

CADERNOS DO IME

Série Informática

Volume 42

Implementação e Avaliação de Co-Processadores para BlackScholes em FPGA	7
Walter Sousa, Marcelle Engle Carvalho Sacramento, Alexandre S. Nery, Maria Clícia S. Castro	
Predição de Processos com Dados não Estruturados Heterogêneos	18
Matheus Augusto de Oliveira, Flavia Maria Santoro, Kate Revoredo, Rosa Maria E. Moreira da Costa.	
Métricas de Tamanho de Software com Métodos Ágeis no Setor Público: uma Revisão Sistemática	34
Alexandre F. Mendonça, Fernanda M. R. Alencar	
Uso da Comunicação Não Violenta na Análise de Cenários de Conflitos em Engenharia de Requisitos	48
Aline F. Barbosa, Maria Lencastre	
Relacionando requisitos de software e competências de recursos humanos através de modelos organizacionais: uma abordagem visando o alinhamento organizacional	65
Henrique Prado de Sá Sousa, Eduardo Kinder Almentero, Julio Cesar Sampaio do Prado Leite	
Requirements Communication in Safety-Critical Systems: na extended literature overview	80
Jessyka Vilela, Jaelson Castro, Luiz Martins, Tony Gorschek, Camilo Almendra	
Desafios e Práticas da Engenharia de Requisitos no Contexto de Fábrica de Software com foco na Documentação e Gestão do Conhecimento.	98
Luana Souza, Erica Miranda, Márcia Lucena, Apuena Gomes	

CADERNOS DO IME

Série Informática

Volume 42

Publicação do Instituto de Matemática e Estatística da Universidade do Estado do Rio de Janeiro. Periodicidade semestral, com circulação em junho e dezembro.

Ruy Garcia Marques
Reitor

Maria Georgina Muniz Washington
Vice-Reitora

Luis Antonio Campinho Pereira da Mota
Diretor do Centro de Tecnologia e Ciências

Geraldo Magela da Silva
Diretor do Instituto de Matemática e Estatística

Sérgio Luiz Silva
Vice-Diretor do Instituto de Matemática e Estatística

Normalização, divulgação e distribuição:
Biblioteca do Centro de Ciências de Tecnologia A (CTC/A) da rede
Sirius de Biblioteca da UERJ – ctca@uerj.br

Organização e Edição do Volume 40:
Vera Maria B. Werneck
Fabiano de Souza Oliveira
Igor Machado Coelho

Correspondência:
Universidade do Estado do Rio de Janeiro
Instituto de Matemática e Estatística
Editores do Cadernos do IME - Série Informática
Rua São Francisco Xavier, 524 - Pavilhão Reitor João Lyra Filho,
6º andar, sala 6019 B
Maracanã - 20550-900 – Rio de Janeiro, RJ

Telefax: +55 21 2334-0144
e-mail: cadernos_inf@ime.uerj.br
site: <http://www.e-publicacoes.uerj.br/index.php/cadinf/>

Os artigos enviados para publicação deverão ser inéditos, com exceção de resumos ou teses, são de responsabilidade de seus autores, e não refletem, necessariamente, a opinião do IME. Sua reprodução é livre, em qualquer outro veículo de comunicação, desde que citada a fonte.

Produção e editoração gráfica:
Vera Maria B. Werneck
Fabiano de Souza Oliveira
Igor Machado Coelho

Julho 2019
ISSN 1413-9014
E-ISSN 2317-2193

Organizadores:

Vera Maria B. Werneck
Maria Clícia Stelling de Castro
Fabiano de Souza Oliveira

Contato: cadernos_inf@ime.uej.br

CADERNOS DO IME

Série Informática

Volume 42

Implementação e Avaliação de Co-Processadores para BlackScholes em FPGA	7
Walter Sousa, Marcelle Engle Carvalho Sacramento, Alexandre S. Nery, Maria Clícia S. Castro	
Predição de Processos com Dados não Estruturados Heterogêneos	18
Matheus Augusto de Oliveira, Flavia Maria Santoro, Kate Revoredo, Rosa Maria E. Moreira da Costa.	
Métricas de Tamanho de Software com Métodos Ágeis no Setor Público: um Survey	34
Alexandre F. Mendonça, Fernanda M. R. Alencar	
Uso da Comunicação Não Violenta na Análise de Cenários de Conflitos em Engenharia de Requisitos	48
Aline F. Barbosa, Maria Lencastre	
Relacionando requisitos de software e competências de recursos humanos através de modelos organizacionais: uma abordagem visando o alinhamento organizacional	65
Henrique Prado de Sá Sousa, Eduardo Kinder Almentero, Julio Cesar Sampaio do Prado Leite	
Requirements Communication in Safety-Critical Systems: na extended literature overview	80
Jessyka Vilela, Jaelson Castro, Luiz Martins, Tony Gorschek, Camilo Almendra	
Desafios e Práticas da Engenharia de Requisitos no Contexto de Fábrica de Software com foco na Documentação e Gestão do Conhecimento.	98
Luana Souza, Erica Miranda, Márcia Lucena, Apuena Gomes	

Nota dos Editores

Este é o primeiro volume lançado em 2019 e é uma edição onde publicamos 2 Artigos submetidos à Revista e os cinco melhores artigos do Workshop de Engenharia de Requisitos (WER 2019) realizado no início de agosto de 2019 na Universidade Politécnica de Pernambuco, Recife. Após o período com problemas estruturais relacionados à manutenção física da revista (crise no estado do Rio de Janeiro e greve na UERJ), impactando a capacidade da revista em manter o fluxo de publicações no Cadernos do IME Série Informática. Agradecemos aos nossos revisores e à Comissão Geral e de Programa do WER 2019. Temos convicção de que, bons ventos virão a nosso favor, o que se demonstra na qualidade dos artigos da presente edição.

Julho de 2019

Conselho Editorial

Alexandre Sztajnberg
Ana Leticia Luboc
Antônio de Pádua A. Oliveira
Fabiano de Souza Oliveira
Guilherme Lucio Abelha Mota
Igor Machado Coelho

Maria Clicia Stelling de Castro
Marinilza Bruno de Carvalho
Neide Santos
Rosa Maria Moreira da Costa
Vera Maria Benjamim Werneck

Revisores *ad hoc* internos

Alexandre Sztajnberg
Francisco Santanna
Marcelo Schots de Oliveira

Rosa Maria Moreira da Costa
Clicia Stelling de Castro
Vera Maria Benjamim Werneck

Revisores *ad hoc* externos

Eduardo Kinder Almenteiro (UFRRJ)
Fernanda Claudia Alves Campos (UFJF)
Igor Machado Coelho (UFF)

Program Chairs do WER 2019:

Fernanda Alencar, Universidade Federal de Pernambuco, Brazil
Henrique Prado de Sá Sousa, Universidade Federal Rural do Rio de Janeiro, Brazil
Marcela Ridao, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

Program Committee do WER 2019:

Alejandra Ruiz, Tecnalia, Spain
Alejandro Maté, University of Alicante, Spain
Alejandro Oliveros, Universidad Nacional de Tres de Febrero, Argentina
Alicia Martinez, Centro Nacional de Investigación y Desarrollo Tecnológico, Mexico
Amador Durán Toro, University of Seville, Spain
Andrea Zisman, The Open University, UK
Beatriz Marín, Universidad Diego Portales, Chile
Carina Alves, Universidade Federal de Pernambuco, Brazil
Carla Silva, Universidade Federal de Pernambuco, Brazil
Carme Quer, Universitat Politècnica de Catalunya, Spain
Clara Ayora, Treelogic, Spain
Cristina Palomares, Universitat Politècnica de Catalunya, Spain
Edgar Sarmiento, Pontifícia Universidade Católica do Rio de Janeiro, Brazil
Eduardo Almentero, Universidade Federal Rural do Rio de Janeiro, Brazil
Fabiane Benitti, Universidade Federal de Santa Catarina, Brazil
Fernanda Alencar, Universidade Federal de Pernambuco, Brazil

Gilberto Cysneiros Filho, Universidade Federal Rural de Pernambuco, Brazil
Giovanni Giachetti, Universidad Tecnológica de Chile, Chile
Gladys Kaplan, Universidad Nacional de La Matanza, Argentina
Graciela Hadad, Universidad Nacional del Oeste, Argentina
Gustavo Rossi, Universidad Nacional de La Plata, Argentina
Henrique Prado de Sá Sousa, Universidade Federal Rural do Rio de Janeiro, Brazil (PC-chair)
Hernan Astudillo, Universidad Tecnica Federico Santa María, Chile
Isabel Brito, Instituto Politécnico de Beja, Portugal
Jaelson Castro, Universidade Federal de Pernambuco, Brazil
João Araújo, Universidade Nova de Lisboa, Portugal
João Pimentel, Universidade Federal Rural de Pernambuco, Brazil
Jorge Barreiros, Universidade NOVA de Lisboa, Portugal
Jorge Horacio Doorn, Universidad del Oeste, Argentina
Jose Luis de La Vara, Carlos III University of Madrid, Spain
Juan Sanchez, Valencia University of Technology, Spain
Juan Pablo Carvallo, Universidad del Azuay, Spain
Julio Leite, Pontificia Universidade Católica do Rio de Janeiro, Brazil
Leandro Antonelli, Universidad Nacional de La Plata, Argentina
Lidia Lopez, Universitat Politècnica de Catalunya, Spain
Luciana Zaina, Universidade Federal de São Carlos, Brazil
Luiz Eduardo Galvão Martins, Universidade Federal de São Paulo, Brazil
Luiz Marcio Cysneiros, University of York, Canada
Lyrene Silva, Universidade Federal do Rio Grande do Norte, Brazil
Marcela Ridao, Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina
Marcia Lucena, Universidade Federal do Rio Grande do Norte, Brazil
Maria Pinto-Albuquerque, Instituto Universitário de Lisboa, Portugal
Maria Lencastre, Universidade de Pernambuco, Brazil
Mariel Ale, Universidad Nacional de Tres de Febrero, Argentina
Nathalie Aquino, Universidad Católica “Nuestra Señora de la Asunción”, Paraguay
Nelly Condori-Fernández, Vrije Universiteit Amsterdam, Netherlands
Oscar Pastor, Universitat Politècnica de València, Spain
Oscar Dieste, Universidad Politécnica de Madrid, Spain
Pedro Valderas, Universitat Politècnica de València, Spain
Renata Guizzardi, Universidade Federal do Espírito Santo, Brazil
Ricardo Ramos, Universidade Federal do Vale do São Francisco, Brazil
Roberta Fagundes, Universidade de Pernambuco, Brazil
Roberto Avila Paldes, Centro Universitário de Brasília, Brazil
Robson Fidalgo, Universidade Federal de Pernambuco, Brazil
Roel Wieringa, University of Twente, Netherlands
Roxana Portugal, Pontificia Universidade Católica do Rio de Janeiro, Brazil
Sabrina Marczak, Pontificia Universidade Católica do Rio Grande do Sul, Brazil
Vera Werneck, Universidade do Estado do Rio de Janeiro, Brazil
Victor Santander, Universidade Estadual do Oeste do Paraná, Brazil
Victor Souza, Universidade Federal do Espírito Santo, Brazil
Wylliams Santos, Universidade de Pernambuco, Brazil

Cadernos do IME – Série Informática

ISSN 1413-9014

E-ISSN 2317-2193

Apresentação

Para o Volume 42 dos Cadernos do IME – Série Informática aceitamos dois artigos submetidos e convidamos os cinco melhores artigos do Workshop de Engenharia de Requisitos (WER 2019) para publicarem seus trabalhos. Os artigos do WER 2019 foram revisados por pelos menos três revisores da comunidade de Engenharia de Requisitos.

O primeiro artigo, *Implementação e Avaliação de Co-Processadores para BlackScholes em FPGA*, implementa a fórmula BlackScholes em FPGA, avaliando sua eficiência com HLS (High level synthesis) e executando em uma placa Pynq-Z1. Os resultados são comparados com a execução em python no ARM presente na mesma placa.

O segundo artigo, *Predição de Processos com Dados não Estruturados Heterogêneos* avalia o impacto de características não estruturadas para predição do processo em uma empresa de prestação de serviços em TI reproduzindo e analisando os resultados de um framework que se baseia em modelos de texto para processamento de linguagem natural e algoritmos de aprendizagem de máquina.

O terceiro artigo, *Métricas de Tamanho de Software com Métodos Ágeis no Setor Público: um Survey*, é uma coletânea que identifica as métricas de tamanho de software em conjunto com métodos ágeis. Além disso, destaca o método com maior probabilidade de ser utilizados no setor público. A métrica COSMIC e os métodos SCRUM, XP e Kanban se mostraram como os mais promissores para o setor público.

O quarto artigo, *Uso da Comunicação Não Violenta na Análise de Cenários de Conflitos em Engenharia de Requisitos*, investiga os conflitos recorrentes nas equipes de desenvolvimento na Engenharia de Requisitos. Foi realizado um estudo qualitativo utilizando como método de coleta uma entrevista semiestruturada com profissionais de empresas distintas. Foram identificados e classificados cenários reais onde houve conflitos, baseado na análise de Comunicação Não Violenta (CVN).

O quinto artigo, *Relacionando Requisitos de Software e Competências de Recursos Humanos através de Modelos Organizacionais: uma Abordagem visando o Alinhamento Organizacional*, aborda as relações entre requisitos humanos e de software para melhor prospectar as formas de alinhamento das áreas organizacional e de recursos humanos. Diferentes requisitos de software são usados num exemplo de modelo organizacional, onde foi acrescentada a modelagem de competências humanas, que são requeridas nas organizações. A partir das informações de competências humanas e dos requisitos de software, foram identificadas e avaliadas as correlações visando encontrar elementos relevantes para auxiliar na decisão em ações de Tecnologia da Informação e de Recursos Humanos.

O sexto artigo, *Requirements Communication in Safety-Critical Systems: an extended literature overview*, investiga abordagens para melhorar os requisitos de comunicação no processo de Engenharia de Requisitos entre diferentes partes, no desenvolvimento de sistemas onde a segurança é crítica SCS (*Safety-Critical System*). Os autores realizaram um estudo de mapeamento sistemático e analisaram diferentes aspectos envolvidos, como desafios, domínio, atividades de requisitos, linguagens, ferramentas entre outros.

O sétimo artigo, *Desafios e Práticas da Engenharia de Requisitos no Contexto de Fábrica de Software com foco na Documentação e Gestão do Conhecimento* descreve as dificuldades na elaboração da documentação de requisitos e a gestão do conhecimento entre os engenheiros de requisitos nas fábricas de software motivaram a identificação de desafios, o uso das práticas para superá-los e o a criação de processo etnográfico organizacional. Dessa forma, este artigo investiga os desafios de duas equipes de engenheiros de requisitos de uma fábrica, que impactem na produção e na manutenção de uma documentação. Foi executada uma etnografia organizacional para levantamento de desafios, que após identificados foram agrupados em três categorias. Além disso, baseadas na literatura, foram sugeridas soluções práticas e factíveis a todos envolvidos.

Renovamos o convite aos pesquisadores para enviarem suas contribuições para os próximos números dos Cadernos do IME – Série Informática, para mantermos o mesmo padrão de qualidade obtido na presente edição.

Julho de 2018

Vera M. Werneck, Maria Clicia Stelling de Castro e Fabiano de S. Oliveira

Organizadores do Volume 42

cadernos_inf@ime.uerj.br

Implementação e Avaliação de Co-Processadores para BlackScholes em FPGA

Walter Sousa¹, Marcelle Engle Carvalho Sacramento¹,
Alexandre S. Nery¹, Maria Clícia S. Castro¹

¹Departamento de Informática e Ciência da Computação
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro (UERJ), Brasil

wjunior08,marcelle.engle@gmail.com,anery,clicia@ime.uerj.br

Abstract. *Option pricing is one of the main topics of the financial market. There are different models to perform this calculation, however BlackScholes is one of the most used nowadays. Due to this, the implementation of this algorithm shows an interesting study case for optimization. FPGAs (Field Programmable Gate Array) are boards commonly used in high performance computing, with promising results in several cases. Therefore, the main objective of this paper is to implement the Black Scholes formula in FPGA and evaluate its efficiency, using HLS (High level synthesis) and running in a Pynq-Z1 board. The results were compared with an execution in python on the ARM present on the board. The obtained results show an unexpected performance, and some of possible explanations to this fact.*

Resumo. *A precificação de opções é um dos principais temas do mercado financeiro. Existem diferentes modelos propostos para a realização desse cálculo, porém o BlackScholes é um dos mais usados atualmente. Por isso, a implementação do seu algoritmo se mostra um caso interessante de estudo de otimização. FPGAs (Field Programmable Gate Array) são placas comumente usadas em computação de alto desempenho, com resultados promissores em diversos casos. Assim, o principal objetivo desse trabalho é implementar a fórmula BlackScholes em FPGA e avaliar sua eficiência, utilizando HLS (High level synthesis) e executando em uma placa Pynq-Z1. O resultado foi comparado com a execução em python no ARM presente na mesma placa. Os resultados obtidos mostram um desempenho não esperado e algumas das possíveis explicações para este fato.*

1. Introdução

No mercado financeiro, opção é um meio de negociar o direito de compra ou venda de um ativo, em uma data futura por um preço prefixado. Ou seja, quem adquire uma opção está adquirindo o direito de comprar ou vender um ativo no futuro, por um preço predefinido que se manterá independente da flutuação desse ativo. Uma opção pode ser utilizada de diversos modos, como especulação, redução dos riscos de se ter um ativo, entre outros. [Ativo-objeto]. As características das opções estão relacionadas ao ativo ao qual se refere. Portanto, seu preço é derivado das variações do ativo objeto.

