MANENTI, & PARISI, 2007); (BEECHER, CAPILUPPI, & BOLDYREFF, 2009); (HU, ZHANG, SUN, LIU, & DU, 2009); (RAINER & GALE,	Número de Desenvolvedores	7; 10; 15.

2005)		
(COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Quantidade de <i>bugs</i> identificados	20; 50; 70.
(WANG, GUO, & SHI, 2007); (COMINO, MANENTI, & PARISI, 2007)	Quantidade de funcionalidades adicionais requeridas	10; 30; 55.
(WOLF, SCHROTER, DAMIAN, & NGUYEN, 2009); (STEPHANY, MENS, & GÎRBA, 2009); (BROOKS, 1975); (RAINER & GALE, 2005); (DEMARCO & LISTER, 1999)	Quantidade de trocas de mensagens	10; 20; 50; 100.
(WANG, GUO, & SHI, 2007)	Quantidade de <i>bugs</i> corrigidos	12; 25; 40.

Quadro 1 – Métricas encontradas na literatura.

Após a identificação das métricas, notou-se que algumas não possuíam uma fórmula de cálculo que pudesse ser utilizada para obter seu valor de referência. Estas métricas normalmente estavam classificadas como um fator exógeno, tais como: tipo de licença; público alvo; tipo de projeto; status do desenvolvimento. Para medir métricas assim alguns autores utilizam variáveis qualitativas para seus cálculos. Assim, Santos Jr., Pearson e Kon (2010) construíram um modelo que eles chamam de modelo de atratividade. Para o cálculo neste modelo eles utilizam variáveis qualitativas em um modelo estatístico chamado de SEM (*Structural Equation Modeling*).

Além disto, vislumbra-se a análise de outras métricas, que embora não encontradas na literatura revisada, podem indicar sucesso/fracasso, pois elas estão diretamente relacionadas ao desenvolvimento e podem ser obtidas do sistema de gerenciamento de atividade (*issue tracker – JIRA*) dos projetos da Apache. Entre

elas, destacam-se o status da atividade, quantidade de atividades com prioridade maior que foram solucionadas, quantidade de atividades solucionadas, entre outras.

2.2. Apache Software Foundation.

Um projeto que queira fazer parte da Apache Foundation deverá primeiramente ser aceito na incubadora Apache (<http://incubator.apache.org>) onde este projeto receberá orientação. Periodicamente este projeto será avaliado para determinar se deve haver sua continuidade ou até mesmo promoção a projeto oficial da Apache. (APACHE FOUNDATION, 2011).

Dentro da Fundação Apache existe o Jenkins, um servidor de integração contínua que permite a compilação e execução de testes de acordo com uma configuração pré-estabelecida. Uma *build* corresponde a um conjunto de classes especificadas, codificadas, testadas e empacotadas. Também é importante mencionar que uma *build* pode se encontrar em dois estados: sucesso ou falha. Para cada estado foram coletados os valores de referências para as métricas. Mais detalhes podem ser encontrados no Capítulo 3 que trata da metodologia de mineração dos dados e obtenção dos valores de referência. Portanto, para nosso estudo foram apenas considerados os projetos que possuíam *builds*.

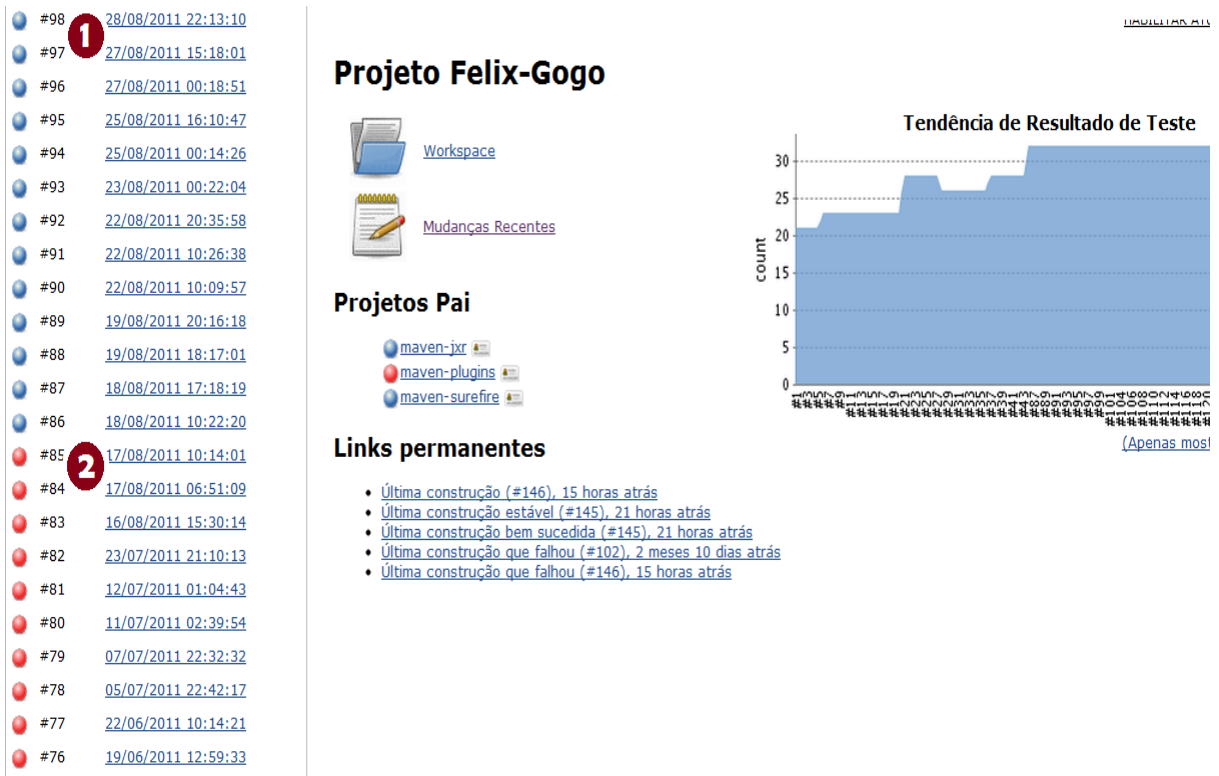


Figura 1 - Página Web do projeto Felix Gogo no Jenkins.

Do dia 25 de Julho de 2009, quando se deu início a geração dos *builds*, até o dia 15 de Julho de 2010, o sistema havia gerado 43 *builds* sendo todos eles classificados com sucesso. A partir desta data todos os *builds* gerados, que totalizam 41, até o dia 30 de Agosto de 2011 foram classificados como falhas. Ao primeiro período dos *builds* foi denominado período de sucesso, e ao segundo, período de fracasso, como forma de facilitar a compreensão. Na figura acima as legendas 1 e 2 indicam partes destes intervalos.

Ainda dentro da estrutura da Apache Software Foundation temos o Jira *Issue Tracker* - Sistema de gerenciamento de *issues* fornecido pela Apache para seus projetos. Neste sistema um usuário cadastrado pode criar, procurar, comentar e votar *issues*, que é a definição dada a qualquer atividade ou trabalho a ser realizado.

A maioria dessas *issues* são atividades de desenvolvimento de código, seja na forma da criação de novas funcionalidades (e.g. *new feature*) ou aperfeiçoamento de algo já desenvolvido (e.g. *improvement*).

Ao se criar uma *issue* o usuário deve fornecer dados que caracterizam aquela atividade. As informações pedidas no cadastro de uma *issue* são:

Tipo: Define o tipo da *issue*.

Improvement – Uma melhoria ou aperfeiçoamento de um recurso ou tarefa existente.

New Feature – Uma nova funcionalidade do produto que ainda precisa ser desenvolvida.

Task – Uma tarefa que precisa ser feita.

Test – Uma nova unidade, integração ou teste do sistema.

Wish – Item geral da lista de desejo.

Nível de Prioridade: Indica o nível de importância da *issue*.

Blocker – Bloqueia o desenvolvimento ou testes e a produção não pode ocorrer.

Critical – Falhas, perdas de dados e perdas de memória grave.

Major – Grande perda de função.

Minor – Menor perda de função ou outro problema em que a solução é fácil.

Trivial – Problemas corriqueiros como palavras com erro de ortografia ou texto desalinhado.

Após a etapa de criação da *issue*, uma página é gerada com mais informações como status e resolução, se dividindo estes nos seguintes valores. (JIRA)

Status: Cada *issue* possui um status que indica o estágio da *issue*.

Open – A *issue* está aberta e pronta para um desenvolvedor trabalhar nela.

In Progress – Esta *issue* está sendo trabalhada ativamente por um desenvolvedor.

Reopened – Esta *issue* já foi resolvida, mas a resolução foi considerada incorreta.

Resolved – A *issue* foi resolvida e está esperando a aprovação de quem a reportou.

Closed – A *issue* foi considerada fechada. A resolução foi correta. *Issues* que não estão fechadas podem ser reabertas.

Resolução: Uma *issue* pode ser resolvida de diversas maneiras.

Fixed – A correção para esta *issue* está checada e testada.

Won't Fix – A o problema descrito nesta *issue* nunca será resolvido.

Duplicate – O problema é uma duplicação de uma *issue* existente.

Invalid – O problema não é valido e não pode ser resolvido.

Incomplete – O problema não está completamente descrito.

Cannot Reproduce – Todas as alternativas de reproduzir esta *issue* falhou ou as informações foram insuficientes ou ainda a leitura do código não forneceu pistas do porque este comportamento estaria ocorrendo.

Later – A resolução desta *issue* será deixada para outro momento.

Not A Problem – O relatado na *issue* não é realmente um problema.

Unresolved – *Issue* não resolvida.

As classificações possíveis de uma *issue* acima listadas foram as encontradas no projeto Felix Gogo, no entanto, outras classificações podem ser dadas a uma *issue* e mais detalhes podem ser vistos em (JIRA).

2.4 Modelos de Predição baseado em BBN (Bayesian Belief Networks)

Redes bayesianas BBN são modelos de representação de causa e efeito de uma grande variedade de domínios, também conhecidas como redes causais, eficazes na modelagem de uma situação onde não todas as informações são conhecidas e são utilizadas para facilitar a tarefa de predição. Baseados no teorema de bayes estas redes realizam a predição com base no aprendizado que elas realizam através dos dados nelas inseridos. Quanto maior a padronização dos dados maior será sua precisão ao realizar a predições (CHARLES RIVER ANALYTICS, 2004).

Modelos de predição utilizando redes bayesianas são criados e utilizados por alguns autores para a predição. Boness e Harrison (2002) criaram um modelo para predição de riscos baseado em dados que podem ser obtidos com os gerentes do projeto e também com dados que possam ser extraídos do projeto. Utilizando métricas de código fonte, e comparado a outras técnicas: *Decision Tree*, *Nearest Neighbor* e redes bayesianas, redes bayesianas se mostraram mais eficientes para predição de risco (XU, ZHENG, & GUO, 2006).

Procurando predizer falhas com base na comunicação, Wolf *et al.* (2009) construíram um modelo bayesiano de predição de falhas e testam se a combinação de métricas de estrutura de comunicação pode predizer falhas de integração. Com este modelo conseguiram indicar se haverá falhas baseados na estrutura de comunicação atual do time de desenvolvimento.