As opções são vendidas e revendidas por um preço conhecido como prêmio e, são divididas em categorias. As categorias mais comuns são as americanas e europeias. A principal diferença entre essas categorias é o direito de exercer a compra ou venda. Como dito em [Opções americanas e européias] na opção americana o titular pode exercer a compra ou venda até a data de expiração da opção. Já as européias, só podem ser executadas na data de expiração da opção.

Diversos modelos diferentes de precificação de opções têm sido usados no mercado financeiro. Atualmente, os modelos mais comuns são o Binomial e o BlackScholes. Ambos consideram diferentes variáveis no modelo de precificação, mas elas garantem resultados mais precisos. O modelo BlackScholes é relativamente simples e possui alta eficácia, assim ele é muito utilizado para encontrar preços justos [Precificação de Opções].

BlackScholes é um dos modelos mais comuns de precificação de ativos do mercado financeiro para as opções europeias. Fischer Black e Myron Scholes elaboraram o modelo, adaptando uma fórmula física para descrever a precificação de derivativos, no final dos anos 60. Eles publicaram em 1973 um artigo sobre o modelo, e tornaram-se ganhadores do prêmio Nobel, posteriormente, em 1997 [Precificação de Opções]. Segundo [Li 2012], este modelo é um dos mais importantes na teoria financeira quantitativa moderna.

Devido a grande importância dessa fórmula no mercado financeiro, este artigo busca mostrar uma forma mais eficiente para realizar os cálculos do modelo BlackScholes, através da utilização de FPGA.

Neste trabalho a implementação foi realizada com HLS (*High Language Synthesis*) para a programação da FPGA, permitindo o uso da linguagem C. Dessa forma, facilitando o entendimento, avaliação e a depuração do código. O ambiente de execução é composto pela FPGA PYNQ-Z1 que possui uma placa com a lógica da família Artix-7 e um processador ARM CortexTM-A9 Dual-Core, onde é possível executar o código python tanto para a chamada do *design*, quanto para comparação de desempenho.

Este artigo está organizado em oito seções. A Seção 2 discute alguns trabalhos relacionados que podem ser encontrados na literatura. A Seção 3 aborda a metodologia usada neste trabalho. As Seções 4 e 5 mostram uma visão geral sobre a fórmula BlackScholes e as FPGAs, respectivamente. As etapas de implementação do algoritmo estão na Seção 6 e os resultados obtidos são discutidos na Seção 7. Finalizamos com nossas conclusões e sugestões de trabalhos futuros na Seção 8.

2. Revisão da Literatura

Por volta dos anos de 2003 e 2004 na pesquisa [Hodjat and Verbauwhede 2004] alguns estudiosos utilizaram uma FPGA para implementar um algoritmo do padrão de criptografia avançado (AES - Advanced Encryption Standard). Este algoritmo é mais conhecido como Rijndael, e foi um padrão adotado pelo governo dos Estados Unidos. O seu maior obstáculo era a velocidade de encriptação e desencriptação dos dados. O processo era muito lento. Por isso, o algoritmo de criptografia AES foi implementado em FPGA, de formas diferentes. Uma das pesquisas, feita por Alireza Hodjat e Ingrid Verbauwhede, utilizou uma abordagem de *pipeline* para a implementação da encriptação, tendo um ren-

dimento de 21.54 Gbits/s. Este resultado é superior aos algoritmos implementados anteriormente sem *pipeline*, que obtiveram rendimento de 17.8Gbits/s e 18.56Gbit/s.

F. Rodríguez-Henríquez, N. A. Saqib e A. Díaz-Pérez realizaram outra pesquisa similar [Rodriguez-Henriquez et al. 2003], que também utilizou a abordagem de *pipeline*. Porém, para um algoritmo de encriptação e descriptação de dados. Eles realizaram algumas melhorias no desempenho no algoritmo e depois implementaram em FPGA. Foi obtido, então, um rendimento de 4.2Gbits/s, também superior aos algoritmos anteriores de 3.2Gbits/s e 2.2Gbits/s.

Mais próximo desse trabalho temos o artigo [Li 2012] Achieving Superior Performance on Black-Scholes Valuation Computing using Intel® Xeon Phi™ Coprocessors. Escrito por Shuo Li, funcionário da intel, que implementou o Black-Scholes no coprocessador Xeon Phi. Ele compara seu desempenho com as plataformas mais conhecidas, e outros tipos de aceleração, como a paralelização. O resultado mostrou que o coprocessador Xeon-Phi é 3.33 vezes mais rápido em comparação a implementação do BlackScholes totalmente paralelizada no Xeon [Li 2012]. A Figura 1 mostra os *speedups* de obtidos com os coprocessadores Intel Xeon e Xeon Phi.

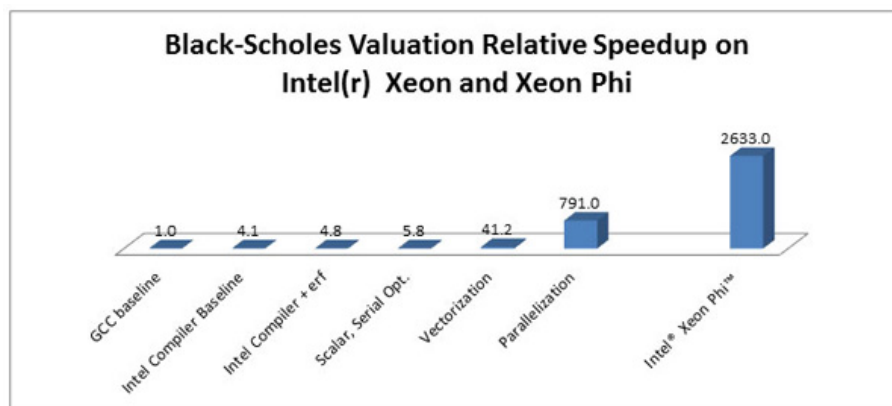


Figura 1. Gráfico dos *speedups* de comparação do processamento do algoritmo BlackScholes nos coprocessadores Intel Xeon e Xeon Phi.

Em [Eriko Nurvitadhi and Marr 2016] foi realizada uma avaliação em relação a melhoria da eficiência de algoritmos de redes neurais binarizadas (*BNNs - Binarized neural networks*) por meio de aceleração de hardware. O acelerador foi implementado no Aria 10 FPGA e no ASIC de 14 nm. Para a comparação dos resultados, foi utilizado também o software otimizado em uma CPU multi-core de alto desempenho e GPU para servidores na nuvem. Pelos resultados obtidos foi concluído que a FPGA se mostra mais eficiente que a CPU e GPU. Porém, A FPGA não é mais eficiente que a implementação no ASIC. Mesmo assim, foi observado que a FPGA é uma solução atrativa, proporcionando uma melhora de eficiência superior, sem ter que se prender a uma solução ASIC fixa.

3. Metodologia

Neste trabalho, para implementar e avaliar o desempenho do modelo BlackScholes em FPGA é utilizado o software Vivado HLS da Xilinx (<https://www.xilinx.com>). O algoritmo foi primeiro adaptado para a linguagem C++, de modo a utilizar as bibliotecas disponibilizadas pela Xilinx, específicas para suas FPGAs.

O algoritmo também foi implementado em python 3 rodando sobre o PYNQ linux, que é baseado no Ubuntu 18.04, utilizando o ARM presente na placa PYNQ-Z1. Essa execução foi utilizada para comparação dos dados de saída e dos tempos obtidos na execução via FPGA.

As etapas do trabalho consistem das otimizações na implementação, da síntese, e da comparação entre soluções já existentes e as implementadas nesse trabalho.

4. BlackScholes

Atualmente muito conhecido no mercado financeiro, o modelo BlackScholes tem como principal proposta a precificação de opções. Para isso, são utilizadas apenas as seguintes variáveis conhecidas [Precificação de Opções]:

$$d1 = \frac{\ln(\frac{s}{x}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \quad (1)$$

$$d2 = d1 - \sigma\sqrt{T} \quad (2)$$

Opção de compra:

$$c = S.CND(d1) - X.e^{-rt}.CND(d2) \quad (3)$$

Opção de venda:

$$p = X.e^{-rt}.CND(-d2) - S.CND(-d1) \quad (4)$$

onde as variáveis têm os seguintes significados:

- σ = Volatilidade, é uma medida que determina a variação do preço de um ativo;
- r = Taxa de juros livre de risco (SELIC no Brasil);
- T = Tempo restante para o exercício da opção em anos;
- S = Preço atual do ativo que a opção se refere;
- X = Preço de exercício, valor que será pago pelo ativo caso a opção seja exercida;
- CND = Função de distribuição cumulativa normal.

O modelo é baseado nas seguintes hipóteses [BlackScholes]:

- Os preços das ações se comportam correspondentemente a um modelo lognormal com desvio padrão e média constante;
- Não existem custos de transação;
- Não existe arbitragem possível;
- Contratos são divisíveis;
- São contínuas as transações de títulos e ações;
- A taxa livre de risco é a mesma para todos os investidores;
- A taxa de juros livre de risco no curto prazo é constante.

A fórmula do BlackScholes propõe um cálculo diferente de acordo com o seu tipo da opção, que pode ser de compra ou de venda.

Call é como as opções de compra são denominadas. Elas conferem ao titular o direito de comprar um ativo pelo valor determinado no contrato da opção, garantindo o poder de compra mesmo que o ativo tenha valorizado no momento de exercer

a opção [Opções de Compra]. Já as opções de venda, chamadas de *Put*, permitem que o titular tenha o direito de vender um ativo específico pelo preço presente no contrato da opção [Opções de Venda].

A fórmula BlackScholes mostra-se eficiente no suporte a tomada de decisões referentes a negociação de opções financeiras. Por isso, é pertinente melhorar a velocidade do seu cálculo. Pouco explorada para esse fim, a FPGA, que está descrita na Seção 5, demonstra um bom potencial de aceleração do cálculo da fórmula BlackScholes, e melhorar, assim, as decisões nas negociações financeiras.

5. FPGA

Uma FPGA (Field Programmable Gate Array) é um *chip* baseado numa matriz de blocos lógicos configuráveis (CLBs). Lançada pela Xilinx Inc., as FPGAs podem ser reprogramadas, mesmo depois de serem manufaturadas para assumirem formas de diferentes circuitos [Field-programmable gate array]. Esses *chips* são compostos por blocos lógicos, blocos de entrada e saída e chaves de interconexão, como pode ser visto na Figura 2. Cada um desses blocos está descrito a seguir [FPGA Fundamentals]:

- **CLBs (Configuration Logical Blocks):** blocos lógicos configuráveis que são a unidade lógica básica de uma FPGA, composto pela reunião de *flip-flops* e *lookup tables* (LUTs);
- **IOB (Input/Output Block):** *buffers* onde podem ser configurados registros de entrada e saída, possibilitando o acesso ao mundo externo;
- **Programmable Interconnects:** são trilhas usadas para conectar as entradas e saídas dos blocos lógicos.

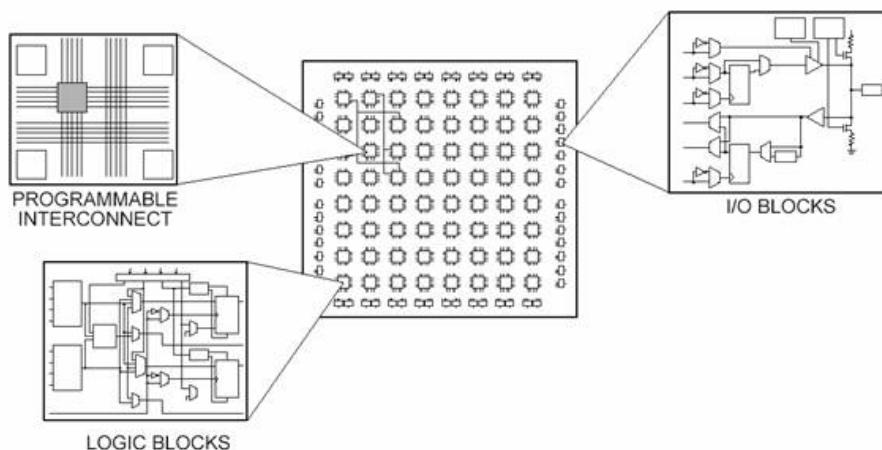


Figura 2. Partes principais que constituem uma FPGA genérica.

Uma FPGA possui uma matriz de portas lógicas reconfiguráveis. Quando a FPGA é configurada, é criado um circuito interno. Este circuito conecta as portas lógicas de forma a criar uma implementação em hardware do aplicativo de software. Diferentemente dos processadores, as FPGAs utilizam um hardware dedicado para o processamento lógico.

Os recursos internos de um *chip* FPGA consistem em uma matriz de blocos lógicos configuráveis (CLBs) cercados por uma periferia de blocos de entrada e saída. Os

sinais são roteados dentro da matriz FPGA por *switches* de interconexão programáveis e rotas de fios.

5.1. Níveis de abstração

Assim como nos computadores, as FPGAs podem ser programadas em diversos níveis de abstração.

Um dos níveis mais comuns é o nível RTL (*Register Transfer Level*), onde se usam linguagens de descrição de hardware, como VHDL e Verilog. Existe também o nível do algoritmo, onde se programa utilizando linguagens de alto nível, como C ou C++. Este nível é conhecido como síntese de alto nível (*High-Level Synthesis*) e facilita a programação das FPGAs.

Neste trabalho foi utilizado o software Vivado HLS (Xilinx), para que a aplicação desenvolvida em linguagem de alto nível C++ fosse compilada para o nível RTL. Sendo, depois, traduzida para a linguagem usada HDL (*Hardware Description Language*).

5.2. Placa FPGA PYNQ-Z1

Os resultados expostos nesse trabalho foram obtidos com a placa PYNQ-Z1 (Figura 3). Essa placa possui um processador ARM Cortex™-A9 Dual-Core 650MHz que inclui um servidor *web* que hospeda o ambiente de *design* do Notebook Jupyter, pacotes e núcleo IPython, linux e biblioteca de hardware e API para FPGA. Possui também um controlador de memória DDR3 com 8 canais DMA e 4 porta AXI3 de alta performance. Ela foi projetada para ser usada com PYNQ (Python Productivity for Zynq), um *framework open-source* que facilita os projetistas de sistemas embarcados explorarem os benefícios exclusivos dos APSoCs (All Programmable System-on-Chip) em seus aplicativos, sem ter que usar ferramentas de projeto no estilo ASIC (Application Specific Integrated Circuits) para projetar circuitos lógicos programáveis. [Digilent PYNQ-Z1]

O protocolo de comunicação AXI (*Advanced Extensible Interface*) [Xilinx 2012] viabiliza a troca de informações entre o processador e a área programável. Esse protocolo foi definido pela AMBA (*Advanced Microcontroller Bus Architecture*). Ele é um padrão aberto de especificação de conexão de *chip* para conexão e gerenciamento de blocos funcionais. Suas interfaces são: AXI4-Lite, AXI4 (ou AXI-Full) e Axi-Stream [Xilinx 2011]. A interface mais simples é a AXI4-Lite, geralmente ela é usada mais para sinais de controle e um fluxo de dados pequeno. Já a interface AXI-Full é um pouco mais complexa, sendo utilizada para uma maior transferência de dados, possui rajadas maiores, tendo um limite de 256 dados de 32 bits por envio, no máximo. A interface AXI-Stream é usada para grandes volumes de dados contínuos, sem limite de rajadas curtas.

6. Implementação

6.1. Otimizações

O pseudocódigo da função CND original, que implementa a função distribuição acumulada, pode ser visto no Algoritmo1. Nele é possível verificar que são utilizadas 4 potências da variável K , que são as potências 2, 3, 4 e 5. Para reduzir o tempo do cálculo das potências é possível reutilizar os resultados de uma potência anterior para calcular as demais. O Algoritmo2 é uma versão otimizada da versão original. Ela gerou uma melhoria de 0,4ns (redução de 6,141ns para 5,740ns) na estimativa de tempo entregue pela síntese do HLS.

```

entrada: X
saída : w
1 a1 = 0.31938153;
2 a2 = -0.356563782;
3 a3 = 1.781477937;
4 a4 = -1.821255978;
5 a5 = 1.330274429;
6 L = abs(X);
7 K = 1.0 / (1.0 + 0.2316419 * L); w = 1.0 - 1.0 / sqrt(2 * Pi) * exp(-L * L / 2) *
  (a1 * K + a2 * pow(K,2) + a3 * pow(K,3) + a4 * pow(K,4) + a5 *
  pow(K,5));
8 se X < 0 então
9   | w = 1 - w;
10 fim

```

Algoritmo 1: Pseudocódigo a função CND original.

```

entrada: X
saída : w
1 a1 = 0.31938153;
2 a2 = -0.356563782;
3 a3 = 1.781477937;
4 a4 = -1.821255978;
5 a5 = 1.330274429;
6 L = abs(X);
7 K = 1.0 / (1.0 + 0.2316419 * L);
8 k2 = K*K;
9 k3 = K*k2;
10 k4 = k2*k2;
11 k5 = k4*K;
12 w = 1.0 - 1.0 / sqrt(2 * Pi) * exp(-L * L / 2) * (a1 * K + a2 * k2 + a3 * k3 + a4
  * k4 + a5 * k5);
13 se X < 0 então
14   | w = 1 - w;
15 fim

```

Algoritmo 2: Pseudocódigo a função CND otimizado.

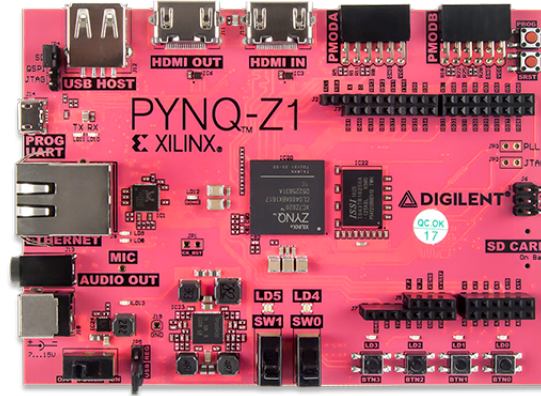


Figura 3. Placa PYNQ-Z1.

6.2. Síntese

Para que fosse feita a síntese do código em C para uma linguagem de descrição de hardware foram necessárias poucas modificações:

- Mudança da biblioteca `math.h` para `hls_math.h`;
- Adição dos pragmas para indicar os tipos de interfaces usadas para entrada e saída.

A síntese desse código teve como resultado um *design* com tempo de execução de aproximadamente foi de 5,740 ns.

O código em linguagem de descrição de hardware é integrado a outros circuitos, como o controle de *clock*, que garantirão o funcionamento da entrada e saída, criando um *design* que pode ser carregado na FPGA, como pode ser visto na Figura 4.

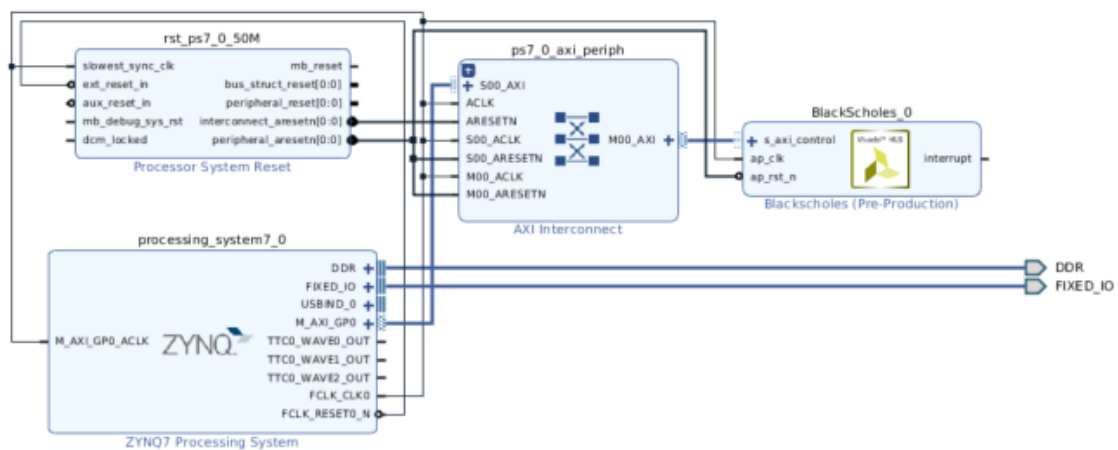


Figura 4. Visão geral da arquitetura.

7. Análise dos Resultados

Foram feitos experimentos com um conjunto de 1.000.000 entradas na FPGA com dados do PARSEC [Bienia 2011]. A Tabela 1 apresenta os resultados para 10 execuções do algoritmo implementado somente em Python e utilizando a FPGA, o tempo foi contabilizado em segundos e não contempla a leitura dos dados.

Tabela 1. Resultados obtidos para 1 Milhão de opções distintas com dados do PARSEC.

Execução	Python	FPGA
1	55,44	328,04
3	55,59	328,09
4	55,39	328,09
5	55,50	328,01
6	55,57	328,08
7	55,55	327,96
8	55,52	328,01
9	55,55	328,11
10	55,48	328,22
Média	55,52	328,08
Desvio Padrão	0,06	0,59

O programa Vivado HLS, ao final da síntese faz uma estimativa de tempo para uma execução no design feito. O tempo estimado para o design do algoritmo do BlackScholes foi de 5,740 ns para cada dado executado, com uma imprecisão de 0,75 ns.

A média dos valores da execução aferidos diretamente em python foi de 55,52 segundos para 1 milhão de execuções, com 0,06 segundos de desvio padrão. Deste modo, para cada opção o tempo foi em média foi de 55524 ns. Este valor é 9673 vezes maior que o estimado pelo HLS. Porém, a execução dos cálculos utilizando a FPGA teve, para o mesmo número de execuções, uma média de 328,08 segundos com desvio padrão de 0,59 segundos. Isto resulta em 328087 ns por opção. Sendo assim, a execução na FPGA se mostrou mais lenta que a realizada com python.

8. Conclusões e Trabalhos Futuros

Este artigo mostrou uma implementação do algoritmo BlackScholes utilizando uma FPGA.

Pelos resultados obtidos, pode-se concluir que o cálculo do algoritmo do BlackScholes implementado em uma FPGA PYNQ-Z1 utilizando a interface AXI-Lite não teve ganho de desempenho se comparado com o algoritmo em Python executado no ARM Cortex™-A9 presente na mesma placa. Assim, a solução implementada na FPGA se mostrou ineficiente. Comparando o tempo de cálculo estimado na síntese do HLS e a média obtida por execução, é possível afirmar que o tempo total para cada entrada é muito maior que o tempo de cálculo. Dessa forma, podemos considerar este fato como a maior causa possível para a baixa eficiência dos resultados obtidos. Isto é, o tempo de comunicação com a FPGA.

Pretendemos continuar este projeto utilizando outras formas de comunicação com a FPGA, de modo a reduzir o tempo de comunicação dos dados de entrada. Esta redução pode ser atingida implementando o mesmo algoritmo na interface AXI-Full utilizando rajada de dados maiores. Uma outra forma de obter um desempenho mais satisfatório seria implementar o cálculo da volatilidade. Assim, se reduziria o volume de dados de entrada no sistema para cada cálculo.

Como trabalhos futuros podemos também considerar uma implementação que envolvesse um *trading* de ações. Para lucrar com as operações de curto prazo no mercado financeiro, um *trader* é um investidor que busca o momento certo para realizar as operações de acordo com a volatilidade do mercado. A FPGA pode auxiliar e facilitar o trabalho de um *trader*, analisando e calculando o lucro de uma operação naquele determinado momento, de acordo com o preço do ativo em tempo real. Realizando assim, um trabalho rápido e preciso para a tomada de decisão do investidor.

Referências

- Ativo-objeto. Ativo-objeto. https://www.bussoladoinvestidor.com.br/abc_do_investidor/ativoobjeto/.
- Bienia, C. (2011). Benchmarking modern multiprocessors.
- BlackScholes. Black-Scholes: O Conhecido Modelo De Precificação De Opções. <https://www.sunoresearch.com.br/artigos/black-scholes/>.
- Digilent PYNQ-Z1. Digilent PYNQ-Z1. <https://www.xilinx.com/products/boards-and-kits/1-hydd4z.html>.
- Eriko Nurvitadhi, David Sheffield, J. S. A. M. G. V. and Marr, D. (2016). Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic.
- Field-programmable gate array. What is an FPGA? <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>.
- FPGA Fundamentals. FPGA Fundamentals. <http://www.ni.com/white-paper/55015/en/#toc1>.
- Hodjat, A. and Verbauwhede, I. (2004). A 21.54 gbits/s fully pipelined aes processor on fpga. *IEEE Xplore Digital Library*.
- Li, S. (2012). Achieving superior performance on black-scholes valuation computing using intel® xeon phi™ coprocessors. *Intel Software Developer Zone*.
- Opções americanas e européias. Opções americanas e européias: diferenças não se devem à posição geográfica. <https://www.infomoney.com.br/educacao/guias/noticia/380248/opcoes-americanas-europeias-diferencas-nao-devem-posicao-geografica>.
- Opções de Compra. O que é uma opção de compra (call)? <https://br.advfn.com/investimentos/opcoes/opcao-de-compra>.
- Opções de Venda. O que é uma opção de venda (put)? <https://br.advfn.com/investimentos/opcoes/opcao-de-venda>.

Precificação de Opções. Precificação de Opções: Modelo Black e Scholes. <https://blog.bussoladoinvestidor.com.br/precificacao-black-scholes/>.

Rodriguez-Henriquez, F., Saqib, N., and Diaz-Perez, A. (2003). 4.2 gbit/s single-chip fpga implementation of aes algorithm. *IEEE Xplore Digital Library*.

Xilinx (2011). Axi reference guide.

Predição de Processos com Dados não Estruturados Heterogêneos

Matheus Augusto de Oliveira¹, Flavia Maria Santoro², Kate Revoredo³, Rosa Maria E. Moreira da Costa²

¹ Universidade Federal do Estado do Rio de Janeiro – RJ– Brasil

² Universidade do Estado do Rio de Janeiro (UERJ) – RJ- Brasil

³ Universidade Federal do Rio de Janeiro – RJ– Brasil

matheuscruz27@gmail.com, {flavia, rcosta}@ime.uerj.br,
kate.revoredo@dcc.ufrj

Resumo. *Devido ao volume de informações e ao grande número de atividades do dia-a-dia dentro das organizações, o interesse na área de Gestão de Processos de Negócios ou Business Process Management (BPM) apresentou um crescimento notável. Portanto, cresce a necessidade de documentação, controle e otimização de processos e fluxos de trabalho existentes. Neste contexto, o monitoramento de processos ou Business Process Monitoring ganha atenção prevendo algumas características específicas de uma instância do processo em execução com base no histórico da execução do processo (log) e nos dados associados aos seus eventos. Essas previsões podem tornar as decisões de um analista de negócios mais assertivas, já que prever uma certa característica de uma instância em execução em um ponto específico do fluxo de trabalho com certo grau de confiança permite a atenuação de um problema antes que ele aconteça. Desta forma, as ações deixam de ser reativas e se tornam proativas, já que não precisamos esperar que a execução termine para observar o resultado e reagir. Por outro lado, decidir sobre o conjunto de características a considerar para a previsão é uma tarefa difícil. Neste artigo, avaliamos o impacto de características não estruturadas para predição do processo em uma empresa de prestação de serviços em TI.*

Abstract. *Due to the volume of information and the large number of day-to-day activities within organizations, the interest in the Business Process Management (BPM) area presented a remarkable growth and therefore, it increases the need for documentation, control and optimization of the existing workflows. In this context, Business Process Monitoring gain attention predicting some specific characteristics of the running process instance based on the history of the process execution and the data associated to the events. These predictions can make the decisions of a business analyst more assertive, since predicting a certain characteristic of a running instance at some specific point of the workflow with a high degree of confidence, allows the mitigation of a problem before it happens. Thus, actions are no longer reactive and become proactive, since we do not need to wait for the execution to finish to observe the result and react. On the other hand, deciding on the set of characteristics to consider for the prediction is a demanding and sometimes not straight forward task. In this paper, we evaluate the impact of unstructured characteristics for prediction on an IT consulting company.*

1. Introdução

Gerenciamento de Processos de Negócio tem sido bastante discutido e está presente nas instituições, sejam elas empresas privadas ou do setor público. Em Sistemas de Informação, o foco está em técnicas relacionadas como a descoberta, melhoria e o auxílio a tomada de decisão em processos (Francisco e Santos, 2012) (Oliveira e Delbem, 2011). Com o aumento significativo do fluxo de dados e das informações a eles associadas, a gestão desses processos passa a apresentar um novo grau de complexidade. Como apontado por Oliveira e Bertucci (2003), o gerenciamento da informação tornou-se um instrumento estratégico necessário para controlar e auxiliar decisões, através de melhorias no fluxo da informação, do controle, análise e consolidação da informação para os usuários.

Por outro lado, através do uso correto desses novos elementos, a predição de resultados da execução de processos fica mais precisa a cada dia. Com o uso de bases de dados mais robustas e abrangentes, é possível analisar atributos associados à execução do processo, antes não considerados, que por sua vez, também exercem influência no resultado obtido. Porém, junto a esses benefícios, alguns desafios também surgem, como por exemplo, o alto consumo de tempo no processamento dos algoritmos usados para análise (Teinemaa et al., 2016).

Teinemaa *et al.* (2016) propõem um *framework* que faz uso de técnicas de mineração de textos associadas ao monitoramento preditivo, para prever possíveis resultados do processo de contrato de empréstimo de uma agência bancária. Ao coletar informações como a renda e bens em posse do contratante, ou informações pessoais, como idade, tempo e tipo de emprego, analisar e comparar ao histórico existente, provenientes de diferentes casos, é possível apontar futuros atrasos e não pagamentos por parte do cliente. Sendo assim, a agência poderia repensar o tipo de contrato a ser firmado ou negar o valor pedido pelo cliente.

Maggi *et al.* (2103) focam em como a predição pode ajudar na gestão dos processos de atendimento de um hospital, seja no diagnóstico de doenças ou na indicação do tratamento adequado para seus pacientes, com base em casos similares do passado. Esses, dentre outros casos, conseguem exemplificar o quão este tema é abrangente e relevante.

O objetivo deste artigo é reproduzir e analisar os resultados do *framework* de Teinemaa *et al.* (2016) para predição das instâncias do processo de resolução de chamados de uma empresa de consultoria em tecnologia. Mais especificamente, prever se um caso em execução ultrapassará ou não o tempo de resolução estabelecido pelas regras de contrato da empresa com seus clientes. A principal contribuição é a utilização de texto não estruturado heterogêneo, que foi uma limitação deixada em aberto por Teinemaa *et al.* (2016).

O elemento chave (atributo que se deseja prever) é a variável que aponta se a instância do processo teve ou não seu acordo de nível de serviço ultrapassado ao final da sua execução. Para isso, parte do registro dos eventos dos casos executados e seus atributos estruturados e não estruturados foram organizados e utilizados como entrada para o treinamento e teste das funções existentes no *framework*. Assumindo-se as configurações apontadas com melhor performance, por seus desenvolvedores, considerando os tipos de dados do processo e seu comportamento. Isso possibilita também, o reforço ou refutação das hipóteses levantadas anteriormente sobre seu funcionamento e aplicação, sendo a principal delas, a de que o uso de texto, combinado aos outros atributos do processo, melhora a eficiência da previsão, mesmo para casos em que essa informação seja altamente heterogênea.

O artigo está estruturado como nas seguintes seções. Na Seção 2 é apresentada a base teórica, conceitos e técnicas aplicadas. Na Seção 3 são discutidos os trabalhos relacionados. Na Seção 4 é descrita a aplicação do Método de Predição no cenário real escolhido. A Seção 5 conclui o artigo e destaca as contribuições da pesquisa, sugerindo perspectivas futuras.

2. Fundamentação Teórica

Nesta seção são brevemente descritos os conceitos de descoberta de conhecimento em base de dados (Seção 2.1), que é utilizado para aprender os modelos preditivos, monitoramento preditivo (Seção 2.2) e o *framework* utilizado como base desse trabalho (Seção 2.3).