O WEKA (*Waikato Environment for Knowledge Analysis*) é um software que contém a implementação de vários métodos de classificação de dados e aprendizagem de máquina, entre elas filtro de bayes. Com o carregamento do software com um arquivo texto no formato .arff é possível realizar o treino da rede bayesiana (SANTOS, 2002). Este arquivo possui uma seção de declaração de dados e outra com o conjunto de valores. A figura abaixo mostra o exemplo de um arquivo deste.

```

1 @relation weather
2
3 @attribute outlook {sunny, overcast, rainy}
4 @attribute temperature real
5 @attribute humidity real
6 @attribute windy {TRUE, FALSE}
7 @attribute play {yes, no}
8
9 @data
10 sunny,85,85,FALSE,no
11 sunny,80,90,TRUE,no
12 overcast,83,86,FALSE,yes
13 rainy,70,96,FALSE,yes
14 rainy,68,80,FALSE,yes
15 rainy,65,70,TRUE,no
16 overcast,64,65,TRUE,yes
17 sunny,72,95,FALSE,no
18 sunny,69,70,FALSE,yes
19 rainy,75,80,FALSE,yes
20 sunny,75,70,TRUE,yes
21 overcast,72,90,TRUE,yes
22 overcast,81,75,FALSE,yes
23 rainy,71,91,TRUE,no

```

Figura 4 – Exemplo de arquivo .arff

Após carregar o .arff no WEKA seleciona-se o filtro bayesiano que utilizará 66% dos dados para o treinamento da rede e os outros 34% para predição. Deste treino obtém-se 3 importantes resultados, sendo eles (BOUCKAERT, ET. AL, 2010) (BIÇER ET, AL, 2011):

Precision: Proporção de exemplos classificados de uma classe x dentre todos os que realmente são desta classe.

Recall: Proporção de exemplos que foram classificados como classe x dentre todos desta classe.

F-measure: É uma combinação de *precision* e *recall* sendo: $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$. Utilizado para comparação de modelo pois indica a qualidade do modelo, isto é, o quão equilibrado está o modelo.

Portanto, estes resultados são os mais importantes do treino bayesiano.

3. METODOLOGIA

Os conceitos e métricas identificadas na revisão bibliográfica serão utilizados para a validação das questões de pesquisa deste trabalho. As métricas serão aplicadas ao projeto OSS armazenado no repositório da Apache Foundation.

3.1 Apache Feliz Gogo

O projeto escolhido foi o Apache Felix Gogo (<http://felix.apache.org/site/apache-felix-gogo.html>). Subprojeto do Apache Felix (<http://felix.apache.org/site/index.html>), Gogo é um *framework* sob a licença Apache que implementa a plataforma de serviços OSGi R4.OSGi (*Open Service Gateway Initiative*) é uma plataforma de serviços para o desenvolvimento de aplicativos na linguagem Java. Dentre os projetos que utilizam OSGi estão a plataforma de desenvolvimento Eclipse (<http://www.eclipse.org>) e o servidor de aplicações JEE (*Java Enterprise Edition*) GlassFish (<http://glassfish.java.net>). Este subprojeto se consiste em um processador de núcleo (*runtime*) de comandos, uma interface textual (*shell*) que interage com o processador e a implementação de comandos básicos (*command*). Felix Gogo tem início de suas atividades no Jira em 24 de Julho de 2005 e seus testes no Jenkins e inicia em 25 de Julho de 2009.

O motivo de escolha deste projeto foi por ser o único projeto que continha em sua página do Jenkins (<https://builds.apache.org/view/A-F/view/Felix/job/Felix-Gogo>) o histórico dos builds desde o início, e períodos longos de sucesso e fracasso, o que possibilitaria um estudo mais consistente.

3.2 Gerônimo Miner

Para a mineração da página do projeto no JIRA foi utilizado um minerador construído em Java. Gerônimo Miner, que recebeu este nome por ter sido

primeiramente desenvolvido para a mineração de um projeto assim denominado na ASF (Apache Software Foundation) é o resultado de esforços de um grupo de pesquisa da UTFPR (Universidade Tecnológica Federal do Paraná) *campus* Campo Mourão. Ele basicamente se consiste em um analisador (*parser*) do código HTML da página do JIRA (<https://issues.apache.org/jira/browse/FELIX-1>) que extrai os dados das issues armazenando-os em uma base de dados.

A figura 2 exibe a página web do projeto de onde o minerador extrai os dados. Existem alguns agregadores de informações importantes, tais como os detalhes de uma tarefa que indicam seu tipo, sua prioridade, seu status, resolução e quais versões e componentes ela afeta. As pessoas indicam para quem a tarefa esta associada e quem deve ser reportado quando ocorre uma alteração. As datas de movimentação da *issue*, descrição e comentários também são mineradas. As legendas de 1 a 9 indicam estes dados minerados.

The Apache Software Foundation
<http://www.apache.org/>

Diogo Fortunato Quick Search

Dashboards Projects Issues Agile Create Issue

Felix / FELIX-1
Resolve OSGi licensing concerns prior to Incubator graduation

Comment Planning Board Task Board More Actions Views

Details

Type: 1 Task	Status: 3 Closed
Priority: 2 Blocker	Resolution: 4 Fixed
Affects Version/s: 5 None	Fix Version/s: None
Component/s: None	
Labels: None	

People

Assignee: 6 Enrique Rodriguez
Reporter: 7 Enrique Rodriguez
Vote (0) Watch (0)

Dates

Created: 8 25/Jul/05 04:36
Updated: 17/Mar/06 01:06
Resolved: 9 02/Dec/05 01:13

Description

We must resolve OSGi licensing issues prior to Incubator graduation. The two main items are:

1. OSGi Service Platform R4 specification sources ("companion code") must be released under a license that clearly allows free distribution and implementation.
2. OSGi Service Platform R4 specification document ("specification") must have a license that clearly allows free distribution and implementation.

Both items will be resolved in the R4 release. In particular, the companion code will be released under the EPL.

The Eclipse Foundation has been through this issue:
https://bugs.eclipse.org/bugs/show_bug.cgi?id=90052

Figura 2 - Página do projeto Felix Gogo no JIRA.

A Figura 3 mostra as tabelas do banco de dados onde estes dados são armazenados para posterior análise.

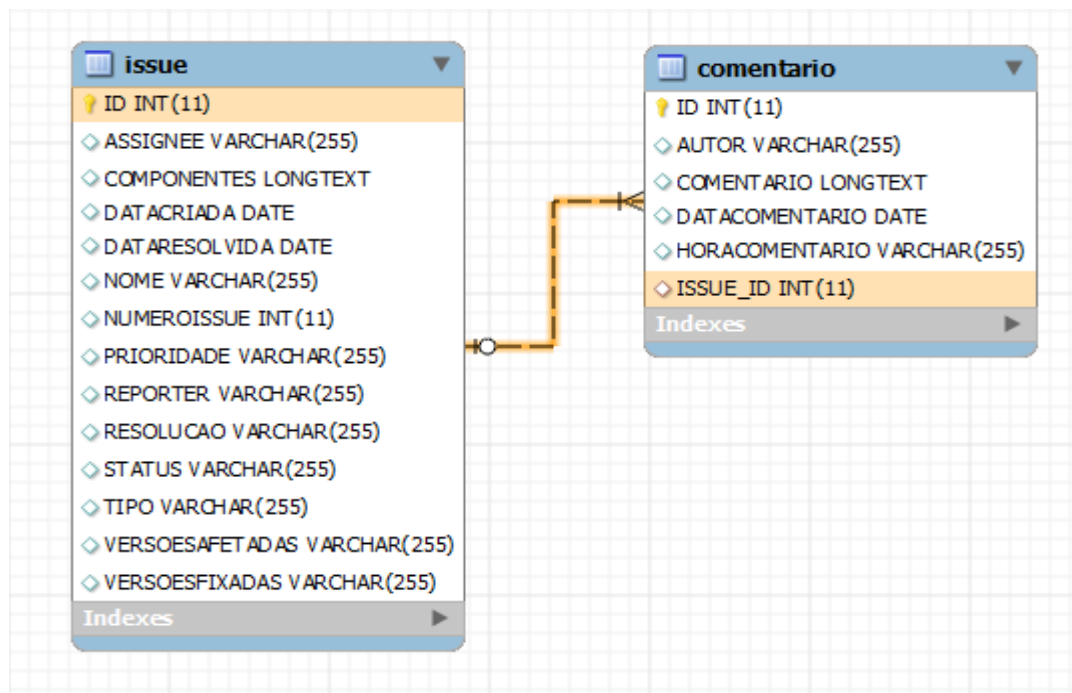


Figura 3 - Tabelas do banco de dados.

Três questões de pesquisa guiaram o estudo empírico:

Q1: É possível estabelecer um valor de referência para períodos *de sucesso* e períodos de falha?

Q2: Com base em um padrão de comportamento, é possível encontrar valores de referência para as métricas?

Q3: Qual a influência de uma métrica sobre as demais para o período de sucesso e o período de falha?

Para responder as questões de pesquisa primeiramente foi realizada uma coleta de dados no repositório de projeto da Apache (APACHE FOUNDATION, 2011). Para esta coleta de dados o minerador Gerônimo *Miner* foi utilizado para capturar as informações necessárias e extrair as métricas apresentadas no Capítulo 2. A Figura 5 exemplifica este processo.

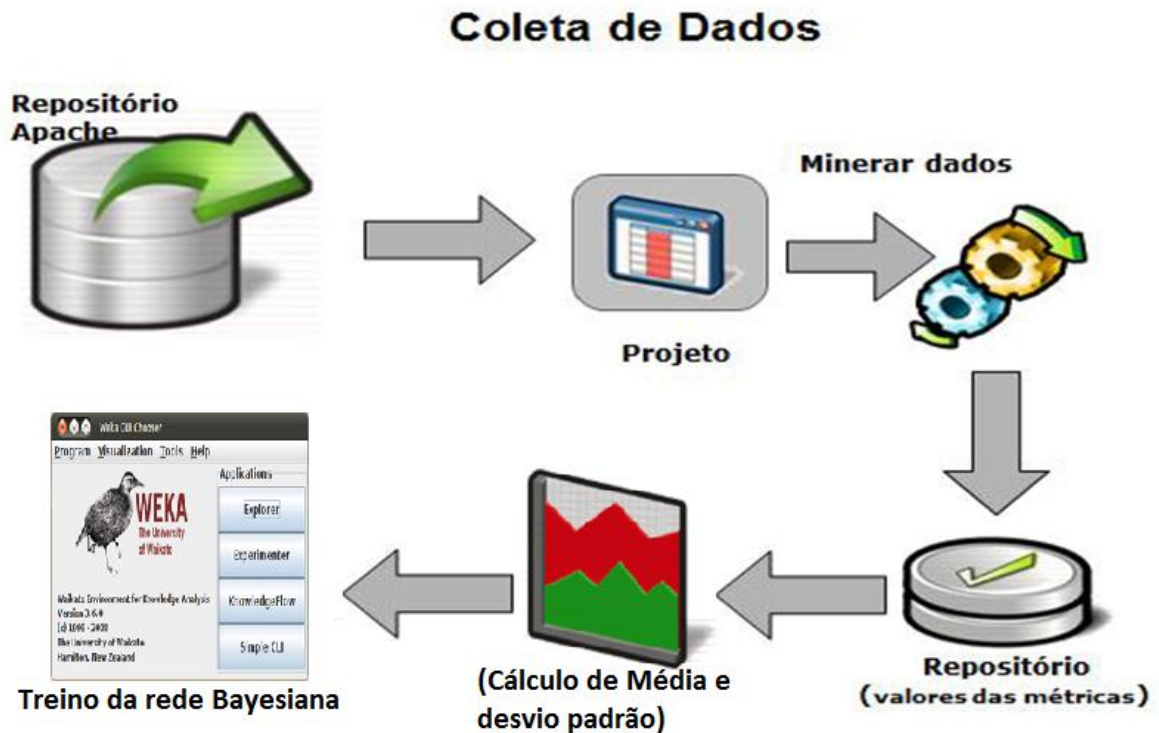


Figura 5 - Fase de coleta e mineração de dados

Para responder Q1, os valores coletados dos subperíodos de 30 dias foram usados para treinar uma rede bayesiana procurando assim validar os valores de referência propostos para cada período (i.e., período de sucesso e falha).

Segundo Ferreira (2011) para identificação dos valores de referência podem ser utilizados métodos estatísticos como: média e desvio-padrão, e através da rede bayesiana estes valores de média e desvio padrão podem ser validados. Uma vez identificados os valores de referência, estes foram utilizados para treinamento da rede bayesiana. Conforme mencionado na seção 2.3, foi utilizada a abordagem bayesiana utilizando a ferramenta WEKA que automatiza o processo de classificação/filtragem e treinamento da rede bayesiana. A Figura 6 apresenta este cenário.