2.1 Descoberta de Conhecimento em Base de Dados

Descoberta de Conhecimento em Base de Dados (DCBD) é definida como o processo de extrair informações das grandes bases de dados para descobrir conexões ocultas e prever tendências (Bishop, 2016). A Figura 1 ilustra o processo DCBD.

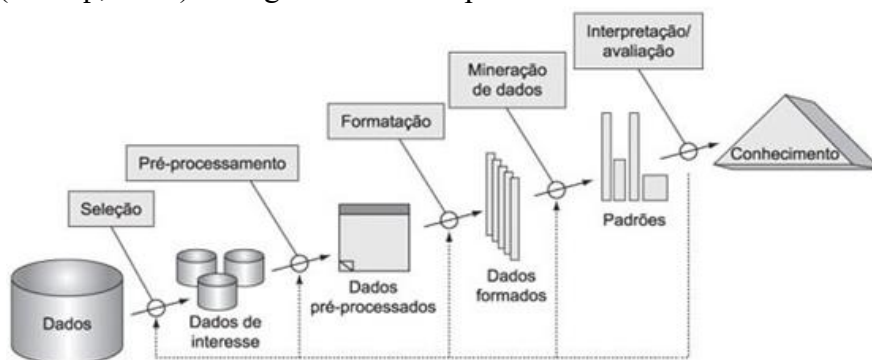


Figura 1 – Processo DCBD (Fayyad et al., 1996).

Geralmente, bases de dados apresentam informações extras que não influenciam no que se deseja analisar, podendo inclusive atrapalhar o processo. Por isso, o primeiro passo é retirar da base. O segundo passo é colocar essa base de dados na estrutura interpretada pelo método ou algoritmo que será utilizado na etapa seguinte, que é a mineração de dados. Nessa etapa os padrões são extraídos. Esses podem ser representados através de um modelo de classificação, que tem por objetivo aprender uma função que mapeia, ou classifica, um determinado dado em uma das várias classes predefinidas (Weiss e Kulikowski, 1991).

Neste artigo são utilizados dois modelos de classificação: Florestas Aleatórias (FA) Breiman (2001) e Regressão Logística (RL) (Walker e Duncan, 1967). Esses modelos apresentam bons resultados em várias configurações de problemas e se adaptam bem ao caso em que os dados são muito escassos (Teinmaa et al., 2016). Floresta Aleatória é baseado na junção de árvores de decisão geradas aleatoriamente. A classificação retornada pela floresta aleatória é a classe mais observada entre as retornadas pelas árvores. Regressão Logística (RL) por sua vez é uma função linear das características do dado, que tem como imagem dois possíveis valores de classe "0" e "1".

2.2 Monitoramento preditivo

O monitoramento de processos de negócio se refere à análise dos eventos produzidos durante a execução de um processo, com o intuito de avaliar o cumprimento de seus requisitos e sua

performance (Dumas *et al.*, 2013). Duas são as abordagens: (i) *offline*, onde usa-se as informações de registros das instâncias finalizadas e (ii) *online*, para analisar a performance do caso em execução.

A gestão de processos de negócio e sua eficácia no auxílio à tomada de decisão está diretamente relacionada ao quanto é possível reduzir o nível de incerteza do resultado de cada uma das instâncias executadas através da predição. Quanto mais rápido e com maior nível de confiabilidade se pode prever os resultados de um processo, melhores decisões são tomadas e mais rapidamente. Utilizando uma restrição do negócio (*business constraint*), definida como um requisito imposto à execução de um processo que separa o comportamento conforme com o não-conforme Pesic e Aalst (2003), o responsável pela gestão do processo irá tomar as medidas necessárias para que se alcance o resultado esperado.

A predição de processo pode ser feita de diversas maneiras. Dentre elas estão as técnicas de mineração de dados, que aprende um modelo preditivo à partir de dados históricos (*log* de eventos de instâncias finalizadas anteriormente). Com o monitoramento preditivo é possível antecipar diversas informações durante a execução do processo, como o tempo necessário para concluir uma tarefa, uma atividade ou operação específica. O grande ganho do monitoramento preditivo é o suporte para que as instituições consigam atingir suas metas previstas para cada um dos casos, podendo interromper a execução, caso percebam que ela sairá do esperado. Outros temas também são abordados, como a estimativa de tempo, seja de violação de prazo (*Service Level Agreement* - SLA) ou previsão do início e fim das tarefas, ou ainda, os casos utilizados para prever os riscos, como exemplificado por Conforti *et al.* (2016).

A Figura 2 ilustra as etapas de predição em processo de negócio. O processo é dividido em dois estágios. No primeiro estágio (*offline*) o modelo preditivo é aprendido. Para isso, são considerados como entrada o modelo de processo e os seus atributos e os dados históricos de execução do processo (*log* de eventos). O aprendizado do modelo preditivo é feito por algoritmos de mineração de dados. No segundo estágio (*online*), o modelo preditivo aprendido é utilizado para avaliar a instância do processo em execução.

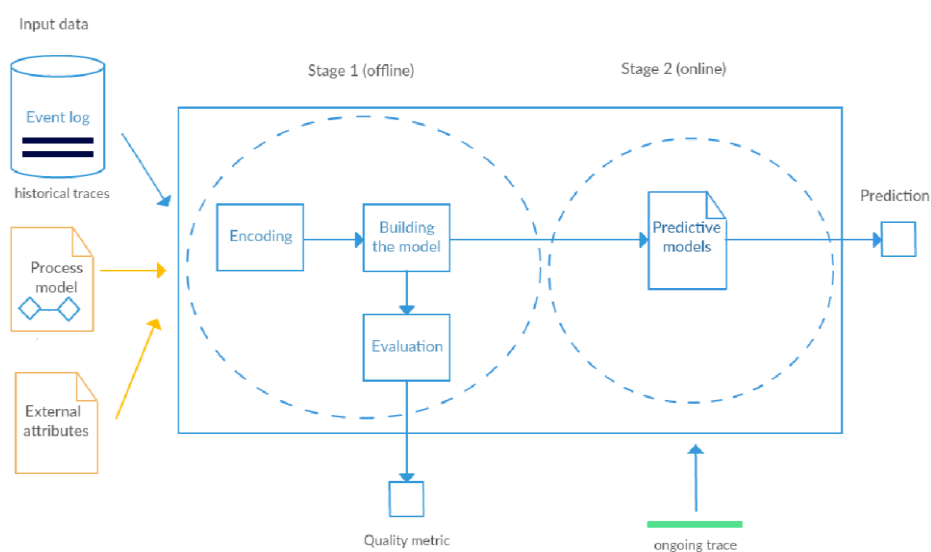


Figura 2 - Predição de processos (Conforti *et al.*, 2016).

Um dos desafios do monitoramento preditivo é a escolha das características consideradas para a construção do modelo preditivo. Teinemaa *et al.* (2016) discutiram o benefício da consideração de dados não estruturados combinados com dados estruturados. Na próxima seção detalhamos como o *framework* desenvolvido pelos autores combinou esses diferentes tipos de dados para a predição.

2.3 *Framework* para predição de processos

Teinemaa *et al.* (2016) analisaram como o uso de texto livre, tido como atributo não estruturado do processo e a sua combinação com os dados estruturados, pode contribuir para o monitoramento preditivo de processos de negócio. A proposta apresenta um *framework* capaz de lidar com *logs* de eventos com essas características e seus resultados são comparados às técnicas existentes. Para isso, técnicas de mineração de texto foram utilizadas para extrair informações relevantes de cada instância de processo enriquecendo o *log* de eventos.

Para confirmar se o resultado apresenta melhorias ou não, um estudo de caso foi feito em *logs* de eventos de dois processos reais de uma empresa financeira: (i) processo de recuperação de dívidas, cujo resultado é o reembolso parcial do valor negociado previamente ou o encaminhamento do caso para uma agência externa de cobrança e (ii) um processo de oferta de contratos em que o resultado é a assinatura ou não de um empréstimo do cliente em potencial. O objetivo então, é conhecer se o valor devido será pago no prazo definido, para o primeiro processo e se o cliente fechará um novo contrato, para o segundo. Em ambos os casos, essa previsão deve ser feita em tempo de execução do processo, tendo como entrada o histórico das ações realizadas e das mensagens trocadas até o dado momento.

A primeira etapa do *framework* é a construção de modelos e extração das características de texto. Ambos feitos com base nas mensagens associadas a cada um dos eventos no registro. Diferentes técnicas e parâmetros foram utilizadas para criar os modelos, visando verificar qual apresenta melhor resultado de acordo com as características específicas de cada *log*.

No segundo momento, as características extraídas dos documentos de texto são combinadas com os atributos estruturados, ambos adicionados a um vetor de tamanho fixo e com seus valores representados por números. Isso é feito para que os algoritmos de mineração de dados consigam interpretar suas informações e assim treinar os classificadores para previsão, que é a última etapa do *framework*.

Para estruturar o texto, os métodos utilizados foram o BoNG; Taxas de contagem de *log* com Naïve-bayes (NB) que é baseado no modelo BoNG, mas ponderado com as taxas de registro do NB (Allahyari *et al.*, 2017); Alocação latente de Dirichlet, onde o modelo de texto é representado por tópicos abordados pelos documentos; e Paragraph Vector (PV), onde não somente os termos, mas também suas sequências são extraídas para criar o modelo. Os dois últimos são detalhados por Blei (2003). BoNG (*bag-of-n-grams*) parte dos mesmos princípios do modelo *bag-of-words* ou saco-de-palavras, e tem o texto representado por uma coleção de palavras que independem da gramática. Esse grupo pode ser formado separando cada termo independentemente e mostrando a frequência com que eles aparecem. Como exemplo, a frase “Não posso falar agora, te ligo depois”, poderia ser representada por {“não”:1, “falar”:1, “agora”:1, “te”:1, “ligo”:1, “depois”:1}. Neste caso, porém, não é possível representar a ordem em que essas palavras estão dispostas no documento. O modelo BoNG soluciona esse problema, uma vez que a seleção das palavras é feita em grupos de *n*-palavras. A

representação do mesmo documento em grupos de 2 palavras ficaria, então, como a seguir {"não posso":1, "posso falar":1, "falar agora":1, ...}.

Como os documentos de texto não apresentam um padrão a ser seguido, são executados o tratamento e a limpeza de dados em cada um dos *logs*, visando manter somente informações relevantes para o contexto. Após essa etapa inicial, o *framework* é dividido em dois componentes. O primeiro, apresentado na Figura 3 é *off-line*. Ele realiza a estruturação dos dados, combinando a sequência de ambos os tipos de atributos dos casos no histórico para treinar os classificadores que, no segundo componente (*online*) fará uso desses dados para realizar as previsões dos casos em execução.

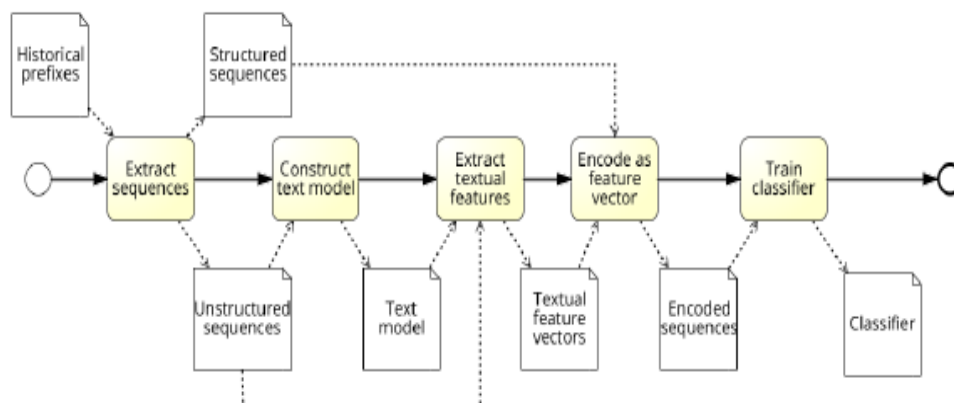


Figura 3 - Componente offline do *framework* (Teinmaa et al., 2016).

A outra parte é feita pelo componente *online* definido pelo *framework*, conforme a Figura 4. Nesse momento que é realizada a predição e classificação da instância em execução. Para sua execução, o grau de confiança mínimo precisa ser adicionado como variável de entrada. Quando executado para previsão do resultado de um caso em andamento, com k eventos realizados, se a probabilidade apontada pelo classificador for maior do que esse limite, o resultado é apontado como positivo. Enquanto esse requisito não for alcançado, o *framework* continua monitorando as próximas atividades. Se o evento observado for terminal, o resultado da previsão é classificado como negativo.

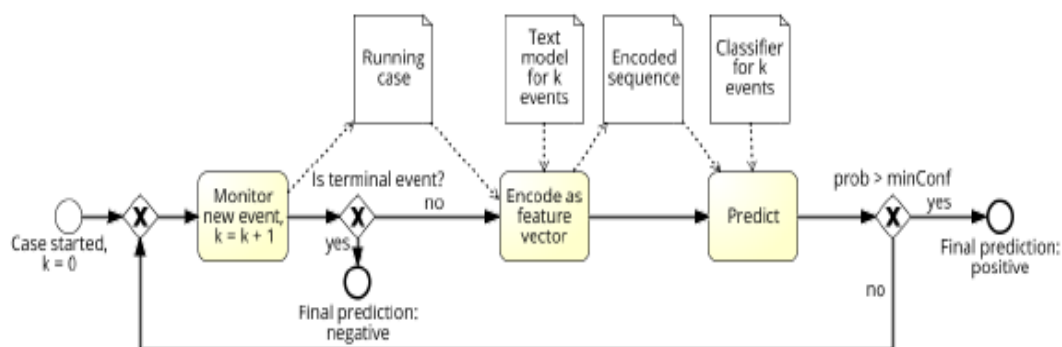


Figura 4 - Componente online do *framework* (Teinmaa et al., 2016).

O *framework* foi avaliado com relação a precisão, *recall* e *f-score* comparando um modelo que foi aprendido a partir de dados estruturados com um modelo aprendido a partir de dados

estruturados e não estruturados. Os *logs* de ambos os processos foram divididos aleatoriamente em duas partes. Uma para treinar os classificadores no componente *offline* e a segunda, menor, para testar no componente *online*.

Dois modelos foram considerados floresta aleatória e regressão logística. O primeiro apresentou um resultado mais satisfatório do que o segundo. Além disso, a avaliação do *framework* mostrou que o uso de informações extraídas de texto combinadas aos outros atributos do processo, melhora a previsão da resolução do caso em execução.

Uma das limitações apontadas por esse trabalho foi com relação aos dados não estruturados considerados. Eles eram textos associados ao processo, com uma formatação conhecida, sendo assim uma análise com relação a textos mais heterogêneos ficou em aberto.

3. Trabalhos relacionados

Esta seção apresenta brevemente outras aplicações do monitoramento preditivo em processos para a resolução de problemas. Áreas de nichos completamente diferentes podem fazer uso de técnicas e conceitos muito parecidos.

Maggi *et al.* (2013), por exemplo, combinam a descoberta de processos baseada no controle de fluxo de eventos com a baseada no fluxo de dados. Eles mostram como essa combinação pode ser usada para auxiliar nas decisões dos médicos durante uma análise clínica. Isso é feito com base nas informações de casos similares. Já De Leoni e Aalst (2013) analisaram o *log* de eventos de um processo para lidar com empréstimos solicitados pelos clientes de um instituto de crédito.

Yeshchenko *et al.* (2018) propuseram uma abordagem de integração de elemento de contexto externo aos processos de negócio em métodos de predição. Os autores desenvolveram uma técnica que associa os sentimentos sobre de notícias *online* ao *log* do processo para a tarefa de predição de tempo restante de execução de suas instâncias.

Também inserido no nicho das organizações, sejam privadas ou não, trabalhos são voltados para o monitoramento de riscos, como mostram Conforti *et al.* (2016) e Conforti *et al.* (2013). Outros possuem uma visão mais acadêmica, como o apresentado em Senderovich *et al.* (2016), que usa as redes de petri estocásticas generalizadas (GNSPs) com o objetivo de melhorar a acurácia das previsões.

4. Análise do Método de Predição em Cenário Real

Utilizando como base os resultados do estudo apresentado em Teinemaa *et al.*, (2016), esse artigo visa contribuir para a análise de como a combinação de dados estruturados e não estruturados pode auxiliar na eficácia do monitoramento preditivo para processos de negócio. Além disso, textos com estrutura livre foram considerados, estendendo a análise feita em Teinemaa *et al.*, (2016).