Foi utilizada a classificação que a Apache dá aos *builds* do projeto em sucesso e falha, e então foi calculado a média e o desvio padrão para as métricas identificando os valores de referência encontrado para cada sub-período, sempre levando em consideração o estado do *build*.

Para responder Q1, foram extraídos os valores dos indicadores do projeto a cada 30 dias e calculado o valor médio e desvio padrão os valores de referência das

métricas propostas no intervalo de tempo existente entre os *builds* de um projeto. Através da média e desvio padrão podemos verificar o comportamento do projeto de acordo com o resultado dos *builds*, e para validar estes resultados uma rede bayesiana foi treinada para validar os resultados obtidos.

Para determinar se os valores de predição são válidos foram utilizadas medidas de *precision*, *recall* e *f-measure*, onde *precision* corresponde à precisão do modelo, *recall* indicará a quantidade de casos positivos que foram capturados em relação aos casos testados e F-measure é o indicativo do desempenho do modelo de predição (BOUCKAERT, ET. AL, 2010) (FLATIRONS, INC., 2006).

Para responder Q2 foram testadas a relação entre as métricas apresentadas, calculando para elas a média e desvio padrão, validadas pela rede bayesiana uma vez que os valores com os quais a rede foi treinada, foram utilizados para calculo da média e desvio padrão. (SANTOS, 2002) (FERREIRA, 2011)

Por fim, em Q3, os valores das métricas colhidos em diferentes *builds* do projeto foram comparados entre eles utilizando a Análise PCA (ARAUJO, 2009). Esta análise permite a normalização dos dados e um comparativo para verificar o grau de influência que uma variável possui sobre um conjunto de amostras. As amostras são os diferentes subperíodos de um projeto.

Assim, em Q3, foi possível descobrir quais métricas influenciaram as amostras do projeto. Assim, em Q3 foi comparado o resultado da análise destas métricas ao longo do seu ciclo de vida do projeto e verificar como os valores de referência das métricas influenciam o desenvolvimento do projeto.

4. RESULTADOS

Para responder as três questões de pesquisa, prioritariamente foi necessária a mineração da página do projeto Felix Gogo do JIRA. Esta mineração ocorreu no dia 10 de outubro de 2011. Nesta data havia 3175 *issues* criadas, no entanto, os *builds* apenas começaram a ser geradas no dia 25 de Julho de 2009, portanto, é a partir desta data que a mineração dos dados nos interessa, isto reduziu a quantidade de *issues* para 1664.

Como forma de normalizar e padronizar os dados (FERREIRA, 2011) a serem analisados, cada período, sucesso e falha, foi dividido em subperíodos menores de 30 dias. Assim, a cada 30 dias os valores dos dados das *issues* foram agrupados .

Outro cuidado na análise foi levar em consideração que havia dois tipos de *issues* diferentes a se considerar. As *issues* criadas durante cada período de 30 dias e as *issues* resolvidas durante o mesmo período. Este cuidado foi tomado, pois embora uma *issue* possa não ter sido criada naquele período, ela pode ter sido resolvida, o que poderia afetar o estado final – sucesso ou falha - do *build*.

Portanto deste trabalho foram obtidos 24 tabelas sendo 12 delas referentes às atividades criadas naquele período e outras 12 das atividades resolvidas. Isto para cada dado proposto a ser analisado, sendo estes: tipo, nível de prioridade, status, resolução, quantidade de desenvolvedores e quantidade de comentários, totalizando 72 tabelas para o cada período – sucesso e fracasso.

4.1 Classificação e identificação das métricas.

Prioritariamente foi realizada a classificação das métricas encontradas na literatura em endógenas e exógenas é mostrada no quadro abaixo (quadro 1).

Autores	Métrica	Fonte	Endógena	Exógena
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007)	Tipo de licença	Informação extraída do repositório.		x
(MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (COMINO, MANENTI, & PARISI, 2007); (SANTOS JR., PEARSON, & KON, 2010); (RAINER & GALE, 2005)	Público alvo	Informação extraída do repositório.		x
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Tipo de projeto	Informação extraída do repositório.		x
(MEIRELLES, JR., KON, TERCEIRO, &	Status do desenvolvimento	Informação extraída do		x

CHAVEZ, 2001); (SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007); (SANTOS JR., PEARSON, & KON, 2010); (RAINER & GALE, 2005)		repositório.		
(XU, ZHENG, & GUO, 2006); (HU, ZHANG, SUN, LIU, & DU, 2009); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001); (WANG, GUO, & SHI, 2007); (GIANG, KANG, & BAE, 2010)	Métricas de código-fonte.	Informação extraída do repositório.	x	
(SANTOS JR., PEARSON, & KON, 2010); (MEIRELLES, JR., KON, TERCEIRO, & CHAVEZ, 2001)	Tamanho da comunidade de usuários/Número de downloads.	Informação extraída do repositório.		x
(SUBRAMANIAM, SEN, & NELSON, 2009); (COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Sistema Operacional	Informação extraída do repositório.		x
(SUBRAMANIAM, SEN, & NELSON, 2009);	Linguagem de Programação	Informação extraída do		x

(COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)		repositório.		
(COMINO, MANENTI, & PARISI, 2007); (BEECHER, CAPILUPPI, & BOLDYREFF, 2009); (HU, ZHANG, SUN, LIU, & DU, 2009); (RAINER & GALE, 2005)	Número de desenvolvedores	Informação extraída do repositório.	x	
(COMINO, MANENTI, & PARISI, 2007); (RAINER & GALE, 2005)	Número de <i>bugs</i> identificados	Informação extraída do repositório.		x
(WANG, GUO, & SHI, 2007); (COMINO, MANENTI, & PARISI, 2007)	Número de funcionalidades adicionais requeridas	Informação extraída do repositório.		x
(WOLF, SCHROTER, DAMIAN, & NGUYEN, 2009); (STEPHANY, MENS, & GÎRBA, 2009); (BROOKS, 1975); (RAINER & GALE, 2005); (DEMARCO & LISTER, 1999)	Número de troca de mensagens	Informação extraída do repositório.	x	
(WANG, GUO, & SHI, 2007)	Correção de <i>bugs</i>	Informação extraída do	x	

		repositório.		
--	--	--------------	--	--

Quadro 1 - Métricas de sucesso/fracasso classificadas de acordo com a definição de Beecher et al. (2009) para endógena e exógena.

Das métricas da literatura classificadas no Quadro 1 nem todas puderam ser mineradas no repositório Apache, ou necessitam de mais de um projeto para serem estudadas. O quadro abaixo mostra quais métricas possuem sua correspondência na Apache e puderam ser estudadas.

Literatura	Apache
Número de desenvolvedores	Quantidade de <i>assignees</i> .
Número de <i>bugs</i> identificados	Quantidade de <i>issues</i> de <i>bug</i> criada.
Número de funcionalidades adicionais requeridas	Informação extraída do repositório.
Número de troca de mensagens	Quantidade de comentários por <i>issue</i> .
Correção de <i>bugs</i>	Quantidade de <i>issues</i> do tipo <i>bug</i> resolvidas

Quadro 2 – Relação das métricas na Apache

Além disto, outras métricas foram identificadas no repositório Apache e puderam ser aplicadas neste trabalho para a busca de valores de referência. Essas métricas estão relacionadas a fatores como a comunicação, desenvolvedor e *issue*. O quadro abaixo mostra quais métricas serão estudadas para cada um desses fatores, e em qual questão será aplicada.

Fator	Métrica	Questão
<i>Issue</i>	Valor das <i>issues</i> criadas e resolvidas de tipo, <i>status</i> , resolução e prioridade.	Q1
	Relação do tipo das <i>issues</i> criadas e resolvidas pelo total de <i>issues</i> .	Q2, Q3

Desenvolvedor	Quantidade de desenvolvedor por período.	Q2
	Relação da quantidade de desenvolvedor pela quantidade de issues com <i>status fixed</i> .	Q2
Comunicação	Quantidade de mensagens por período.	Q2
	Relação da quantidade de mensagens do período pela quantidade de issues com <i>status fixed</i> .	Q2

Quadro 3 – Métricas que serão calculadas

As métricas dos fatores de desenvolvedor e comunicação foram relacionadas ao valor de métricas como *fixed* pois esta é a classificação dada a uma *issue* resolvida, testada e aceita, ou seja, ela está diretamente relacionada com sucesso.

4.1.2. Q1: É possível estabelecer um valor de referência para períodos de sucesso e períodos de falha?

Para responder esta questão os dados referentes ao período de sucesso e ao período de falha foram analisados. Cada métrica proposta a ser estudada foi isolada e então foram extraídos do repositório seus valores a cada 30 dias, resultando em 24 meses, uma vez que cada período teve 12 meses.

Um dos principais comportamentos identificados foi que o período de falha teve um índice de atividade bem inferior ao período de sucesso.

Os gráficos 1, 2 e 3 mostram no decorrer de cada período o número de *issues* criadas para os tipos *improvement*, *bug* e *new feature*. Apesar de haver outros valores possíveis para tipo de *issue*, estes três em específico demonstraram maior ocorrência no projeto.

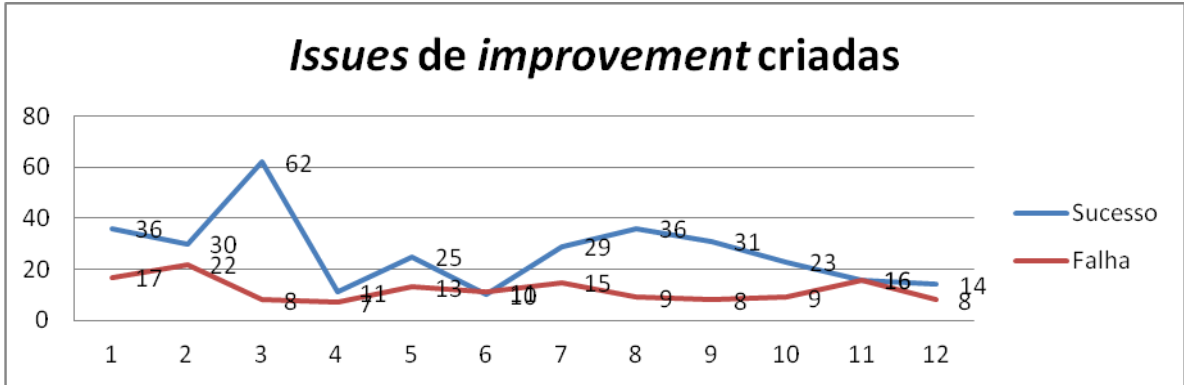


Gráfico 1 – Issues de *improvement* criadas.

Podemos visualizar no gráfico acima que a quantidade de *issues* de *improvement* foi bem superior no período de sucesso.

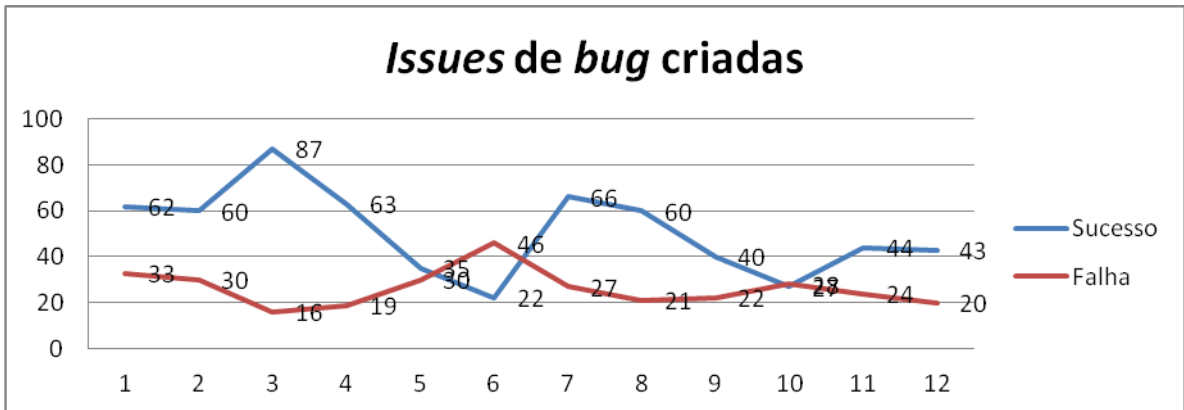


Gráfico 2 – Issues de *bug* criadas.

O Mesmo se pode verificar na quantidade de *issues* de *bug* (gráfico 2) que se na maior parte dos 12 meses superior ao período de falha.

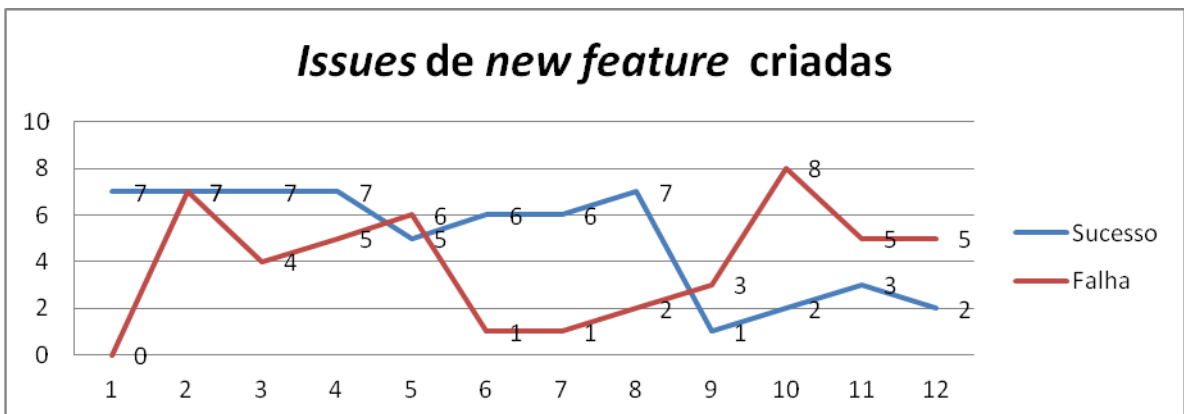


Gráfico 3 – Issues de *new feature* criadas.

A quantidade de *issues* de *new feature* embora tenha se mostrado instável, variando seus valores, ainda assim manteve-se quase 80% do tempo com valores maiores durante o período de sucesso.

Para melhor compreender os valores obtidos para cada tipo de tipo de *issue* a Tabela 1 mostra os valores mínimos, médios, máximos e desvio padrão encontrados para o período de sucesso e o período de falha. A média do período de falha é quase sempre inferior em mais de 50% com relação ao período de sucesso.

Tabela 1 - Valores de mínimo, máximo, média, desvio padrão das métricas de tipo - *issues* criadas.

Tipo das <i>issues</i> criadas	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Improvement</i>	10	7	62	22	26,917	11,917	14,368	4,699
<i>Bug</i>	22	16	87	46	50,750	26,333	18,719	8,060
<i>Task</i>	0	0	9	6	4,167	1,167	2,368	1,899
<i>Subtask</i>	0	0	4	5	1,250	0,500	1,485	1,446
<i>New Feature</i>	1	0	7	8	5,000	3,917	2,335	2,539
<i>Wish</i>	0	0	2	1	0,667	0,083	0,651	0,289
<i>Test</i>	0	0	1	1	0,083	0,083	0,289	0,289

O mesmo comportamento é repetido com a quantidade de tipo de *issues* resolvidas durante cada período.

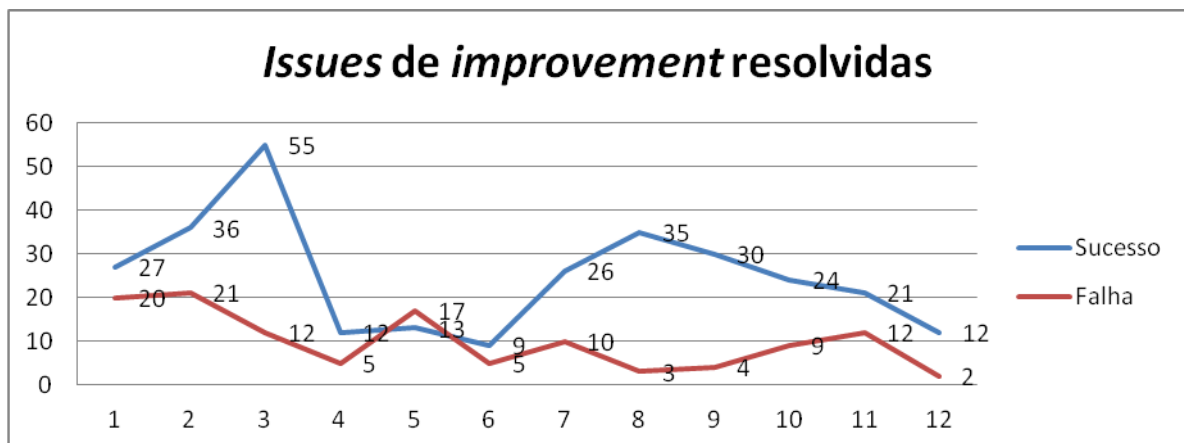


Gráfico 4 – *Issues* de *improvement* resolvidas.

Podemos visualizar no gráfico 4 que a quantidade de *issues* de *improvement* resolvidas no período de sucesso foi superior ao de falha.

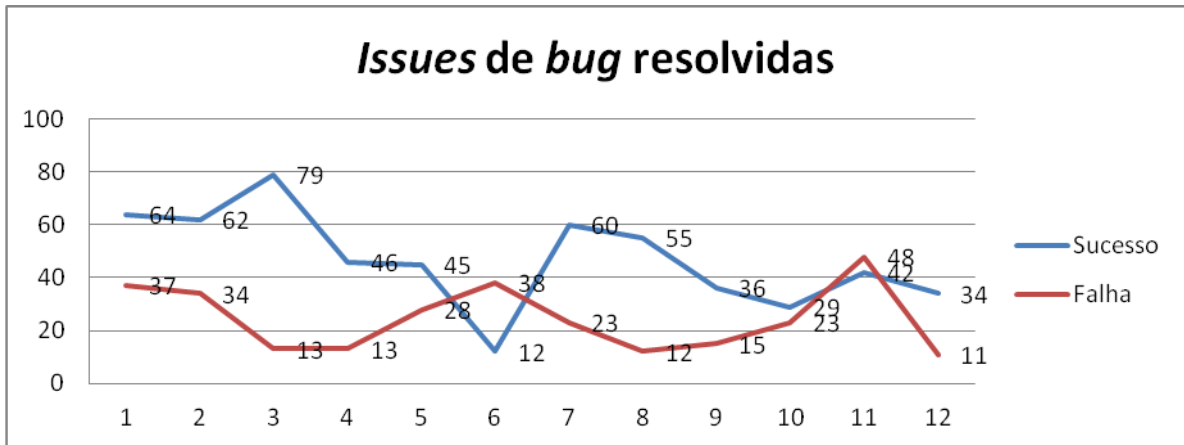


Gráfico 5 – Issues de bug resolvidas.

O mesmo podemos visualizar no gráfico 5, onde embora tenha havido variações, no período de falha a quantidade de *bugs* resolvidos foi superior no período de sucesso.

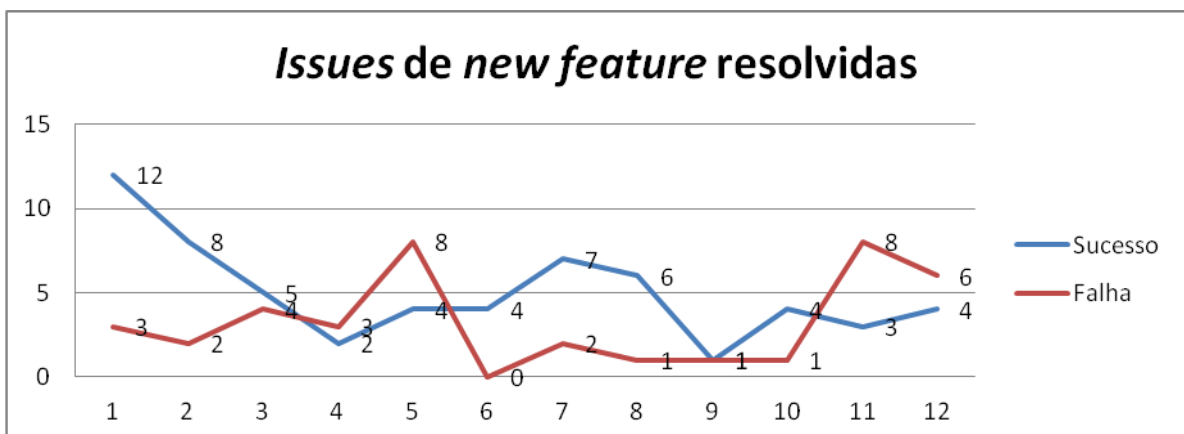


Gráfico 6 – Issues de new feature resolvidas

No gráfico 6 podemos ver que em 8 meses a quantidade de *issues* de *new feature* resolvidas foi superior no período de sucesso.

A Tabela 2 mostra os valores mínimos, máximos, média e desvio padrão obtidos para cada tipo de tipo de *issue* ao longo dos 12 meses de cada período.

Tabela 2 – Valores de mínimo, máximo, média, desvio padrão das métricas de tipo - *issues* resolvidas.

Tipo das <i>issues</i> resolvidas	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Improvement</i>	9	2	55	21	25,000	10,000	13,163	6,592
<i>Bug</i>	12	11	79	48	47,000	24,583	18,191	12,398
<i>Task</i>	0	0	9	3	3,500	1,167	2,680	1,115
<i>Subtask</i>	0	0	6	4	1,333	0,500	1,875	1,168
<i>New Feature</i>	1	0	12	8	5,000	3,250	2,954	2,734
<i>Wish</i>	0	0	2	1	0,167	0,250	0,577	0,452
<i>Test</i>	0	0	0	1	0,000	0,083	0,000	0,289

Com relação ao *status*, o comportamento também se mostrou na maior parte ser igual ao de tipo de *issue*, onde o período de sucesso teve maior atividade

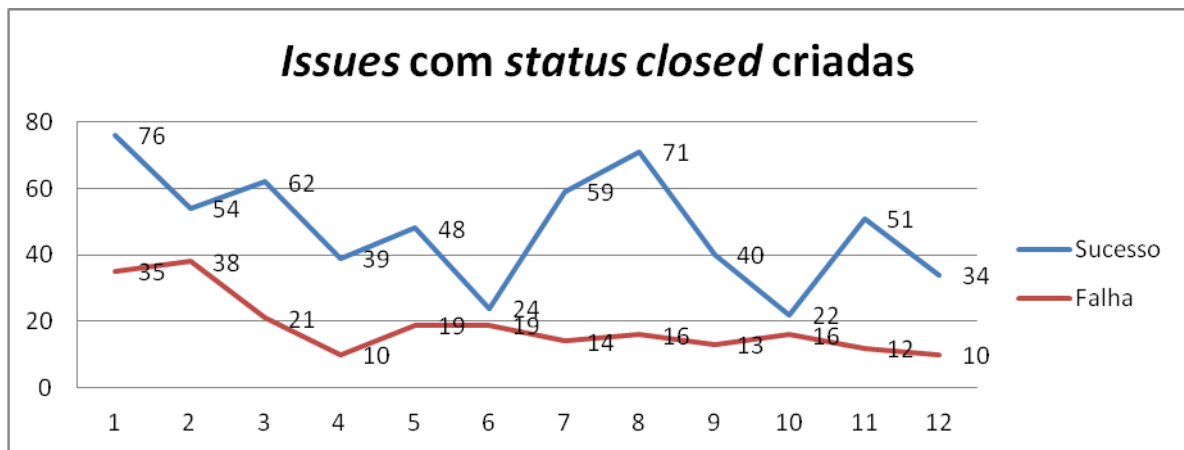


Gráfico 7 – *Issues* com *status closed* criadas.