O *log* de eventos estudado contém registros dos *tickets* de um processo de resolução incidentes na área de Tecnologia da Informação (TI) e redes, de uma empresa de consultoria localizada no Rio de Janeiro. Essas informações foram registradas durante o ano de 2015 através do sistema de Gerenciamento de Serviços de Tecnologia de Informação (ITSM) utilizado pela empresa. Dentro desse *log* encontram-se informações de cada *ticket* e os atributos relacionados a cada um dos eventos executados durante o processo de resolução dos incidentes. Esses atributos podem representar informações do *ticket*, como tipo de

atendimento e a sua prioridade, ou dados relacionados às atividades executadas. No primeiro caso, as informações são coletadas no início da execução do processo e são mantidas durante a execução. Já no segundo, a cada atividade, a informação do atributo pode alterar.

O atributo utilizado para análise da influência do texto no processo é o corpo dos *e-mails* trocados pelos solicitantes do atendimento e os membros das equipes solucionadoras. Caso em determinado *ticket* ou *e-mails* não tenham sido trocados, então este atributo tem valor zero.

Um atributo extra é considerado para indicar se o processo foi executado dentro da SLA definida ou não. O atributo é denominado MissedSLA, e tem como possíveis valores “0” e “1”, respectivamente. Esses valores são obtidos através da comparação do tempo de execução com o prazo pré-estabelecido para as diferentes prioridades.

A seguir, estão detalhados o processo de negócio, o *log* estudado, as etapas de execução do método de predição e os resultados encontrados.

4.1 Processo utilizado

Como mencionado anteriormente, o *log* utilizado apresenta registros dos casos de um processo de tratamento de incidentes (também chamados de *tickets*) relacionados ao setor de tecnologia de informação encontrados pelos usuários da mesma empresa. O objetivo desse processo é executar a operação de resolução dos problemas de maneira eficaz e dentro do prazo estabelecido em contrato com seus clientes.

O processo se inicia quando um usuário de outro setor da empresa percebe algum problema que bloqueie ou atrapalhe seu trabalho e esteja diretamente relacionado com tecnologia. Os diferentes níveis de impacto definem a prioridade dos chamados, e consequentemente suas SLAs.

O processo de tratamento se inicia quando um colaborador da empresa abre um chamado através da ferramenta de gerenciamento de serviços de TI, adicionando informações iniciais como, por exemplo, qual o tipo de problema enfrentado. Esse *ticket* é, então, direcionado para a equipe de nível 1, que faz a triagem das informações e verifica qual equipe de suporte deve ser acionada para resolução do incidente.

Caso o problema possa ser resolvido pela própria equipe, o *ticket* continua na fila da equipe e as ações necessárias são tomadas diretamente por um de seus membros. Caso o problema seja causado por uma área mais específica, o atendente transfere o *ticket* para a equipe que possui capacidade de resolução. Esse *ticket* pode passar pelas demais equipes até sua resolução, caso o problema esteja relacionado a mais de uma área técnica.

Após as atividades consideradas suficientes pelo responsável, o solicitante é notificado com a proposta de resolução e pode confirmar se o problema inicial foi resolvido, encerrando as atividades para o caso, ou se ainda existem problemas a serem solucionados. Nesse caso, o *ticket* retorna para quem propôs a solução e permanece nesse ciclo até que a resolução atenda aos requisitos. Quanto menos vezes o chamado precisar voltar para a equipe solucionadora, melhor é o tempo de resolução. Sendo assim, o melhor caso é aquele em que a primeira solução é aceita pelo usuário.

4.2 Log de eventos

O *log* de eventos utilizado possui registros da execução de *tickets* do primeiro trimestre do ano de 2015. O arquivo contém informações referentes a 6.337 *tickets* resolvidos durante o

período e as atividades necessárias para resolução de cada um, divididos entre as 84 diferentes classificações de chamado.

Dentre os 6.337 *tickets*, 3.370 foram resolvidos dentro da SLA (SLAMissed=0) e 2.967 fora (SLAMissed=1). Em média, a quantidade necessária de passos para a resolução do problema é de 39. Pode-se observar também os extremos em que, no menor caso, esse número foi de somente 10 e no maior, que precisou de 357 até ser resolvido. A grande maioria dos casos não possui esse número acima de 50. Em uma análise direta, observa-se que quanto maior o número de atividades executadas, maior a probabilidade de ultrapassar a SLA.

Os atributos do *log* considerados para análise estão descritos na Tabela 1, assim como o tipo de dado armazenado.

Tabela 1 - Tipos de dados do *log*.

Nome do atributo	Descrição
CaseID	Número de identificação do <i>ticket</i> no registro.
EventID	Número que identifica e representa a ordem em que a atividade foi executada durante o processo.
EventName	Classe que identifica e classifica os diferentes tipos de atividades possíveis no processo.
Priority_ID	Número usado para identificar a prioridade, de acordo com o impacto de cada <i>ticket</i> .
ServiceType	Classe que indica o tipo de problema para o atendimento.
a_body	Corpo dos <i>e-mails</i> trocados durante as atividades de cada <i>ticket</i> .
SLAMissed	Classe que indica para cada caso, se o problema foi resolvido dentro do prazo estabelecido, de acordo com as prioridades.

A data de resolução e abertura dos incidentes não foram utilizadas, uma vez que a coluna “SLAMissed” indica se o caso foi resolvido ou não dentro do tempo esperado, não havendo necessidade de calcular o tempo total da carga de trabalho. Seguindo o objetivo deste artigo de verificar a combinação de dados estruturados e texto livre para a melhoria dos resultados de predição, somente o corpo do *e-mail* foi mantido para a análise. As informações de remetente e destinatário são consideradas dados semiestruturados.

Por possuir diversas informações não relevantes ao contexto do que é tratado durante a resolução dos *tickets*, foi realizada uma limpeza no texto. Apresentações e assinaturas, por exemplo, foram retiradas, assim como dados de telefone dos atendentes e diferentes *links* para o *site* da empresa. Além disso, um pré-processamento foi executado para remover as chamadas “palavras vazias”. Essas palavras são importantes para o entendimento do texto entre seus locutores, por serem utilizadas para definir gênero ou ligar as diferentes frases do texto, por exemplo. Mas, para o algoritmo de aprendizado não acrescentam muito valor ao que se é aprendido, uma vez que são palavras muito comuns. Portanto, aparecem com muita frequência nas mensagens. Alguns exemplos são ‘e’, ‘o’, ‘a’, ‘em’ e ‘no’. Além desse, outros tratamentos foram aplicados e estão detalhados na sequência.

4.3 Execução do método de predição

Uma vez que o estudo realiza uma comparação entre diferentes métodos para construção do modelo de texto e extração das características de maior impacto para o processo, o modelo

utilizado foi o *bag-of-words*. Ele apresenta uma melhor desenvoltura nos resultados, em comparação aos demais.

Como o objetivo de aprimorar o processo de predição, após o processamento do texto, um novo atributo foi adicionado, contendo a quantidade de palavras utilizadas durante as mensagens para cada *ticket*. Dessa forma, é possível analisar também como a quantidade de palavras presentes em cada *ticket* afeta a resolução do problema. A Figura 5 ilustra esse comparativo e é possível verificar uma redução no número de *tickets* dentro da SLA quando a quantidade de palavras passa de 1.900.

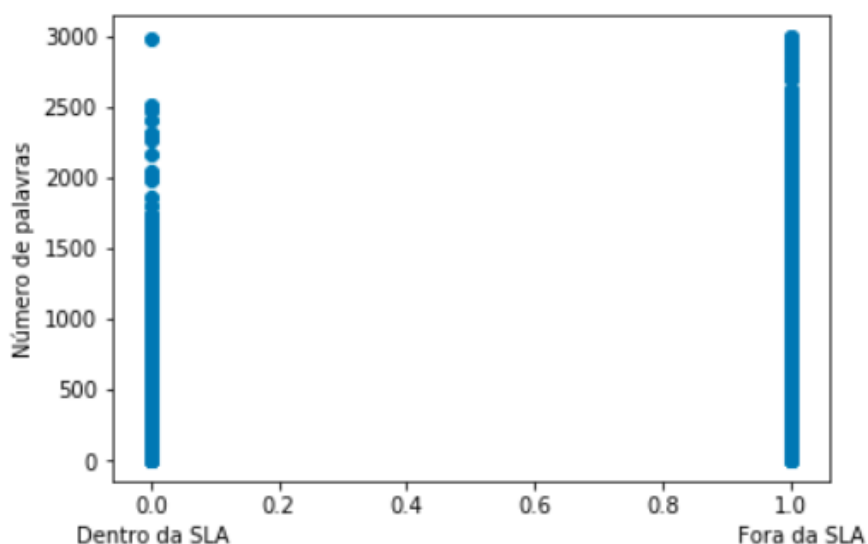


Figura 5 - Classificação por quantidade de palavras.

Essas são algumas das análises diretas que podem ser feitas observando os dados relativos às mensagens de texto trocadas durante um atendimento. Elas exemplificam bem como a troca de informações na forma de linguagem natural pode afetar o resultado de um processo e reforçam a necessidade desse e outros trabalhos sobre o tema. A seguir estão as etapas de execução até o treinamento e teste dos princípios abordados.

4.3.1 Preparação do log

Após o pré-processamento do texto e adição de um novo atributo, os valores nominais de cada atributo foram discretizados permitindo que os algoritmos de mineração escolhidos pudessem ser aplicados.

Após a preparação inicial do *log*, os próximos passos se deram através do processamento do texto e criação do vetor de características. O conteúdo do texto foi utilizado para extração desses vetores. Isso indica que o tamanho do vetor é dado de acordo com o tamanho do dicionário de palavras selecionado. Dessa forma, por se tratar de documentos que podem conter um grande número de palavras, as características extraídas irão apresentar um grande volume, criando vetores muito esparsos, dado que diversas palavras devem aparecer poucas vezes em textos diferentes. Isso reduz o desempenho, pois o algoritmo tem que percorrer cada um e aprender pouco com os vários valores nulos que estão representados.

Visando melhorar o desempenho de processamento e trazer informações relevantes ao conteúdo, uma limpeza e seleção de palavras foi realizada. Ela ocorreu da seguinte forma:

1. Extração do texto não relevante para a resolução dos incidentes: Normalmente, *e-mails* apresentam uma estrutura similar para cada tipo de usuário e empresa, como assinaturas,

sites web, telefones de contato e apresentações. Nessa etapa, o objetivo foi retirar esse tipo de informação, que normalmente não fazem parte do contexto descritivo do problema e nem das atividades realizadas durante o processo. Além de excluir informações que não agregam para a previsão, as retiradas desses termos também auxilia no desempenho do algoritmo, pois o número de palavras lidas é menor. No exemplo a seguir, pode-se ver um corpo do *e-mail* como seria registrado no *log*:

Bom dia!

*Não consigo enviar mensagens do meu e-mail corporativo.
Não está funcionando para receber novas mensagens também.*

Podem verificar, por favor?

*Att,
Nome Sobrenome
Analista de Marketing
Tel.:(xx) 0000-0000*

Na sequência, como o mesmo texto ficaria somente após o primeiro filtro das informações:

*Não consigo enviar mensagens do meu e-mail corporativo.
Não está funcionando para receber novas mensagens também.*

2. Retirada de pontuação: Para que seja possível passar a entonação e consequentemente, dar o sentido desejado ao que está escrito, a pontuação se faz presente nos *e-mails* trocados. Porém, para o modelo de texto a ser criado, esses caracteres não agregam valor, além de aumentarem o número de informação a ser computada. Nesse segundo passo, a tarefa foi retirar esses marcadores antes de separarmos as palavras dos textos.
3. Retirada das palavras vazias: Mais uma vez, para dar sentido a escrita diversas palavras são utilizadas, como conectores ou artigos para definição do gênero, e consequentemente não refletem o conteúdo da conversa. Durante essa etapa esses vocábulos foram retirados. A mensagem passaria a ser representada como a seguir:

*Não consigo enviar mensagens e-mail corporativo
Não funcionando receber novas mensagens*

4. Criação de *tokens*: Esta etapa é essencial para a criar o modelo de texto, pois é nesse momento que as palavras são separadas e adicionadas às posições no vetor. Como houve a retirada de pontuação, as palavras estão separadas através dos espaços em branco no texto. Para que o algoritmo não computasse separadamente as palavras somente por conter letras maiúsculas, todas foram normalizadas para minúsculo. No exemplo, um vetor x , contendo os *tokens* retirados do texto teria seus valores indicados por $x = \{'não', 'consigo', ..., 'novas', 'mensagens'\}$.
5. *Stemming*: O termo em inglês é utilizado em processamento de linguagem natural, e define a prática de se extrair o radical dos termos presentes no texto. Em outras palavras, as diferentes formas das palavras derivadas da mesma raiz, são todas reduzidas a ela. Para exemplificar, os vocábulos “apresentar”, “apresentação”, “apresentando”, seriam reduzidas à “apresent”, uma vez que somente os sufixos são diferentes. Como cada língua apresenta características específicas, foi utilizada a função *stemm* do Python para

português. Seguindo a nova etapa, as informações no vetor ficariam $x = \{\text{"não"}, \text{"conseg"}, \dots, \text{"funciona"}, \text{"receb"}, \text{"nova"}, \text{"mensagem"}\}$

Esse tratamento das informações contidas nas mensagens de texto foi importante para evitar conteúdo que interferisse no resultado negativamente. Além disso, ajudou a reduzir a quantidade de informação presente nos documentos de texto, melhorando a velocidade de processamento. O uso desses dados para extração do vetor de características e treinamento dos classificadores estão detalhados a seguir.

4.3.2 Extração dos vetores de características

Como cada uma das mensagens está representada por uma lista de palavras, ou *tokens*, o próximo passo foi convertê-las em um vetor de características. Dessa forma, ele pode ser utilizado como entrada pelos métodos de aprendizado de máquina, que não trabalham com texto bruto diretamente. Para isso, foi preciso criar um dicionário com as palavras a serem utilizadas. Como a redução do número de termos apresentou uma perda considerável de predição e baixo ganho no desempenho de processamento, as palavras presentes nos diferentes documentos foram mantidas.

Seguindo as etapas do modelo *bag-of-words*, foram contadas quantas vezes cada uma das palavras aparecem em cada documento. Isso quer dizer que cada uma delas corresponde à uma coluna da tabela e os valores em cada linha representam a quantidade de vezes que elas aparecem no documento associado a cada atividade. Sendo assim, atividades que não tiveram troca da mensagem tiveram seus valores iguais a zero.

4.3.3 Aplicação do TF-IDF

Após criar o vetor de características, o próximo passo foi normalizar os valores encontrados para cada termo. Existem palavras que podem aparecer diversas vezes nos documentos, mas que não adicionam informações tão relevantes para o contexto. Caso não sejam tratadas, essas palavras podem sobrepujar outras que não aparecem tantas vezes, mas que podem ser específicas sobre o domínio estudado, por exemplo.

Para que esses dados não dominassem o modelo com uma alta contagem e influenciassem a previsão de forma negativa, foi utilizada a técnica de pontuação TF-IDF. Essa abordagem compara a frequência que as palavras aparecem nos documentos separadamente com o quão raras são entre todos os outros. Nesse caso, as expressões frequentes no corpo do *e-mail* analisado, e em todos os outros documentos, têm sua pontuação penalizada. O efeito disso é que palavras distintas recebem um peso maior se comparadas àquelas que são muito utilizadas. A partir de então, os termos deixam de ser representados pela sua frequência no documento e passam a ser representados pelo seu valor ponderado.

Após essas etapas, o *log* está pronto para o aprendizado, já que possui seus valores representados por vetores de números de tamanho fixo. A combinação dos valores para as variáveis estruturadas e do texto processado de cada linha foi utilizada pelo modelo como treino e teste para prever o valor da classe associada como resultado. Em outras palavras, o valor de cada linha da coluna “SLAmissed” apresenta uma combinação de eventos, de atributos associados e de palavras presentes no corpo dos *e-mails*, que é utilizada para mapear o comportamento padrão através dos algoritmos de aprendizado de máquina.

4.3.4 Treinamento dos classificadores

Para treinar os classificadores, após toda a etapa de preparação, os registros do *log* foram separados em grupos diferentes, para que o conjunto de dados de treinamento fosse diferente do utilizado para teste e dessa forma, o resultado fosse o mais próximo possível do caso real.

Como linha de base, foram utilizados os resultados obtidos do treinamento e teste utilizando somente as informações disponíveis como dados estruturados. Assim, é possível comparar se a combinação dos diferentes atributos realmente apresenta uma melhora de resultado. Para comparação, outro teste foi realizado considerando somente as características extraídas do corpo dos *e-mails*. As Tabelas 2 e 3 mostram os resultados de precisão, *recall* e *f-score* quando considerando Floresta Aleatória e Regressão Logística.

Esses resultados mostram uma melhora em relação aos modelos aprendidos apenas considerando os dados estruturados. Esse comportamento é mais acentuado para os casos positivos.

Tabela 2 - Relatório de classificação - dados estruturados e não estruturados (FA).

SLAMissed	Dados estruturados			Dados estruturados & não estruturados		
	Precisão	<i>Recall</i>	<i>F-score</i>	Precisão	<i>Recall</i>	<i>F-score</i>
0	0.62	0.63	0.62	0.69	0.79	0.74
1	0.57	0.56	0.56	0.71	0.61	0.66
Media/Total	0.60	0.60	0.60	0.70	0.70	0.70

Tabela 3 - Relatório de classificação – dados estruturados (RL)

SLA Missed	Dados Estruturados		Dados estruturados & Não estruturados			
	Precisão	<i>Recall</i>	<i>F-score</i>	Precisão	<i>Recall</i>	<i>F-score</i>
0	0.53	0.87	0.66	0.70	0.79	0.74
1	0.46	0.13	0.20	0.72	0.62	0.67
Media/Total	0.50	0.52	0.44	0.71	0.71	0.71

Um outro comportamento observado é que o algoritmo utilizado possui um melhor desempenho para analisar os casos que não apresentam desvio. Ou seja, esse método é mais preciso quando o objetivo for monitorar os *tickets* que são finalizados dentro da SLA. Como no contexto em que esta pesquisa está inserida, o objetivo era monitorar e tentar prever os casos que ultrapassarão o prazo inicial (para auxílio a tomada de decisão), esse resultado indica que esse pode não ser o melhor algoritmo, mesmo que a diferença entre os casos não seja tão grande.

A linha de base para esse algoritmo apresenta um resultado muito diferente do que foi obtido com o primeiro. Observa-se uma grande melhora em relação a previsão dos casos reais positivos, *Recall* de 0,87 (comparado aos 0,67 do primeiro algoritmo), mas que ao mesmo tempo apresenta um piora acentuada em relação aos casos falso positivos com *Recall* bem abaixo do primeiro caso, 0,13. Para essa configuração, a regressão logística tende a classificar os casos como não desviantes, o que, na média entre os valores para casos positivos e negativos, faz com que seu resultado acerte em menos do que 50% dos casos (*F-score* = 0,44), sendo pior do que um palpite aleatório.

O resultado da combinação dos atributos também apresenta progresso. Como a linha de base desse algoritmo teve desempenho muito baixo, a melhora é ainda mais significativa.

Assim como o primeiro algoritmo, a regressão logística consegue prever com mais precisão os casos não desviantes. Porém, com uma diferença menor em relação a esse para os *tickets* cuja resolução que fogem da SLA pré-estabelecida. Isso mostra que, apesar de ainda não ser ideal, funciona melhor para previsão em um *log* com as características do contexto desse projeto. Isso pode ser explicado, uma vez que a regressão logística funciona melhor onde os dados são esparsos, que é o caso do modelo *bag-of-words*.

4.4 Discussão dos resultados

Como mencionado anteriormente, algumas características influenciam diretamente no tempo de resolução dos chamados. Um exemplo disso é a quantidade de atividades necessárias até que se chegue à resolução do problema. Além dessa característica presente no processo, as informações atreladas a essas atividades também podem exercer grande influência, como tipo do *ticket* ou a fila de atendimento em que ele está associado, visto que pessoas diferentes operam de modo diferente.

Neste artigo, o objetivo foi verificar como as mensagens trocadas durante a execução de cada *ticket* influenciam no tempo de resolução e entender como alguns dos algoritmos de aprendizado de máquina se comportam para um *log* com características semelhantes. No contexto descrito, a base de dados apresenta uma quantidade balanceada de valores positivos e negativos para o que se deseja prever.

Outro ponto é que diferentemente do apresentado por Teineema *et al.* (2016), o corpo dos *e-mails* associados às atividades é escrito pelos próprios usuários e não seguem nenhuma estrutura padrão. Assim, foi possível estudar as hipóteses levantadas durante o artigo citado para o comportamento em textos mais heterogêneos.

Uma das questões foi que o seu artigo informa que os resultados obtidos com floresta randômica foram melhores quando comparados aos da regressão logística, o que não foi observado em nossa pesquisa. Por mais que a linha de base tenha apresentado um resultado ruim para algoritmo, sua previsão apresenta resultado tão bom quanto o de floresta randômica, ao combiná-los com as informações extraídas dos textos.

Com base nas análises feitas durante a apresentação das matrizes de confusão e relatórios de classificação, observa-se que mesmo com comportamento intermediário diferente para cada um dos casos, previsão dos falsos positivos e negativos, por exemplo, o resultado reforça que a utilização de dados extraídos de texto ajudam na eficiência para previsão de um determinado valor para uma classe, ao final do processo em andamento, já que em todos os casos o valor para *F-score* sofreu grande acréscimo.

5. Conclusões

O principal objetivo da pesquisa foi o de reforçar ou refutar a hipótese de que a combinação de características extraídas de texto livre e atributos estruturados inerentes ao processo, melhora a eficiência na predição de resultado para os casos em andamento. Para alcançar esse objetivo, outros estudos foram utilizados como base da pesquisa, em especial o de Teinmaa *et al.*, (2016), que realizou testes com diversos modelos de texto para processamento de linguagem natural e algoritmos de aprendizagem de máquina. Os resultados e análise desse artigo serviram como ponto de partida para as definições do projeto. Um exemplo foi a escolha do modelo de texto a ser utilizado.

Por mais que o comportamento dos algoritmos tenha sido um pouco diferente do mencionado no artigo, os resultados obtidos reforçam a hipótese de que essa combinação melhora consideravelmente a predição em processos de negócio. Isso mostra mais uma vez o quão importante é a área de processamento de linguagem natural dentro do contexto atual, onde a quantidade e velocidade de mensagens trocadas na internet aumenta gradativamente.

Um dos principais pontos de dificuldade ao se trabalhar com texto livre é a quantidade de informação extra existente nos documentos estudados. Para lidar com essas questões, existem diversas técnicas de limpeza, com o objetivo de obter o mínimo possível do conteúdo que apresente dados que sejam relevantes para o processo.

Neste artigo, diversas dessas técnicas foram aplicadas para eliminar as palavras irrelevantes ao contexto. Porém, por ser tratar da troca de *e-mails* entre o suporte e o cliente, muitas das mensagens analisadas possuíam parte da sequência anterior, dentro do mesmo corpo. Caso fosse aplicado para todos os casos, uma solução seria a seleção somente do último *e-mail* trocado. Como isso não se aplica ao *log*, todas as mensagens foram mantidas para que não houvesse a perda de conteúdo, porém essa solução gera tópicos repetidos.

Como trabalhos futuros, serão realizados estudos com outros algoritmos de aprendizado máquina e mineração de texto e em um *log* com características semelhantes. Além disso, será considerada a análise do impacto de descrições erradas dos problemas, durante a resolução.

6. Referências Bibliográficas

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. and Kochut, K. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. In Proceedings of KDD Bigdas, Halifax, Canada, August 2017, 13 pages.

Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 0-387-31073-8.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation. The Journal of machine Learning research 3 (2003), 993–1022.

Breiman, L. (2001), Random forests. Machine learning 45(1), 5-32.

Conforti R., Fink S., Manderscheid J., Röglinger M. (2016). PRISM – A Predictive Risk Monitoring Approach for Business Processes. In: La Rosa M., Loos P., Pastor O. (eds) Business Process Management. BPM 2016. Lecture Notes in Computer Science, vol 9850. Springer, Cham.

Conforti, R., de Leoni, M., Rosa, M. L., Aalst, W. M. P. (2013). Supporting risk-informed decisions during business process execution. In: Proc. of CAiSE. pp. 116–132.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (2013). Fundamentals of Business Process Management. Springer.

Fayyad, U; Piatetsky-Shapiro, G; Smyth, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, 1996.

Francisco, R. Portela Santos, E. A. (2011). Aplicação da Mineração de Processos como uma prática para a Gestão do Conhecimento. Simpósio Brasileiro de Sistemas de Informação (SBSI). Salvador, BA. 447-484.

Maggi, F.M., Francescomarino, C. D., Dumas, M., Ghidini, C. (2013). Predictive Monitoring of Business Processes. arXiv:1312.4874v2 [cs.SE], 19 Dec 2013.

Oliveira, D., Delbem, A. (2018) Compreendendo e Prevendo o Processo Legislativo na Câmara dos Deputados do Brasil. Simpósio Brasileiro de Sistemas de Informação, SBSI 2018. Caxias do Sul.

Oliveira, M. e Bertucci, M. G. E. S. (2003). A pequena e média empresa e a gestão da informação. Informação & Sociedade: Estudos, João Pessoa, v. 13, n. 2.

Pesic, M., Aalst, W.M.P. (2006). A Declarative Approach for Flexible Business Processes Management. In: BPM Conference 2006 Workshops. pp. 169–180.

Senderovich, A., Shleyfman, A., Weidlich, M., Gal, A. and Mandelbaum, A. (2016). P3-Folder: Optimal Model Simplification for Improving Accuracy in Process Performance Prediction. BPM 2016, LNCS 9850, pp. 418-436.

Teinemaa, I.; Dumas, M.; Maggi, F. M.; Francescomarino, C. D. (2016). Predictive Business Process Monitoring with Structured and Unstructured Data. In Business Process Management, pages 401–417.

Walker, SH; Duncan, DB (1967). "Estimation of the probability of an event as a function of several independent variables". Biometrika. 54: 167–178. doi:10.2307/2333860.

Weiss, S. I., and Kulikowski, C. (1991). Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems. San Francisco, California: Morgan Kaufmann.

Yeshchenko A., Durier F., Revoredo K., Mendling J., Santoro F. (2018) Context-Aware Predictive Process Monitoring: The Impact of News Sentiment. In: Panetto H., Debruyne C., Proper H., Ardagna C., Roman D., Meersman R. (eds). OTM 2018 Conferences.

Métricas de Tamanho de Software com Métodos Ágeis no Setor Público: uma Revisão Sistemática

Alexandre F. Mendonça¹, Fernanda M. R. Alencar^{1,2}

¹Programa de Pós-Graduação Acadêmica em Engenharia de Computação
– Escola Politécnica de Pernambuco – Universidade de Pernambuco (UPE)
Rua Benfica, 455 – 50720-001 – Recife – PE – Brazil

²Universidade Federal de Pernambuco (UFPE) – Recife, PE – Brazil

{afm2, fernandaalenc}@ecomppoli.br, fernanda.alencar@ufpe.br

Abstract. *Public administration has used agile methods, adapting to the reality of the market. In addition, your deliveries must be measured using some functional size measurement. Therefore, your choice can define the level of success of public projects. In this paper, functional size measurements are identified in conjunction with agile methods that are most common in the public sector. The methodology used was a systematic literature review. The IFPUG Function Point metric and the Agile Manifesto and Scrum method were identified as the most used in the public sector. The definition of the use of functional size measurements and agile methods in the public sector makes it possible to direct decision-making towards the successful completion of projects.*

Resumo. *A administração pública tem utilizado métodos ágeis, adequando-se à realidade do mercado. Além disso, suas entregas devem ser mensuradas através de alguma métrica de tamanho de software. Portanto, a sua escolha pode definir o nível de sucesso dos projetos públicos. Neste artigo, identifica-se as métricas de tamanho de software em conjunto com métodos ágeis de maior ocorrência no setor público. A metodologia utilizada foi a revisão sistemática da literatura. A métrica Ponto de Função IFPUG e o método Manifesto Ágil e Scrum foram identificados como os mais utilizados no setor público. A definição do uso de métricas e métodos ágeis no setor público torna possível direcionar a tomada de decisões para a conclusão com sucesso dos projetos.*

1. Introdução

Mais e mais organizações estão adotando o desenvolvimento de software Ágil [Cohn 2010]. Os métodos ágeis (MA) são particularmente úteis quando os requisitos estão mudando continuamente durante a vida do projeto e/ou estão fortemente incompletos na fase de inicialização do projeto [Abrahamsson et al. 2017]. Nesse contexto, as organizações estão engajadas na busca de métodos quantitativos aceitáveis para medir a eficiência e eficácia de seus processos.

Inúmeros métodos de dimensionamento de software foram criados, mas se depararam com limitações, tais como: não serem aplicados no início do processo de desenvolvimento de software; não serem aplicados uniformemente durante a vida útil do software; e, não serem entendidos de forma significativa pelos usuários do software que

recebem estimativas baseadas nesses métodos de dimensionamento. Assim, os conceitos de métrica de tamanho de software (MTS) são projetados para superar essas limitações, deslocando o foco de medir como o software é implementado, para medir o tamanho em termos das funções requeridas pelo usuário. Neste trabalho, são consideradas as MTS mais conhecidas, como: Ponto de Função (PF) do IFPUG [Committee et al. 2010]; COSMIC [Abran et al. 2015]; e, Use Case Point [Ochodek et al. 2011].

Nas atividades de Tecnologia da Informação (TI) no setor público, deve-se estar em conformidade com as diretrizes das Agências de Supervisão Nacionais. No Brasil, essas diretrizes são regulamentadas pelo Tribunal de Contas da União (TCU) e indicam que os MA são incentivados e que podem ser escolhidas melhores MTS, para o gerenciamento de projetos de TI. Atualmente, o PF é uma das MTS mais utilizadas na administração pública brasileira [Hazan 2010]. Entretanto interessa saber se outras configurações de MTS com MA podem ser mais adequadas ao setor público. Assim, a discussão sobre a definição de um conjunto de configurações de MTS e MA no setor público é o objetivo do trabalho. Para o levantamento das principais MTS e MA aderentes ao setor público utilizou-se a Revisão Sistemática da Literatura (RSL). Ao todo foram avaliados 2.054 trabalhos, relevando duas métricas de tamanho de software (Ponto de Função do IFPUG e COSMIC) e dois métodos ágeis (SCRUM e XP) que são mais utilizados em conjunto no setor público, de acordo com a literatura, nos últimos nove anos.

Segue a estrutura do trabalho: na seção 2, apresenta-se as MTS, os MA e as regulações do TCU; na seção 3, apresenta-se todo o processo da RSL, exibindo as MTS e os MA mais aderentes ao setor público; na seção 4, apresenta-se os principais resultados da RSL e os trabalhos relacionados; na seção 5, se analisa as MTS e os MA com intuito de identificar suas principais configurações e características; por fim, na seção 6, apresenta-se a conclusão e trabalhos futuros.

2. Background

Nesta seção, apresenta-se descrição sobre as MTS e os MA selecionados nessa pesquisa, além das Regulamentações do TCU.

2.1. Métricas de Tamanho de Software

O fator de custo mais comum entre todos os métodos de estimativa é o tamanho de software [Munialo and Muketha 2016]. Portanto, esforço e custo podem ser estimados diretamente do tamanho do software usando alguma das MTS [Javdani et al. 2013]. MTS é um exemplo de quantificação de valor para o cliente [Fehlmann and Santillo 2010].

Cada MTS tem sua especificidade e pode ser melhor aplicada em contextos específicos. Algumas das MTS mais conhecidas são Ponto de Caso de Uso (PCU), COSMIC [Abran et al. 2015] e Ponto de Função do IFPUG (PF) [Fehlmann 2002].

2.2. Métodos Ágeis

Os métodos ágeis tiveram sua origem no Manifesto Ágil em 2001 [Manifesto 2001]. Ele detalha quatro valores e doze princípios que delimitam a forma de atuação do pensamento ágil em projetos de desenvolvimento de software.

Os princípios e valores descritos no Manifesto Ágil são respeitados, em menor ou maior grau, pelos métodos ágeis vigentes. Por isso, temos algumas variedades de

métodos que focam em características distintas do Manifesto Ágil. Alguns MA relevantes apresentados estão incluídos nessa pesquisa por serem os primeiros MA reconhecidos pelo mercado. Eles são Dynamic Systems Development Method (DSDM), Lean, Kanban, Scrum e eXtreme Programming (XP).

2.3. Regulamentações do TCU

De acordo com a Instrução Normativa - IN 04/2010 [Federal 2010], as métricas baseadas em horas são desencorajadas, pois exigem que o setor público se justifique previamente e que sempre vincule as entregas aos critérios de prazo e qualidade previamente definidos.

O acórdão 2314/2013 do TCU [de Contas da União 2013] relata que os princípios da administração pública relacionados à aquisição de TI estão alinhados aos valores ágeis. Posteriormente, a auditoria pertencente ao acórdão 002.116 / 2015-4 do TCU [de Contas da União 2015] recomenda que o Ponto de Função do IFPUG (PF) não deva ser obrigatório. No entanto, a administração pública deve definir métricas apropriadas associando a pagamentos relacionados com resultados.