No gráfico 7 podemos ver que a quantidade de *issues* com *status closed* foi bem superior no período de sucesso.

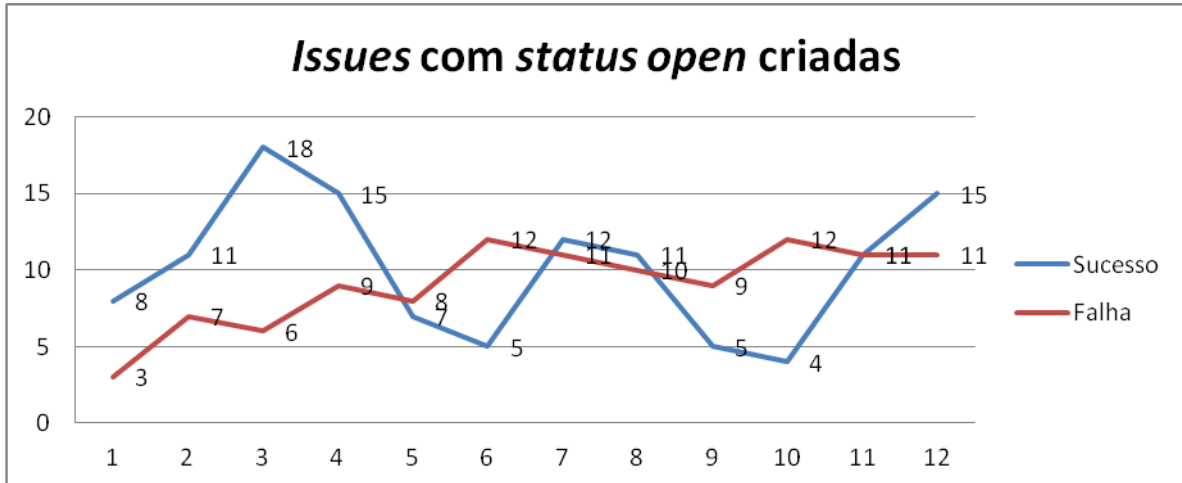


Gráfico 8 – Issues com status open criadas.

A quantidade de issues com status open se mostrou bem variável (gráfico 8).

Já a quantidade de issues criadas com o *status* de *resolved* tanto para o período de falha quanto para o período de sucesso foi igual a 0 ao longo dos períodos. Isto provavelmente se deve ao fato de não haver sentido em criar uma *issue* com o *status* de *resolved*, uma vez que a maioria das *issues* sempre serão um trabalho ainda a ser desenvolvido.

A Tabela 3 evidencia os valores mínimos, máximos, média e desvio padrão encontrado para o *status* das *issues* criadas para cada período.

Tabela 3 – Valores de mínimo, máximo, média, desvio padrão das métricas de *status* - *issues* criadas.

Status das issues criadas	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Closed	22	10	76	38	48,333	18,583	17,233	9,090
In Progress	0	0	0	0	0,000	0,000	0,000	0,000
Open	4	3	18	12	10,167	9,083	4,469	2,712
Reopened	0	0	1	1	0,250	0,167	0,452	0,389
Resolved	9	7	88	27	29,333	16,167	21,296	5,828

Em quase todos os valores possíveis de *status*, exceto *open*, a média do período de sucesso foi quase 50% maior que no período de falha. Este comportamento se repete para as *issues* de *status* resolvida, como mostra Tabela 4.

Tabela 4 – Valores de mínimo, máximo, média, desvio padrão das métricas de *status* - *issues* resolvidas.

<i>Issues</i> resolvidas de <i>Status</i>	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Closed</i>	12	9	88	55	51,167	23,750	22,036	16,344
<i>In Progress</i>	0	0	0	0	0,000	0,000	0,000	0,000
<i>Open</i>	0	0	0	0	0,000	0,000	0,000	0,000
<i>Reopened</i>	0	0	0	0	0,000	0,000	0,000	0,000
<i>Resolved</i>	10	8	96	29	30,833	16,083	23,257	7,821

Com relação à resolução das *issues* criadas e resolvidas, as tabelas abaixo resumem o valor para cada tipo de resolução.

Tabela 5 – Valores de mínimo, máximo, média, desvio padrão das métricas de resolução - *issues* criadas.

<i>Issues</i> criadas por resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Fixed</i>	33	18	139	55	71,500	30,667	29,346	11,904
<i>Won't fix</i>	0	0	3	3	1,250	1,417	1,138	1,165
<i>Duplicate</i>	0	0	4	2	1,667	0,833	1,303	0,718
<i>Invalid</i>	0	0	5	2	1,167	0,583	1,467	0,669
<i>Cannot Reproduce</i>	0	0	3	1	1,500	0,167	1,243	0,389
<i>Unresolved</i>	4	3	19	12	10,417	9,333	4,776	2,839
<i>Incomplete</i>	0	0	1	0	0,083	0,000	0,289	0,000
<i>Not a problem</i>	0	0	4	2	1,250	1,000	1,603	0,426
<i>Later</i>	0	0	0	0	0,000	0,000	0,000	0,000

A tabela 6 mostra os valores de mínimo, máximo, média e desvio padrão para as *issues* resolvidas.

Tabela 6 – Valores de mínimo, máximo, média, desvio padrão das métricas de resolução - *issues* resolvidas.

Issues resolvidas por resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Fixed</i>	25	15	137	56	75,000	33,667	30,648	15,263
<i>Wontfix</i>	0	0	3	9	1,333	1,833	1,303	2,368
<i>Duplicate</i>	0	0	5	4	2,167	1,000	1,403	1,206
<i>Invalid</i>	0	0	5	2	1,250	0,833	1,485	0,937
<i>Cannot Reproduce</i>	0	0	4	6	1,083	1,083	1,443	1,730
<i>Unresolved</i>	0	0	0	0	0,000	0,000	0,000	0,000
<i>Incomplete</i>	0	0	1	1	0,083	0,083	0,289	0,289
<i>Not a problem</i>	0	0	4	4	1,083	1,167	1,564	1,267
<i>Later</i>	0	0	0	0	0,000	0,000	0,000	0,000

É muito interessante perceber que a média das *issues fixed* é bem superior no período de sucesso, podendo ser um provável indicador do porque do sucesso do período. Outro dado interessante é que a média das *issues* com resolução *wontfix* também é superior.

No processo de criação de uma *issue* uma prioridade precisa ser dada a ela. Com relação à prioridade, percebe-se que a maior parte das *issues* são criadas com prioridade *major*.

A Tabela 7 mostra os valores para cada nível de prioridade das *issues* criadas no período de sucesso e no período de falha.

Tabela 7 – Valores de mínimo, máximo, média, desvio padrão das métricas de prioridade - *issues* criadas.

Prioridade das <i>issues</i> criadas	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Minor</i>	5	5	28	23	15,333	11,667	7,037	5,105
<i>Trivial</i>	0	0	8	3	2,417	1,000	2,151	0,853
<i>Major</i>	33	21	132	45	70,083	29,833	27,959	7,837
<i>Blocker</i>	0	0	1	2	0,167	0,583	0,389	0,793

<i>Critical</i>	0	0	5	3	1,167	0,917	1,528	1,084
-----------------	---	---	---	---	-------	-------	-------	-------

A tabela 8 mostra o nível de prioridade das *issues* resolvidas.

Tabela 8 – Valores de mínimo, máximo, média, desvio padrão das métricas de prioridade - *issues* resolvidas.

Prioridade das <i>issues</i> resolvidas	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
<i>Minor</i>	3	2	26	21	14,333	9,917	7,365	5,501
<i>Trivial</i>	0	0	7	2	1,917	0,750	2,234	0,866
<i>Major</i>	20	11	122	50	64,750	27,917	26,952	13,866
<i>Blocker</i>	0	0	1	2	0,167	0,583	0,389	0,793
<i>Critical</i>	0	0	3	2	0,833	0,667	0,937	0,778

Para descobrir se o comportamento detectado é válido, uma rede bayesiana foi treinada com os valores coletados de cada 30 dias dos 24 meses do projeto. Esta rede foi treinada com os valores das *issues* criadas e *issues* atualizadas de tipo, *status*, resolução e prioridade. Para que este comportamento seja considerado válido, o valor de *f-measure* da rede precisa ser alto, uma vez que quanto mais próximo de 1, maior a qualidade da rede. A porcentagem de dados utilizados para treino foi de 66% dos dados, enquanto 34% foi utilizado para predição. O tabela 9 demonstra o resultado do treino da rede bayes realizado na ferramenta WEKA.

Tabela 9 – Resultado do treino da rede bayesiana para Q1.

Classe	Precision	Recall	F-measure
Sucesso	1	0.6	0.75
Falha	0.6	1	0.75

4.1.2.1 Considerações dos resultados de Q1

O resultado do treino da rede bayesiana para Q1 mostra que valores de referência podem ser encontrados neste projeto uma vez que o resultado do treino teve valores de *precision*, *recall* e *f-measure* altos. No entanto, o valor de *precision* para sucesso mostrou-se bem superior ao do período de falha. Portanto o comportamento do projeto durante o período de sucesso possui um padrão mais estável enquanto os valores do período de falha mostram-se mais instáveis.

4.1.3. Q2: Com base em um padrão de comportamento, é possível encontrar valores de referência para as métricas?

Com os resultados em Q1 que validaram um padrão de comportamento, buscou-se estabelecer métricas para investigar a relação entre as principais métricas estudadas até aqui, de forma que seja possível encontrar então os valores de referência para elas.

De acordo com o quadro 3, foi feita investigação da relação entre as métricas do fator *issue*.

O primeiro valor de referência que se acredita ser importante é a relação entre as métricas que obtiveram maior frequência pelo total de *issues* daqueles 30 dias. Assim, a média e o desvio padrão foram calculados para estas métricas para cada período.

Tabela 10 – Valores de mínimo, máximo, média, desvio padrão das métricas de tipo pelo total - *issues* criadas.

Relação entre métricas de tipo pelo total	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Bug / Total	0,473	0,078	0,733	0,793	0,574	0,597	0,078	0,293
Improvement / Total	0,128	0,084	0,418	0,356	0,297	0,266	0,084	0,123
New Feature / Total	0,013	0,000	0,150	0,178	0,059	0,094	0,034	0,079

Esta relação também foi buscada entre as métricas. A tabela a baixo mostra a relação buscada entre elas.

Tabela 11 – Valores de mínimo, máximo, média e desvio padrão das métricas de relação de tipo - issues criadas.

Relação das métricas de tipo	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Improvement / Bug	0,175	0,000	0,852	0,733	0,541	0,460	0,202	0,142
New Feature / Bug	0,025	0,000	0,273	0,286	0,105	0,165	0,062	0,103
New Feature / Improvement	0,032	0,000	0,636	0,889	0,236	0,381	0,188	0,272

Desta forma, nosso valor de referência é a média, tendo como desvio o desvio padrão calculado para o período de sucesso e o período de falha.

Para validar estes valores, outra rede bayesiana foi treinada com os valores referentes a cada 30 dias de extração de dados (tabela 12).

Tabela 12 – Resultado do treino da rede bayesiana para tipo criada.

Classe	Precision	Recall	F-measure
Sucesso	0.8	0.8	0.733
Falha	0.667	0.667	0.733

O resultado do treino da rede, que foi feito da mesma forma como o treino para validar os dados referente ao comportamento das *issues* (ver figura 5) obteve como resultado de *precision* 0.8 e *recall* de 0.8 para sucesso, e 0.667 para ambos na classificação de falha, já *f-measure* obteve 0.8 para sucesso e 0.667 para falha. Isso demonstra que o comportamento do período de falha é instável diminuindo assim o número de *precision* e *recall* da rede e dificultando encontrar para estas métricas valores de referência para o período de falha.

A Tabela 13 mostra os valores de mínimo, máximo, média e desvio padrão encontrado para os períodos de 30 dias do tipo de *issue*.

Tabela 13 – Valores de mínimo, máximo, média, desvio padrão das métricas de tipo pelo total - *issues* resolvidas.

Relação de métricas de tipo pelo Total	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Bug / Total Improvement /	0,480	0,406	0,730	0,844	0,576	0,620	0,078	0,113
Total	0,186	0,105	0,423	0,375	0,302	0,238	0,078	0,089
New Feature /								
Total	0,014	0,000	0,160	0,316	0,066	0,092	0,040	0,085

A tabela 14 mostra as métricas relacionadas de acordo com os tipos de *issue*.

Tabela 14 – Valores de mínimo, máximo, média e desvio padrão das métricas de relação de tipo - *issues* resolvidas.

Relação das métricas de tipo	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Improvement / Bug	0,261	0,132	0,833	0,923	0,549	0,415	0,202	0,227
New Feature / Bug	0,028	0,000	0,333	0,545	0,119	0,163	0,081	0,156
New Feature / Improvement	0,033	0,000	0,444	3,000	0,233	0,518	0,130	0,808

Para validar se estes valores de média podem ser usados como valor de referência, uma rede bayesiana (tabela 15) foi treinada para validar estes valores.

Tabela 15 – Resultado do treino da rede bayesiana para tipo - *issues* resolvidas.

Classe	Precision	Recall	F-measure
Sucesso	0.6	0.6	0.75
Falha	0.333	0.333	0.75

O resultado obtido nesta classificação não foi satisfatório, uma vez que os valores de *precision* e *recall* e *f-measure* foram baixos, portanto, não seria confiável utilizar aquelas médias como valores de referência.

O mesmo processo foi realizado para os valores encontrados das issues criadas e resolvidas para *status*, resolução e prioridade.

Abaixo temos os valores mínimos, máximos, médios e desvio padrão encontrados para a relação das issues criadas para os intervalos de 30 dias de cada período – sucesso e falha.

Tabela 16 – Valores de mínimo, máximo, média, desvio padrão das métricas de status pelo total - issues criadas.

Relação das métricas de status pelo total	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Closed / Total	0,367	0,267	0,718	0,648	0,560	0,413	0,120	0,132
Resolved / Total	0,127	0,206	0,527	0,467	0,318	0,363	0,130	0,084

A tabela 17 mostra as métricas relacionadas de acordo com os status das *issues*.

Tabela 17 – Valores de mínimo, máximo, média e desvio padrão das métricas de relação de status - issues criadas.

Relação das métricas de status	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Resolved / Closed	0,176	0,333	1,419	1,750	0,641	0,999	0,402	0,446
Open / Closed	0,105	0,086	0,441	1,100	0,222	0,615	0,102	0,312

Open / Resolved	0,138	0,188	1,222	1,000	0,479	0,623	0,362	0,254
-----------------	-------	-------	-------	-------	-------	-------	-------	-------

E o resultado do treino para validar estes valores podem ser vistos na figura abaixo (tabela 18).

Tabela 18 – Resultado do treino da rede bayesiana para valores de *status* - *issues* criadas.

Classe	Precision	Recall	F-measure
Sucesso	0.8	0.8	0.8
Falha	0.667	0.667	0.667

Os valores de *precision* e *recall* para estes valores foram relativamente altos para a classificação de sucesso, no entanto, o mesmo não ocorreu com a classificação de falha. Portanto, utilizar os valores de média calculados para o período de falha não seria recomendado, pois seu *precision* e *recall* são baixos.

Com relação às métricas de *issues* resolvidas por *status* a tabela 19 mostra os valores encontrados.

Tabela 19 – Valores de mínimo, máximo, média, desvio padrão das métricas de status pelo total - *issues* resolvidas.

Relação de <i>issues</i> de <i>status</i> pelo total	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Closed / Total	0,368	0,361	0,854	0,797	0,629	0,568	0,154	0,153
Resolved / Total	0,146	0,203	0,632	0,639	0,371	0,432	0,154	0,153

A tabela 20 mostra as métricas relacionadas de acordo com os tipos de *issue*.

Tabela 20 – Valores de mínimo, máximo, média e desvio padrão das métricas de relação de *status* - issues resolvidas.

Relação de Issues de <i>status</i>	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Resolved / Closed	0,170	0,255	1,714	1,769	0,693	0,889	0,475	0,535
Open / Closed	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Open / Resolved	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000

A tabela 21 mostra o resultado do treino da rede.

Tabela 21 – Resultado do treino da rede bayesiana para *status* - issues resolvidas.

Classe	Precision	Recall	F-measure
Sucesso	0.5	0.2	0.286
Falha	0.333	0.667	0.444

Os valores de média e desvio padrão encontrados para os valores de *issues* por resolução encontram-se na tabela 22.

Tabela 22 – Valores de mínimo, máximo, média, desvio padrão das métricas de resolução pelo total – issues criadas.

Issues resolvidas de resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Total	0,61	0,53	0,91	0,87	0,801	0,679	0,078	0,114
Unresolved / Total	0,07	0,06	0,24	0,32	0,121	0,226	0,052	0,085

A tabela 23 mostra as métricas relacionadas de acordo com as resoluções das *issues*

Tabela 23 – Valores de mínimo, máximo, média e desvio padrão das métricas de relação de resolução – issues criadas.

Issues resolvidas de resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Unresolved / Fixed	0,07	0,06	0,39	0,61	0,158	0,359	0,090	0,174

A validação destes valores estão na tabela 24. Os valores de *precision* e *recall* deste treino foram altas, o que permite usar aqueles valores de média como valores de referência para seus respectivos períodos do projeto.

Tabela 24 – Resultado do treino da rede bayesiana para resolução – *issues* criadas.

Classe	Precision	Recall	F-measure
Sucesso	0.833	1	0.909
Falha	1	0.667	0.8

Note que neste caso não serão estudadas as métricas de relação entre os valores possíveis para as resoluções das *issues*, uma vez que os valores neste caso eram sempre 0.

Tabela 25 – Valores de mínimo, máximo, média, desvio padrão das métricas de resolução pelo total – *issues* resolvidas.

Issues resolvidas de resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Total	0,83	0,70	1,00	0,95	0,919	0,855	0,052	0,072

Para esse valor de referência, a rede bayesiana (tabela 26) demonstrou uma baixo valor em *precision* e *recall*, invalidando portanto estes valores para referência.

Tabela 26 – Resultado do treino da rede bayesiana para métricas de resolução.

Classe	Precision	Recall	F-measure
--------	-----------	--------	-----------

Sucesso	0.667	0.8	0.727
Falha	0.5	0.333	0.4

Sobre a prioridade, as seguintes métricas foram relacionadas com o intuito de se encontrar e validar um valor de referência para elas (tabela 20 e 21).

Tabela 27 – Valores de mínimo, máximo, média, desvio padrão das métricas de prioridade pelo total – *issues* criadas.

Métricas de prioridade pelo total	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Minor / Total	0,066	0,152	0,273	0,426	0,178	0,259	0,067	0,076
Major / Total	0,689	0,556	0,880	0,788	0,781	0,681	0,072	0,082

A tabela 28 mostra as métricas relacionadas de acordo com a prioridade das *issues*.

Tabela 28 – Valores de mínimo, máximo, média e desvio padrão da métrica de relação de prioridade – *issues* criadas.

Métricas de prioridade	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Minor / Major	0,077	0,192	0,395	0,767	0,236	0,397	0,107	0,164

A validação destas métricas, que também se deu pelo treino de uma rede bayesiana (tabela 29) demonstrou que os valores de *precision* e *recall* são baixos, o que torna o valor encontrado da média um valor não preciso como valor de referência.

Tabela 29 – Resultado do treino da rede bayesiana para prioridades – criadas.

Classe	Precision	Recall	F-measure
Sucesso	0.6	0.6	0.6
Falha	0.4	0.333	0.333

Para os valores de *issues* resolvidas por prioridade temos as seguintes relações propostas (tabela 30 e 31).

Tabela 30 – Valores de mínimo, máximo, média, desvio padrão das métricas de prioridade pelo total – *issues* resolvidas.

Métricas de prioridade	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Minor / Total	0,042	0,095	0,257	0,421	0,177	0,248	0,061	0,084
Major / Total	0,712	0,579	0,915	0,810	0,791	0,697	0,056	0,077

A tabela 31 mostra as métricas relacionadas de acordo com a resolução das *issues*.

Tabela 31 – Valores de mínimo, máximo, média e desvio padrão da métrica de relação de prioridade – *issues* resolvidas.

Issues resolvidas de resolução	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Minor / Major	0,046	0,118	0,351	0,727	0,229	0,370	0,088	0,161

Com o treino de validação da rede bayesiana para estes valores, descobrimos que o valor de *precision* para classificar os valores do período de sucesso foi igual a 1, isto é, 100% de precisão, no entanto, o mesmo não ocorreu com o *precision* do período de falha, tornando assim os valores médios de falha inapropriados para valores de referência. Já o valor de f-measure se mostrou em 0.75 em ambos os casos, mostrando que a qualidade do modelo é relativamente alta.

Tabela 32 – Resultado do treino da rede bayesiana para prioridade – resolvida.

Classe	Precision	Recall	F-measure
Sucesso	1	0.6	0.75
Falha	0.6	1	0.75

Investigando agora os aspectos do fator desenvolvedor, conforme quadro 3, os valores das métricas da tabela 33 foram calculados.

Tabela 33 – Valores de mínimo, máximo, média, desvio padrão da métrica de desenvolvedor.

Métrica de desenvolvimento	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Quantidade de desenvolvedor	10	7	17	13	13,583	10,500	2,109	1,931

O resultado do treino da rede bayesiana desta vez mostrou que a rede pode prever com maior qualidade o período de sucesso, uma vez que a rede conseguiu encontrar nos valores dos períodos de sucesso um padrão, tornando assim a predição do sucesso com valores de *precision*, *recall* e *f-measure* maiores.

Tabela 34 – Resultado do treino da rede bayesiana para quantidade de desenvolvedor.

Classe	Precision	Recall	F-measure
Sucesso	0.8	0.8	0.933
Falha	0.667	0.667	0.667

Ainda dentro do fator desenvolvedor, criamos uma métrica que tentasse descobrir se há relação entre o número de desenvolvedores com a quantidade de *issues* resolvidas (*fixed*). A tabela abaixo mostra o valor mínimo, máximo, média e desvio padrão para esta relação.

Tabela 35 – Valores de mínimo, máximo, média, desvio padrão da métrica de relação de desenvolvedor por *issues fixed – issues criadas*.

Métrica de desenvolvimento	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Assignee	2,200	2	8,176	4,273	5,214	2,904	1,672	0,870

O interessante desta métrica foi descobrir que sua validação obteve valores de *precision*, *recall* e *f-measure* bem elevados, nos indicando assim que o valor de média calculado para esta métrica pode ser utilizado como valor de referência.

Tabela 36 – Resultado do treino da rede bayesiana para métrica de relação de desenvolvedor por *fixed – issues criadas*.

Classe	Precision	Recall	F-measure
Sucesso	1	0.8	0.889
Falha	0.75	1	0.857

O mesmo foi realizado com os valores obtidos dos subperíodos de issues resolvidas.

Tabela 37 – Valores de mínimo, máximo, média, desvio padrão da métrica de relação de desenvolvedor por *issues fixed – issues resolvidas*.

Métrica de desenvolvimento	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Assignee	1,667	1	8,059	5,300	5,488	3,178	1,785	1,227

E o resultado da rede bayesiana para estes valores também se mostraram muito satisfatórios (tabela 38).

Tabela 38 – Resultado da rede bayesiana para métrica de relação de desenvolvedor por *fixed – issues* resolvidas.