3. Revisão Sistemática da Literatura

Uma revisão sistemática da literatura (RSL) é um meio de identificar, avaliar e interpretar toda a pesquisa disponível relevante para uma questão de pesquisa específica [Keele et al. 2007]. A principal razão para realizar uma RSL é identificar qualquer lacuna que sugira áreas para investigação adicional.

Para essa pesquisa, o trabalho de revisão teve sua primeira abordagem realizada no primeiro semestre de 2018. Essa abordagem resultou em trinta e sete artigos foram selecionados para a lista final. Esses artigos respondiam a pelo menos uma das questões de pesquisa (principal ou secundárias). O subconjunto que responde apenas à pergunta principal da pesquisa era composto por apenas três artigos. Esse baixo quantitativo de artigos que responde a pergunta de pesquisa principal desse estudo não se mostrou suficientemente representativa. Por consequência, uma nova estratégia para realizar a RSL foi proposta e executada no segundo semestre de 2018, mais especificamente de 05 a 09/out/2018. As subseções a seguir apresentam detalhes da RSL em sua segunda abordagem.

3.1. Questões de Pesquisa

O principal objetivo desta pesquisa é identificar configurações de MTS e MA melhores avaliadas para o setor público. Para tal, é necessário identificar primeiramente quais MTS e MA em conjunto são mais aderentes ao setor público. Esta pesquisa analisa a literatura para responder a seguinte questão de pesquisa principal:

(QPP) Qual é o impacto da medição de tamanho de software no emprego de métodos ágeis no setor público?

Perguntas de pesquisa secundárias são necessárias para entender todo o contexto por etapas. Essas questões secundárias escolhidas são:

(QPS1) Quais são as métricas de tamanho de software mais aderentes aos métodos ágeis?

(QPS2) Quais são os métodos ágeis mais aderentes ao setor público?

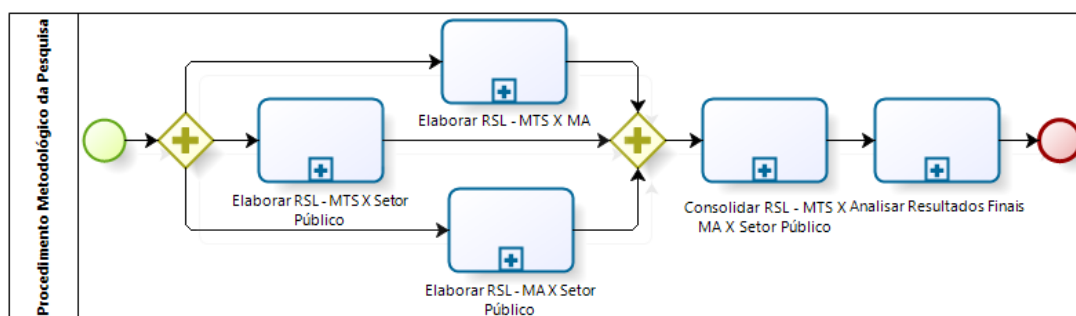
(QPS3) Quais são as métricas de tamanho de software mais aderentes ao setor público?

3.2. Estratégia de Pesquisa

Depois de entender as questões de pesquisa, algumas *keywords* foram escolhidas para criar a *string* de busca. Para aumentar as possibilidades de identificar diferentes contextos relevantes entre MTS, MA e o setor público, na mesma *string* de busca foram utilizados termos em português e inglês. Esses termos de pesquisa estão listados a seguir.

- Métricas de Tamanho de Software: ("Function Point") OR (COSMIC) OR ("Test Point") OR (UCP) OR ("Use Case Point") OR (FSM) OR ("Functional Size Measurement") OR ("Ponto de Funcao") OR ("Ponto de Teste") OR ("Ponto de Caso de Uso") OR ("Metrica de Tamanho Funcional")
- Métodos Ágeis: (KANBAN) OR (LEAN) OR (SCRUM) OR (XP) OR ("eXtreme Programming") OR (FDD) OR ("Feature Driven Development") OR (DSDM) OR ("Dynamic Systems Development Method") OR ("agile metho*") OR ("agile project") OR ("agile development") OR ("metod* agil") OR ("projeto agil") OR ("desenvolvimento agil")
- Setor Público: (government) OR ("public sector") OR ("public administration") OR ("setor publico") OR ("administracao publica")

A estratégia de pesquisa dessa RSL trabalhou com escopos intermediários delimitados pelas perguntas de pesquisa secundárias, como observado na Figura 1. Cada RSL foi conduzida de acordo com três fases principais, como sugerido por [Keele et al. 2007] e [Kitchenham and Brereton 2013]: planejamento, execução e relatório.



Powered by
bizagi
Modeler

Figura 1. Metodologia de Pesquisa (Autores, 2019).

Com os resultados de cada RSL, o foco estaria em identificar artigos que respondam a questão de pesquisa principal, consolidando a lista final de artigos dessa pesquisa.

3.3. Processo de Pesquisa

O processo adaptado para essa pesquisa contempla cinco subprocessos principais: importar artigos, aplicar critérios de inclusão/exclusão, aplicar critérios de qualidade e aplicar *snowballing* em duas rodadas consecutivas.

O objetivo da primeira etapa do processo de pesquisa é importar documentos dos engenhos de busca de acordo com a *string* de pesquisa indicada na seção 3.2. Os seguintes engenhos de busca foram selecionados como fontes de dados: ACM, Google Acadêmico, IEEE e Science Direct. Além disso, alguns trabalhos foram coletados por indicação de especialistas. Trabalhos que podem ser identificados no resultado do estudo de Vacari [Vacari and Prikladnicki 2014]. Como partimos de uma base de indicação de artigos por parte de especialistas, não recorremos a bases manuais para essa pesquisa. Após a importação de artigos, um total de 984 estudos foram disponibilizados para estudo.

A partir desse ponto, os critérios de inclusão e exclusão previamente definidos devem ser utilizados para filtrar os estudos relacionados ao objetivo da pesquisa. Essa lista de critérios deve ser usada sempre que for necessário classificar os artigos. Para esse estudo foi elaborada uma lista com um total de 14 critérios de inclusão e exclusão.

No segundo subprocesso da RSL, é realizada a leitura dos títulos, *keywords* e resumos dos artigos importados. Com esse entendimento do contexto de cada artigo, os critérios de inclusão e exclusão foram aplicados. Como resultado, 150 estudos foram selecionados para seguir para a próxima etapa do processo de pesquisa.

Na sequência, os critérios de qualidade devem ser aplicados para orientar a interpretação dos resultados determinando a força das inferências, e orientando as recomendações para novas pesquisas [Keele et al. 2007]. Uma lista de verificação de qualidade foi personalizada com base nas recomendações fornecidas em [Keele et al. 2007] e [Kitchenham and Brereton 2013]. Essa lista de verificação foi concluída com um total de 11 critérios de qualidade.

A lista de verificação de qualidade foi personalizada com base nas recomendações fornecidas em [Keele et al. 2007] e [Kitchenham and Brereton 2013]. Cada pergunta na lista de verificação de qualidade foi respondida usando uma escala de três pontos, ou seja, Sim (1 ponto), Não (0 ponto) e Parcialmente (0,5 Ponto), exceto a última questão referente ao número de citações (menos de 5 citações = 0 Ponto, entre 5 e 10 citações = 0,5 ponto e mais de 10 citações = 1 ponto). As perguntas estão listadas na Tabela 1. Cada estudo poderia obter de 0 a 11 pontos. Usamos o primeiro quartil ($11/4 = 2,75$) como ponto de corte para incluir um estudo, ou seja, se um estudo contabilizar menos de 2,75, seria removido da nossa lista final de estudos primários [Usman et al. 2014]. Estudos antes do ano de publicação de 2015, inclusive, sem citação também foram rejeitados.

Nesse terceiro subprocesso da RSL, é realizada a leitura na íntegra dos artigos resultantes do subprocesso anterior. Com o entendimento mais completo de cada artigo, os critérios de qualidade foram aplicados. Como resultado, 54 estudos foram selecionados para seguir para a próxima etapa do processo de pesquisa.

Neste ponto do processo de pesquisa, aplicamos a técnica *snowballing* [Wohlin 2014] para melhorar nossos resultados. Usando os estudos resultantes da última etapa, todas as suas citações e artigos referenciados foram coletados. Como consequência, um adicional de 704 estudos foram incorporados à pesquisa na primeira rodada e 366 na segunda rodada de *snowballing*.

Os critérios de inclusão e exclusão foram aplicados a todos os artigos coletados por *snowballing*. Como resultado temos 24 artigos adicionados à lista final de artigos na primeira rodada e 12 artigos na segunda rodada. Neste momento da RSL, temos o

Tabela 1. Lista de Verificação de Qualidade

Question	Score
Objetivos estão claramente apresentado?	Sim/Não/Parcialmente
Método de pesquisa está claramente apresentado?	Sim/Não/Parcialmente
Ambiente na qual a pesquisa foi realizada está claramente apresentado?	Sim/Não/Parcialmente
Método de análise de dados está claramente apresentado?	Sim/Não/Parcialmente
Resultados estão claramente apresentados?	Sim/Não/Parcialmente
Perguntas de pesquisa estão respondidas?	Sim/Não/Parcialmente
Artigo compara seus resultados com resultados de artigos relacionados?	Sim/Não/Parcialmente
Ameaças a validade estão claramente apresentadas?	Sim/Não/Parcialmente
Dificuldades e limitações encontrados para replicar o estudo do artigo estão claramente apresentados?	Sim/Não/Parcialmente
Artigo apresenta benefícios e/ou falhas no relacionamento entre MTS, MA e/ou setor público?	Sim/Não/Parcialmente
Número de citações	Mais de 10 citações/Menos de 5 citações/Entre 5 e 10 citações

resultado final de cada RSL individual que responde às perguntas secundárias de pesquisa. RSL direcionada a MTS com MA obteve 44 artigos em sua lista final, MTS com setor público 6 artigos e MA com setor público 40 artigos, conforme pode ser observado na Tabela 2.

Tabela 2. Resultado das Rodadas de *Snowballing*

RSL	Critérios de Qualidade	<i>Snowballing</i> (Rodada 1)	<i>Snowballing</i> (Rodada 2)
MTS x MA	28	42	44
MTS X Setor Público	5	6	6
MA X Setor Público	21	30	40

3.3.1. Consolidação dos Resultados

Com as listas finais de artigos de cada RSL, o objetivo nessa etapa é verificar artigos de cada RSL que contemplem os três elementos dessa pesquisa para formar a lista final da RSL. O processo de consolidação dos resultados foi completado através da identificação de artigos pertencentes a pelo menos duas das três RSL, que identificou quatro artigos.

Além disso, artigos pertencentes a apenas uma RSL também foram avaliados com intuito de identificar sua aderência ao resultado final dessa pesquisa, que identificou outros três artigos. Por consequência, responde à pergunta de pesquisa principal desse estudo.

Como resultado dessa consolidação foram identificados sete artigos que encerram a lista final de artigos desse estudo. Esses artigos estão detalhados na Tabela 3.

Tabela 3. Lista de Estudos Primários

ID	Título	Autor	Ano de Publicação
[P01]	A replicated study on correlating agile team velocity measured in function and story points	Hennie Huijgens and Rini van Solingen	2014
[P02]	Adoção de métodos ágeis em uma Instituição Pública de grande porte - um estudo de caso	Claudia de O. Melo and Gisele R. M. Ferreira	2010
[P03]	Contracting for agile software development in the department of defense: an introduction	Eileen Wrubel and Jon Gross	2015
[P04]	Contratação de Serviços de Desenvolvimento de Software na Administração Pública Federal por meio de uma Métrica Orientada ao Comportamento de Sistema (Regras de Negócio)	Claudson dos Santos Melo	2017
[P05]	Using Function Points in Agile Projects	Célio Santana, Fabiana Leoneo, Alexandre Vasconcelos and Cristine Gusmão	2011
[P06]	Using scrum in outsourced government projects: An action research	Thatiany L. de Sousa, Elaine Venson, Rejane M. da C. Figueiredo, Ricardo A. Kosloski and Luiz C. M. Ribeiro Júnior	2016

Continua na próxima página

Tabela 3 – *Continua da página anterior*

ID	Título	Autor	Ano de Publicação
[P07]	Using the COSMIC method to evaluate the quality of the documentation of Agile user stories	Jean-Marc Desharnais, Buğra Kocatürk and Alain Abran	2011

4. Resultados Encontrados

A partir dos resultados identificados da RSL, pode-se focar nas respostas das questões de pesquisa. A análise das RSL intermediárias estão apresentadas a seguir, seguida da análise da consolidação das RSL, que responde a questão de pesquisa principal.

A RSL MTS X MA evidenciou 44 artigos em sua lista final (Tabela 2). As MTS com maior ocorrência na ordem são COSMIC e PF, totalizando 82% das ocorrências, como pode ser observado na Tabela 4. Esse resultado mostra claramente a relevância da métrica COSMIC sendo utilizada em projetos ágeis com mais da metade de participação dos artigos selecionados por essa RSL. Vale salientar a relevância do PF com praticamente um terço de participação.

Do ponto de vista de MA, temos como maiores ocorrências, Manifesto Ágil e Scrum, totalizando 95% das ocorrências, como pode ser observado na Tabela 4. A grande maioria dos projetos que utilizam alguma MTS executam seus projetos através de conceitos do Manifesto Ágil, sem estar associado a uma MA específica. Vale salientar a relevância do Scrum com um quinto de participação. DSDM, FDD, Kanban e Lean não foram identificadas associadas a MTS independente de setor privado ou público.

Consolidando as combinações entre MTS e MA independente de setor privado ou público, as mais observadas são COSMIC e PF combinados com Manifesto Ágil e Scrum, totalizando 78% das ocorrências, como pode ser observado na Tabela 4. MTS e MA com maior ocorrência individualmente também são os que aparecem com mais frequência quando combinados.

A RSL MTS X Setor Público evidenciou apenas 6 artigos em sua lista final (Tabela 2). A MTS predominante em projetos do setor público sem presença de MA é o PF com 83% de ocorrências, como pode ser observado na Tabela 5. É importante tornar clara a ausência no setor público das MTS utilizadas nessa RSL: COSMIC e PCU.

A RSL MA X Setor Público evidenciou 40 artigos em sua lista final (Tabela 2). As MA com maior ocorrência são Manifesto Ágil e Scrum, totalizando 87% das ocorrências, como pode ser observado na Tabela 5. O interessante dessa RSL é a presença da maioria dos MA utilizados no termo de pesquisa no resultado final, com exceção de FDD e DSDM.

4.1. RSL Consolidada

A RSL consolidada resulta em sete artigos que compõem a lista de estudos primários, respondendo assim a questão de pesquisa principal desse estudo. A MTS mais rele-

Tabela 4. Resultado RSL MTS X MA

(a) Ocorrências de MTS			(b) Ocorrências de MA		
MTS	Ocorrências	(%)	MA	Ocorrências	(%)
COSMIC	26	51%	Manifesto Ágil	33	75%
PF	16	31%	Scrum	9	20%
PCU	4	8%	XP	2	5%
Test Point	2	4%			
LOC	2	4%			
NESMA	1	2%			

(c) Ocorrências das Combinações de MTS e MA			
MTS	MA	Ocorrências	(%)
COSMIC	Manifesto Ágil	21	41%
PF	Manifesto Ágil	12	24%
COSMIC	Scrum	4	8%
PF	Scrum	3	6%
PCU	Manifesto Ágil	2	4%
PCU	Scrum	2	4%
Test Point	Manifesto Ágil	2	4%
LOC	Manifesto Ágil	2	4%
COSMIC	XP	1	2%
PF	XP	1	2%
NESMA	Manifesto Ágil	1	2%

vante identificada nessa consolidação é PF com 75% das ocorrências. Já as MA com maior ocorrência são Manifesto Ágil e Scrum, totalizando 89% das ocorrências. As combinações mais observadas são exatamente as maiores ocorrências de MTS e MA combinadas, totalizando 70% das ocorrências. Todos os resultados da RSL consolidada podem ser observados na Tabela 6.

5. Análise dos Resultados

A partir dos resultados dessa RSL é possível identificar MTS e MA mais relevantes no setor público e fazer comparações para entendermos as lacunas e oportunidades de pesquisas. Os requisitos dessas MTS e MA devem ser elencados para que possam ser melhor discutidos com intuito de que projetos no setor público obtenham uma maior probabilidade de sucesso. Outros aspectos importantes também devem ser analisados como a comparação com os resultados do setor privado e a tendência baseada na evolução dos artigos publicados.