Classe	Precision	Recall	F-measure
Sucesso	1	0.8	0.889
Falha	0.75	1	0.857

O último fator restante (conforme quadro 3) para estudo é o de comunicação. A tabela abaixo mostra os valores mínimos, máximos, médio e desvio padrão para a troca de mensagem de cada período.

Tabela 39 – Valores de mínimo, máximo, média, desvio padrão da métrica de comunicação.

Métrica de comunicação	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Número de mensagens	132	62	420	182	257	123,750	92,119	38,653

Para investigar esses valores uma rede bayesiana foi montada. Na tabela 40 podemos ver o resultado do treino. Os valores de *precision* e *recall* se alternaram entre 1 e 0,6, mas o valor de *f-measure* que ficou em 0.75 mostra que o modelo tem uma capacidade de 75% de prever o resultado do *build*.

Tabela 40 – Resultado do treino da rede bayesiana para quantidade de mensagens.

Classe	Precision	Recall	F-measure
Sucesso	1	0.6	0.75
Falha	0.6	1	0.75

Por ultimo, para se encontrar um valor de referência ainda no fator comunicação uma relação foi estabelecida conforme quadro 3 entre quantidade de issues com resolução *fixed* pela quantidade ou número de mensagens do período.

Tabela 41 – Valores de mínimo, máximo, média, desvio padrão da métrica de relação de comunicação por *issues fixed – issues criadas*.

Métrica de comunicação	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Nº. de mensagens	0,212	0,156	0,350	0,346	0,277	0,253	0,043	0,069

A rede bayesiana para estes valores mostrou um alto valor de *precision*, *recall* e *f-measure*.

Tabela 42 – Resultado do treino da rede bayesiana para métrica de relação de mensagem por *fixed – issues criadas*.

Classe	Precision	Recall	F-measure
Sucesso	1	0.8	0.889
Falha	0.75	1	0.857

Como possuímos dois valores para a métrica de *fixed*, valores para as *issues* criadas e valores para as *issues* resolvidas, portanto, o mesmo processo foi executado para os valores das *issues* resolvidas do período. A tabela abaixo exhibe os valores calculados.

Tabela 43 – Valores de mínimo, máximo, média, desvio padrão da métrica de relação de comunicação por *issues fixed – issues resolvidas*.

Métrica de comunicação	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha	Sucesso	Falha
	Mínimo		Máximo		Média		Desvio Padrão	
Fixed / Nº. de	0,189	0,153	0,363	0,482	0,289	0,271	0,049	0,095

mensagens								
-----------	--	--	--	--	--	--	--	--

O resultado da rede bayesiana montada para validar estes valores está na tabela 44.

Tabela 44 – Resultado do treino da rede bayesiana para métrica de mensagem – resolvida.

Classe	Precision	Recall	F-measure
Sucesso	0.75	0.6	0.667
Falha	0.5	0.667	0.571

O resultado de treino desta rede obteve valores baixos para *precision*, *recall* e *f-measure*, invalidando a utilização daqueles valores.

4.1.3.1 Considerações dos resultados de Q2

Foi percebido que algumas métricas obtiveram em seu valor de média um valor de referência, isso causado pela pouca quantidade de dados. Uma maior quantidade de dados poderia ter tornado os valores mais precisos. Considerando um valor de *f-measure* acima de 0.7 as métricas que não obtiveram seus valores validados, são as das seguintes tabelas: Tabela 16 e 17 para período de falha; Tabela 25 também para falha; Tabela 27 e 28 para sucesso e falha; Tabela 34 para o período de falha e a tabela 44 para ambos os períodos. Para todas as outras os valores de média puderam ser validados, isto é, podem ser utilizados como valor de referência para inferência do estado do projeto.

A tabela 25 exhibe os valores de *precision*, *recall* e *f-measure* para todas as métricas investigadas em Q2.

Tabela 45 – Valores de precision, recall e f-measure das métricas de Q2.

Métrica	Sucesso			Falha		
Fator Issue	Precision	Recall	F-measure	Precision	Recall	F-measure
Issues criadas						
Bug/Total	0.8	0.8	0.733	0.667	0.667	0.733
Improvement/Total						
New Feature/Total						
Improvement/Bug						
New Feature/Bug						
New Feature/Improvement						
Issues resolvidas						
Bug/Total	0.6	0.6	0.75	0.333	0.333	0.75
Improvement/Total						
New Feature/Total						
Improvement/Bug						
New Feature/Bug						
New Feature/Improvement						
Issues Criadas						
Closed/Total	0.8	0.8	0.8	0.667	0.667	0.667
Resolved/Total						
Resolved/Closed						
Open/Closed						
Open/Resolved						
Issues Resolvidas						
Closed/Total	0.5	0.2	0.286	0.333	0.667	0.444
Resolved/Total						
Resolved/Closed						

Open/Closed						
Open/Resolved						
Issues Criadas						
Fixed/Total	0.833	1	0.909	1	0.667	0.8
Unresolved/Total						
Unresolved/Fixed						
Issues Resolvidas						
Fixed/Total	0.667	0.8	0.727	0.5	0.333	0.4
Issues Criadas						
Minor/Total	0.6	0.6	0.6	0.4	0.333	0.333
Major/Total						
Minor/Major						
Issues Resolvidas						
Minor/Total	1	0.6	0.75	0.6	1	0.75
Major/Total						
Minor/Major						
Fator Desenvolvedor						
Quantidade de desenvolvedor	0.8	0.8	0.933	0.667	0.667	0.667
Issues Criadas						
Fixed/Assigneed	1	0.8	0.889	0.75	1	0.857
Issues Resolvidas						
Fixed/Assignee	1	0.8	0.889	0.75	1	0.857
Fator Comunicação						
Número de Mensagens	1	0.6	0.75	0.6	1	0.75
Issues Criadas						
Fixed /Nº. Mensagens	1	0.8	0.889	0.75	1	0.857
Issues Resolvidas						

Fixed/Nº. Mensagens	0.75	0.6	0.667	0.5	0.667	0.571
---------------------	------	-----	-------	-----	-------	-------

4.1.4. Q3: Qual a influência de uma métrica sobre as demais para o período de sucesso e o período de falha?

Para responder esta questão uma Análise de Componentes Principais (PCA) foi utilizada para cada período. Com o resultado desta análise podemos ver qual a influência das métricas e as forças que elas representam.

Para a PCA foi utilizados os valores das relações das métricas. Isto é, foi criada uma matriz de correlação com os valores das métricas que foram relacionadas (e.g., a relação de *improvement* por *bug*). O gráfico abaixo exibe o resultado da análise.

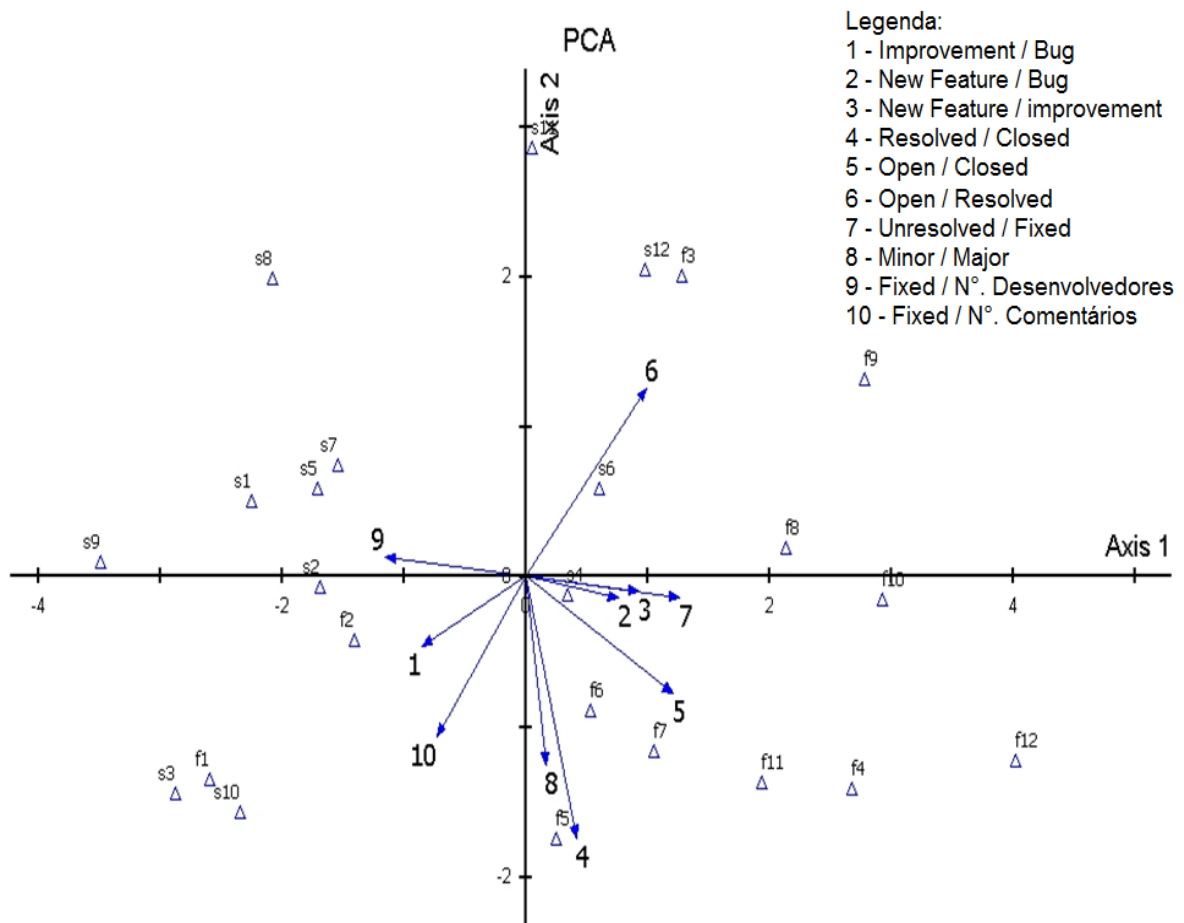


Figura 22 – Análise PCA para das métricas.

Os pontos S1 à S12 correspondem aos 12 meses de sucesso, já os pontos F1 à F12 correspondem aos 12 meses de falha.

4.1.4.1 Considerações dos resultados de Q3

O resultado desta análise pode nos demonstrar que a métrica 9 (*fixed/ n°. desenvolvedores*) é influenciadora por ela estar correlacionada às amostra de sucesso (s7, s5, s1, s2 e s9), portanto, quando maior o valor desta métrica maior será o sucesso, ao passo que quanto menor seu valor, maior o probabilidade de ocorrer falha. Ao contrário de 9 temos a métrica 7 (*unresolved/fixed*) que se mostrou diagonalmente contrária a 9 sendo portanto influenciadora de falha, isto é, quanto maior seu valor maior, maior será a probabilidade de haver falha. Altamente correlacionadas a 7 estão 2 (*new feature/bug*) e 3 (*new feature/improvement*) portanto estas métricas também estão relacionadas a falha. Como o eixo x é responsável por explicar a maior parte da correlação e influência (Oliveira, 2011) estas se mostraram as principais métricas influenciadoras de sucesso/falho das métricas analisadas pela PCA. No entanto se pode observar também métricas como 6 (*open/resolved*) e 10 (*fixed/n°. comentários*) são métricas que possuem valores diagonalmente opostos.

5. AMEAÇAS DE VALIDAÇÃO

A calibragem de uma rede bayesiana está diretamente ligada à quantidade de dados a ela fornecida. Quanto maior a quantidade de dados a ela fornecida, maior será sua precisão, portanto uma das ameaças de validação deste trabalho se consiste principalmente no fato de que a quantidade de dados obtidos do projeto Felix Gogo é pequena. Os períodos de estudo totalizam 24 meses, sendo 12 meses para cada período, o que gera uma quantidade de dados inferior ao ideal para poder calibrar uma rede.

Outra ameaça se consiste no fato de que os valores de referencia encontrados são válidos apenas para o projeto Felix Gogo em particular. A inexistência de projetos que tivessem o histórico dos *builds* completo, ou fases de sucesso e falha capazes de fornecer um estudo consistente. Portanto, diante disto o único projeto que pode se enquadrar nestes quesitos foi o projeto por este trabalho estudado.

6. CONCLUSÃO

Diante de um cenário onde boa parte dos projetos de código fonte aberto podem ser considerados falhos este trabalho buscou encontrar um padrão de comportamento em um projeto de software de código fonte aberto e através disso encontrar valores de referência para métricas com o intuito de fornecer meios para um mantenedor de projeto, desenvolvedor ou a quem possa interessar inferir sobre o andamento do projeto.

Após uma revisão na literatura onde se buscou encontrar as métricas influenciadoras de sucesso em um projeto e o selecionamento do repositório da Apache Software Foundation como ambiente para selecionamento do projeto de estudo uma análise no repositório foi feita para se descobrir as métricas que dele poderiam ser também estudadas.

O projeto selecionado foi o Apache Felix Gogo, pois foi o único projeto encontrado no repositório que possuía o histórico das *builds* completo além de períodos de sucesso e falha bem definidos, o que possibilitaria um estudo mais consistente.

Após a coleta dos dados do projeto, os dados foram normalizados para períodos de 30 dias e então o valor de média e desvio padrão foi calculado para cada uma das métricas propostas e então submetidas as nossas questões de pesquisa de acordo com nossa metodologia.

Com relação a nossos resultados, em Q1 pudemos verificar que o nível de atividade no projeto se mostrou maior durante o período de sucesso o que pudemos observar através dos gráficos comparativos para cada período de cada métrica. Ainda em Q1 este comportamento foi validado ao se submeter os dados dos 24 meses estudados do projeto a uma rede bayesiana que obteve como resultado um valor de *f-measure* relativamente alto.

Em Q2 nem todas as métricas puderam ter seus valores propostos validados, isso possivelmente está ligado a pequena quantidade de dados que puderam ser encontrados, pois esta foi uma dos maiores problemas para a execução deste estudo. No entanto as métricas que obtiveram valores de *precision*, *recall* e *f-measure* altos podem ser usadas no Felix Gogo como forma de inferir o andamento do projeto. Uma vez com o valor de referência em mãos o mantenedor ou desenvolvedor deve realizar uma coleta de dados de um período de 30 dias e submeter os valores coletados às métricas neste trabalho e dessa forma inferir como anda o projeto levando-se em conta o desvio padrão para cada métrica.

Sobre a Análise PCA, ela pode verificar quais métricas possuem relação entre si e a força que cada uma tem dentro o conjunto de métricas estudadas na análise. Pudemos verificar através dela que a métrica 9 (*fixed/nº. desenvolvedores*) é a principal responsável pelo fator sucesso ao passo que a métrica 7 (*unresolved/fixed*) é a influenciadora de falha, está oposta ao valores da métrica 9 e que os valores das métricas 2 (*new feature/bug*) e 3 (*new feature/improvement*) estão correlacionados com 7. Portanto a PCA combinada com Q2 indica que a métrica 9 e 7 são métricas que são influenciadoras de sucesso/falha e por terem obtido altos valores de *precision* seus valores de média são capazes de inferir com alto grau de qualidade o estado do projeto.

Portanto, diante das questões levantadas neste trabalho e por ter sido elas todas respondidas através de métodos científicos este trabalho contribui por fornecer métricas que podem ser utilizadas com valores de referência capazes de inferir sobre o estado do projeto Apache Felix Gogo e por ter descoberto as métricas que possuem maior influência sobre o sucesso/falha do projeto.

7. TRABALHOS FUTUROS

Durante a execução deste estudo métricas da literatura foram utilizadas para se estudo além da investigação de outras métricas encontradas dentro do Repositório Apache tendo também algumas relações entre estas métricas analisadas. Com relação às métricas da literatura nem todas puderam ser investigadas, pois o estudo foi realizado no repositório Apache, diante disto, seria interessante escolher um repositório de forma a aplicar todas as métricas encontradas na literatura.

Pode-se perceber a importância de se ter uma grande quantidade de dados para serem analisados, por isso, um trabalho futuro pode advir exatamente desta questão. Para isso, uma coleta de dados de um longo período dos projetos seria necessário. Este tempo se faz necessário porque este estudo para ser replicado para outros projetos Apache necessitaria coletar continuamente os dados das *builds* uma vez que os dados não são armazenados pela fundação Apache.

Outro trabalho que pode ser executado é a replicação destes dados para projetos do Sourceforge.com, um dos repositórios de projetos de OSS mais conhecidos.

As métricas propostas de relação deste estudo foram baseadas nos fatores issues, comunicação e desenvolvedor, portanto seria interessante levantar outras métricas para investigar seus valores e sua influência sobre o sucesso/falha.

REFERÊNCIAS

- ARAUJO, Wagner O. D. **Análise de Componente Principais (PCA)**. Mestrado Sociedade, Tecnologia e Meio Ambiente. Anápolis, p. 12. 2009.
- BEECHER, Karl; CAPILUPPI, Andrea; BOLDYREFF, Cornelia. **Identifying exogenous drivers and evolutionary stages in FLOSS projects**. In: Journal of Systems and Software. New York: ACM Digital Library. 2009. p. 739 - 750.
- 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.
- BONESS, Kenneth; HARRISON, Rachel. **A Data Collection Case Study Supporting Requirements Oriented Prediction and Management in Software Developments**. In: Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02). 2002. p. 37 - 40.
- BOUCKAERT, Remco R; FRANK, Eike; HALL, Mark; KIRKBY, Richard; REUTEMANN, Peter; SEEWALD, Alex; SCUSE, David. **Weka Manual**. In: The University Of Waikato. 2010. p. 1 – 303.
- BROOKS, Frederick P. **The Mythical Man-Month: Essays on Software Engineering.**, 1975.
- CAPRA, Eugeio; FRANCALANCI, Chiara; MERLO, Francesco; ROSSI-LAMASTRA, Cristina. **Firms' involvement in Open Source projectsA trade-off between software structural quality and popularity**. In: 2011. Milano: Journal of Systems and Software. 2008. p. 144-161.
- CHARLES RIVER ANALYTICS. **About Bayesian Belief Networks**. 2004. p. 14.
- COMINO, Stefano; MANENTI, Fabio M.; PARISI, Maria L. **From planning to matureOn the success of open source projects**. 2007. p. 1575 - 1586.
- DEMARCO, Tom; LISTER, Timothy. **Peopleware: Productive Projects and Teams.**, 1999.
- FERREIRA, Kecia A. M. **Um Modelo De Predição de Amplitude da Propagação de Modificações Contratuais em Software Orientado por Objetos**. In: Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Belo Horizonte. 2011.
- FERSHTMAN, Chaim; GANDAL, Neil. **The Determinants of Output Per Contributor in Open Source ProjectsAn Empirical Examination**. 2004.
- FLATIRONS, INC. **Accuracy and Precision**. 2006. p. 3.

APACHE FOUNDATION, Apache. Apache. **Apache Foundation**, 2011. Disponível em: <<http://projects.apache.org>>. Acesso em: 18 junho 2011.

GIANG, Le T.; KANG, Dongwon; BAE, Doo-Hwan. **Software Fault Prediction Models for Web Applications**. In: 34th Annual IEEE Computer Software and Applications Conference Workshops. 2010. p. 51 - 56.

GONZALEZ-BARAHONA, Jesus M.; ROBLES, Gregorio; DUEÑAS, Santiago. **Collecting data about FLOSS development**. 2010. p. 32-38.

HU, Yong; HUANG, Jiaying; CHEN, Juhua; LIU, Mei; XIE, Kang. **Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches**. In: Natural Computation, 2007. ICNC 2007. Third International Conference on. Haikou: 2007. p. 358 - 362.

HU, Yong; ZHANG, Xiangzhou; SUN, Xin; LIU, Mei; DU, Jianfeng. **An Intelligent Model for Software Project Risk Prediction**. In: 2009. p. 629 - 632.

MEIRELLES, Paulo; JR., Carlos S.; KON, Fabio; TERCEIRO, Antonio; CHAVEZ, Christina. **A Study of the Relationships between Source Code Metrics and Attractiveness in Free Software Projects**. In: Proceeding SBES '10 Proceedings of the 2010 Brazilian Symposium on Software Engineering. Washington, DC: 2001. p. 11-20.

OLIVEIRA, Elton Celton de. **A influência de fatores ambientais na estrutura da ictiofauna demersal em um estuário subtropical do Brasil: Efeitos em micro e macro-escala**. In: Tese Doutorado Universidade Federal do Paraná. Curitiba 2011. p. 1 – 145.

QU, Ying; TANG, Xin-Liang. **Software project risk assessing model based on credal networks**. In: Machine Learning and Cybernetics (ICMLC), 2010 International Conference on. Qingdao: 2010. p. 1976 - 1979.

RAINER, A; GALE, S. **Evaluating the Quality and Quantity of Data on Open Source Software Projects**. In: First International Conference on Open Source Software. Genova: 2005. p. 11 - 15.

SAMOLADAS, Ioannis; ANGELIS, Lefteris; STAMELOS, Ioannis. **Survival analysis on the duration of open source projects**. In: Journal Information and Software Technology. Butterworth-Heinemann Newton: ACM Digital Library. 2010. p. 902 - 922.

SANTOS JR., Carlos D.; PEARSON, John; KON, Fabio. **Attractiveness Of Free And Open Source Projects**. In: 18th European Conference on Information Systems. 2010. p. 432 - 444.

SANTOS, Maria A. M. R. D. **Extraindo regras de associação a partir de textos**. In: Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada. Curitiba: 2002. p. 71.

SATO, Danilo Toshiaki. **Uso Eficáz de Métricas em Métodos Ágeis de Desenvolvimento de Software**. In: Dissertação Instituto de Matemática e Estatística da Universidade de São Paulo. São Paulo. 2008. p. 1 – 155.

STEPHANY, François; MENS, Tom; GÎRBA, Tudor. **Maispion A Tool for Analysing and Visualising Open Source Software Developer Communities**. In: IWST '09 Proceedings of the International Workshop on Smalltalk Technologies. Nova York: ACM Digital Library. 2009. p. 50 - 57.

SUBRAMANIAM, Chandrasekar; SEN, Ravi; NELSON, Matthew L. **Determinants of open source software project success: A longitudinal study**. In: Journal Decision Support Systems. Amsterdam: ACM Digital Library. 2009. p. 576 - 585.

VEN, Kris; VERELST, Jan; MANNAERT, Herwing. **Should You Adopt Open Source Software?** In: Journal IEEE Software. Los Alamitos: ACM Digital Library. 2008. p. 54 - 59.

WANG, Yi; GUO, Defeng; SHI, Huihui. **Measuring the evolution of open source software systems with their communities**. In: ACM SIGSOFT Software Engineering Notes. New York: ACM Digital Library. 2007. p. 208 - 214.

WEISS, Dawid. **Measuring Success of Open Source Projects Using Web Search Engines**. In: The First International Conference on Open Source Systems. Genova: 2005. p. 234 - 240.

WOLF, Timo; SCHROTER, Adrian; DAMIAN, Daniela; NGUYEN, Thanh. **Predicting build failures using social network analysis on developer communication**. In: Proceeding ICSE '09 Proceedings of the 31st International Conference on Software Engineering. Washington, DC: ACM Digital Library. 2009. p. 1 - 11.

WU, Jiang; GOH, Khim Y. **Evaluating Longitudinal Success of Open Source Software ProjectsA Social Network Perspective**. In: HICSS '09. 42nd Hawaii International Conference. IEEE Xplore Digital Library. 2009. p. 1-10.

XU, Zhihong; ZHENG, Xin; GUO, Ping. **Empirically Validating Software Metrics for Risk Prediction Based on Intelligent Methods**. In: Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on. Jinan: IEE Xplore. 2006. p. 1049 - 1054.