A MTS mais relevante dessa pesquisa é o PF. Projetos regidos pela contagem de PF se deparam com desafios e precisam de uma preparação adequada para que tenham condições de gerar projetos com alto potencial de sucesso [Hazan 2010]. A seguir, algumas das características do PF são apresentadas:

- A maioria dos órgãos públicos já utiliza o PF, facilitando o *benchmark* entre empresas do setor público.

Tabela 5. Resultado das RSL do Setor Público**(a) RSL MTS X Setor Público**

MTS	Ocorrências	(%)
PF	5	83%
Test Point	1	17%

(b) RSL MA X Setor Público

MA	Ocorrências	(%)
Manifesto Ágil	25	57%
Scrum	13	30%
Lean	3	7%
XP	2	5%
Kanban	1	2%

Tabela 6. Resultados da RSL Consolidada**(a) Ocorrências de MTS**

MTS	Ocorrências	(%)
PF	6	75%
COSMIC	1	12,5%
Test Point	1	12,5%

(b) Ocorrências de MA

MA	Ocorrências	(%)
Manifesto Ágil	5	56%
Scrum	3	33%
XP	1	11%

(c) Ocorrências da Combinações de MTS e MA

MTS	MA	Ocorrências	(%)
PF	Manifesto Ágil	4	40%
PF	Scrum	3	30%
COSMIC	Manifesto Ágil	1	10%
PF	XP	1	10%
Test Point	Manifesto Ágil	1	10%

- Alternativa de medição da estimativa de tamanho de funcionalidades do produto viável a partir da documentação no início do projeto.
- Medição de software independentemente da tecnologia adotada pelo projeto de desenvolvimento.
- Contagem baseada na visão do usuário.
- Apoio na avaliação de aquisição de softwares.
- Requisitos não funcionais não contabilizam pontos de função, mas impactam nas estimativas de esforço e custo do projeto.
- Mudança abrupta de tamanho para funcionalidades de complexidade muito próximas.
- Documentação de requisitos deve ser aprovada previamente pelo cliente.
- Regras para tratamento de mudanças de requisitos específica para PF já deve estar definida previamente.
- Definição de níveis mínimos de qualidade para considerar uma entrega realizada e apta para ser contada.

- Contratada fica com o risco de produtividade.
- Necessária adequação do processo de desenvolvimento.

A RSL MTS X MA, que apresenta resultados do setor privado e público, evidencia COSMIC como a MTS mais relevante. Isso induz ao entendimento que somente o setor privado aplica essa MTS em seus projetos de software.

Os conceitos ágeis encontrados no Manifesto Ágil foram os mais relevantes no resultado dessa pesquisa, e estão elencados a seguir [Nunes 2017]:

- Ambiente global sujeito a rápidas mudanças: projetos têm de responder a novas oportunidades e mercados, mudanças de condições econômicas e ao surgimento de produtos e serviços concorrentes.
- Muitos negócios estão dispostos a equilibrar a qualidade e compromisso, com requisitos do software e com a entrega rápida.
- Processos de desenvolvimento rápido são projetados para criar software útil rapidamente de forma iterativa nos quais a especificação, o projeto, o desenvolvimento e o teste são intercalados.
- O produto não é desenvolvido e disponibilizado integralmente, mas em uma série de incrementos.

Com o entendimento das principais características do Manifesto Ágil, é importante destacar também as principais desvantagens desse método. Elas também devem consideradas como critério de escolha em projetos específicos. Entre as principais desvantagens podemos considerar que o processo não é escalável e a gestão de custos do projeto dificultada.

O método ágil mais utilizado de acordo com essa pesquisa é o Scrum. Por ser um método ágil, suas características são aderentes ao Manifesto Ágil. Scrum define papéis, artefatos e cerimônias que sistematizam o desenvolvimento ágil de projetos de software [Nunes 2017].

Os papéis que o Scrum estabelece são *Product Owner*, *Scrum Master* e Equipe Scrum. O *Product Owner* é o dono do projeto. Ele é responsável por determinar os requisitos do projeto (histórias de usuários) e prioriza a execução dessas histórias dentro das iterações do projeto (*Sprint*). O *Scrum Master* é responsável por remover os impedimentos que dificultam o bom andamento do projeto. Ele também fica responsável por manter o projeto aderente ao processo Scrum. A equipe Scrum é uma equipe multidisciplinar responsável por executar as atividades do projeto. Os integrantes da equipe Scrum devem ter acesso direto ao *Product Owner*.

Os artefatos definidos no Scrum são *Product Backlog*, *Sprint Backlog* e Gráfico *Burndown*. O *Product Backlog* é uma lista de histórias de usuários estimadas pela equipe Scrum e priorizadas pelo *Product Owner*. O *Sprint Backlog* é uma lista criada através de histórias de usuário selecionadas do *Product Backlog* que serão trabalhadas e entregues no *Sprint* que está sendo iniciado. Essa lista é composta por atividades que implementam cada uma das histórias de usuário selecionadas para o *Sprint*. Já o gráfico *Burndown* é utilizado no acompanhamento do *Sprint* por parte da equipe Scrum.

As cerimônias do Scrum são tradicionalmente relacionadas com o *Sprint* do projeto. Elas são *Sprint Planning*, *Sprint Review* e *Sprint Retrospective*. O *Sprint Planning*

é uma reunião de planejamento do *Sprint* realizada com a participação de todos os papéis do método Scrum. Seu objetivo é a criação do *Sprint Backlog* no início do *Sprint*. A *Sprint Review* acontece no final do *Sprint* com intuito de apresentar todas as histórias de usuário selecionadas pela equipe Scrum implementadas para o *Product Owner*. A *Sprint Retrospective* é realizada sem a participação do *Product Owner* no final do *Sprint*. Nela são identificadas melhorias a serem executadas no próximo *Sprint* do projeto.

A RSL MTS X MA, que apresenta resultados do setor privado e público, evidencia Manifesto Ágil e Scrum como as MA mais relevantes. Resultado análogo é apresentado na RSL consolidada, o que sugere nenhuma mudança significativa das MA entre o setor público e privado quando aplicadas em projetos de software.

A evolução dos artigos ao longo dos anos releva uma tendência de pesquisas sobre temas que tenham como base os elementos desse estudo. Um levantamento de quantitativo de artigo por ano de publicação indica que praticamente metade dos artigos foram publicados nos últimos três anos. Essa tendência também é observada entre setor privado e público, como visto na Tabela 7. Com isso, se percebe que se trata de um assunto atual que merece a atenção dos pesquisadores.

Tabela 7. Cumulativo por Ano de Publicação (2016 a 2018)

RSL	Crítérios	% Cumulativo
MTS x MA	MTS	51%
MTS x MA	MA	48%
RSL Consolidada	MTS	50%
RSL Consolidada	MA	44%

Fechando essa análise, é importante ressaltar que algumas limitações foram identificadas para essa pesquisa. Apesar da utilização de diferentes bases de dados, outras fontes de publicação não utilizadas podem conter artigos aderentes a esse estudo. Além disso, a leitura dos artigos foi realizada prioritariamente por um único pesquisador.

6. Conclusões e Trabalhos Futuros

O procedimento metodológico adotado nessa pesquisa visa identificar a configurações de métricas de tamanho de software e métodos ágeis mais aderentes ao setor público. O resultado foi o evidenciamento da métrica PF e os métodos ágeis Manifesto Ágil e Scrum como as mais relevantes no contexto do setor público. Esse resultado final foi potencializado com a adoção da prática *snowballing*.

Com a definição de configurações de uso de métricas e métodos ágeis no setor público, espera-se possível direcionar a tomada de decisões para a conclusão com sucesso dos projetos, no menor tempo possível, com o menor custo aceitável e entregando valor agregado perceptível à sociedade.

Como trabalhos futuros, algumas vertentes podem ser exploradas. Um melhor entendimento da falta de utilização da métrica COSMIC no setor público vale ser melhor estudado visto que essa métrica é muito explorada nos projetos do setor privado. Os resultados dessa pesquisa podem ser apresentados para avaliação de especialistas da administração pública para ratificar o entendimento podendo identificar possíveis novas adaptações de configurações de MTS e MA para o setor público. Finalmente, uma

elaboração de um guia para escolha de MTS e MA mais aderentes aos requisitos da cultura organizacional da empresa, à *expertise* dos profissionais que executarão o projeto e às características do próprio projeto. Com isso espera-se como benefício a melhoria na tomada de decisões sobre MTS e MA a serem adotados pelo setor público antes de iniciar o projeto, idealmente no momento de definição do edital de licitações dessa administração pública.

6.0.1. Agradecimentos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.
- Abran, A., Londeix, B., O'Neill, M., Santillo, L., Vogelezang, F., Desharnais, J., Morris, P., Rollo, T., Symons, C., Lesterhuis, A., et al. (2015). The cosmic functional size measurement method. *Measurement Manual, Version*, 4(1).
- Cohn, M. (2010). *Succeeding with agile: software development using Scrum*. Pearson Education.
- Committee, I. C. P. et al. (2010). Function point counting practices manual-version 4.3. 1. *International Function Point User Group (IFPUG), Princeton Junction, NJ*.
- de Contas da União, T. (2013). Acórdão nº 2314/2013. available in <https://contas.tcu.gov.br/etcu/ObterDocumentoSisdoc?seAbrirDocNoBrowser=true&codArqCatalogado=6488150>.
- de Contas da União, T. (2015). Acórdão nº 002.116 / 2015-4. available in <http://portal.tcu.gov.br/lumis/portal/file/fileDownload.jsp?fileId=8A8182A24FFC4F0B0150010D71242BB0&inline=1>.
- Federal, P. E. (2010). Instrução normativa nº 04 de 12 de novembro de 2010. available in https://www.governodigital.gov.br/documentos-e-arquivos/legislacao/IN%20SLTI%20MP%2004%202010%20-%20Consolidada%20-%20Modificada%20%20pela%20I.pdf/at_download/file.
- Fehlmann, T. (2002). Combinatory metrics for software development. In *8th International Symposium in QFD*.
- Fehlmann, T. and Santillo, L. (2010). From story points to cosmic function points in agile software development—a six sigma perspective. In *Metrikon-Software Metrik Kongress*, page 24.
- Hazan, C. (2010). Como evitar armadilhas em contratos de fábricas de software. *Revista do TCU*, 117(1).
- Javdani, T., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M., and Parizi, R. M. (2013). On the current measurement practices in agile software development. *arXiv preprint arXiv:1301.5964*.

- Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE.
- Kitchenham, B. and Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075.
- Manifesto, A. (2001). Agile manifesto. *available in <http://www.agilemanifesto.org>*.
- Munialo, S. W. and Muketha, G. M. (2016). A review of agile software effort estimation methods.
- Nunes, R. D. (2017). A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming (xp): uma alternativa para pequenas empresas do setor de tecnologia da informação. *ForScience*, 4(2).
- Ochodek, M., Nawrocki, J., and Kwarciak, K. (2011). Simplifying effort estimation based on use case points. *Information and Software Technology*, 53(3):200–213.
- Usman, M., Mendes, E., Weidt, F., and Britto, R. (2014). Effort estimation in agile software development: a systematic literature review. In *Proceedings of the 10th international conference on predictive models in software engineering*, pages 82–91.
- Vacari, I. and Prikladnicki, R. (2014). Metodologias ágeis na administração pública: uma revisão sistemática da literatura. In *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*. In: WORKSHOP BRASILEIRO DE MÉTODOS ÁGEIS, 5., 2014, Florianópolis. Resumos
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10.

Uso da Comunicação Não Violenta na Análise de Cenários de Conflitos em Engenharia de Requisitos

Aline F. Barbosa¹, Maria Lencastre¹

¹Ecomp – Universidade de Pernambuco (UPE) – Recife, PE – Brazil

{afb, mlpm}@ecomp.poli.br

Abstract. *Requirements Engineering involves several social interactions, which register analysts at a high level of cooperation and empathy with people of different personalities, values and interests. In software development teams, regardless of the size of the project, conflicts are part of their life cycle. Therefore, it is important for the requirements analyst to continuously monitor the signs that indicate the existence of conflicts, avoiding the compromise of the project. This article focuses on identifying recurrent conflict scenarios in software development teams and uses a strategy for analyzing them, Nonviolent Communication (CNV). A qualitative study was carried out using a semi-structured interview with professionals from 9 different companies as a collection method, aiming to deepen the identified conflict situations. A set of real conflict scenarios were raised, as well as their analysis, based on the CNV; this helped the participants (interviewees) to observe the fact, regardless of judgments, as well as to identify the feelings and needs present in the situation; thus, they started to have a better understanding of the scenario where the conflict occurred.*

Resumo. *A Engenharia de Requisitos envolve inúmeras interações sociais, que requerem do analista um alto nível de cooperação e empatia com pessoas de personalidades, valores e interesses diferentes. Nas equipes de desenvolvimento de software, independentemente do tamanho que seja o projeto, os conflitos fazem parte do seu ciclo de vida. Portanto, é importante o analista de requisitos monitorar continuamente os sinais que apontam a existência de conflitos, evitando o comprometimento do projeto. Este artigo se concentra em identificar cenários de conflitos recorrentes nas equipes de desenvolvimento de software e utiliza uma estratégia para análise dos mesmos, a Comunicação Não Violenta (CNV). Foi realizado um estudo qualitativo utilizando como método de coleta uma entrevista semiestruturada com profissionais de 9 empresas distintas visando um aprofundamento nas situações de conflitos identificadas. Foram levantados um conjunto de cenários de conflito reais, assim como feita a análise dos mesmos, com base na CNV; isto ajudou os participantes (entrevistados) a observar o fato, independente de julgamentos, assim como identificar os sentimentos e necessidades presentes na situação; dessa forma, eles passaram a ter uma melhor compreensão do cenário onde ocorreu o conflito.*

1. Introdução

Nas últimas décadas, maior atenção foi dada à pesquisa sobre os fatores humanos na Engenharia de Software (ES); em particular com o movimento ágil, a prática da ES começou a se concentrar mais nas pessoas e na colaboração da equipe [Schwaber 2002]. Como consequência, profissionais e pesquisadores da ES fortaleceram o trabalho conjunto com outras áreas, como sociologia e psicologia, para assim poder entender e aprender a lidar com os comportamentos e habilidades humanas [Lenberg 2014].

Especificamente, a subárea da ES, a Engenharia de Requisitos (ER) - que envolve elicitar, analisar, estruturar e gerenciar os requisitos - requer do analista um domínio sólido da linguagem, que saiba ouvir, compreender e consolidar de forma clara as necessidades das partes interessadas [Pohl 2011]. Por se tratar de uma área que necessita constante interação humana, onde falhas de comunicação e de entendimento acontecem com frequência nas relações interpessoais (entre o analista de requisitos, partes interessadas e equipe) ocorre, conseqüentemente, a geração de conflitos [Camacho 2005].

De acordo com [Karn 2008], o conflito é um desacordo natural entre indivíduos que divergem em atitudes, crenças, valores ou necessidades. Neste sentido, compete ao analista de requisitos saber lidar com pessoas de diferentes personalidades, assim como monitorar continuamente os sinais que apontam a existência de conflitos, evitando o comprometimento do projeto [Pohl 2011]. Identificar esses sinais e o motivo por trás do conflito pode não ser uma tarefa fácil, pois ainda se tem dificuldades em ouvir o outro sempre que houver discordâncias de ponto de vista ou até mesmo expressar-se honestamente sem incluir suposições e acusações. Para [Muszkat 2003] as formas de comunicação ambíguas, referências equivocadas e afirmações acusatórias fazem da comunicação um dos vilões responsáveis pelo conflito.

Por outro lado, com objetivo de auxiliar na resolução de conflitos na sociedade de um modo geral, Marshall Rosenberg desenvolveu uma abordagem denominada Comunicação Não-Violenta (CNV) [Rosenberg 2006]. Esta abordagem propõe ajudar a reformular a maneira pela qual nos expressamos e escutamos os outros, mediante o uso de quatro (4) etapas que incluem a identificação: “do que observamos”, “do que sentimos”, “do que necessitamos” e “o que pedimos”. Esta abordagem vem sendo aplicada em diferentes contextos, tais como: educação [Baesler 2014], empresas [Garcia 2017], saúde [Nosek 2014], entre outros. Apesar da realização de diferentes buscas, os autores deste artigo não encontraram evidências de sua aplicação na ER, abrindo desta forma, uma oportunidade a ser investigada.

Este trabalho tem como objetivo investigar os conflitos na ER e seus impactos nos projetos e nas equipes, considerando cenários reais. Para isso, foram contatados e entrevistados profissionais de 9 empresas de grande porte, distintas, buscando uma identificação desses cenários e sua classificação. Posteriormente, os mesmos foram analisados no contexto da CNV, por meio de um *template* proposto em [Barbosa 2019]; o objetivo foi o aprofundamento complementar considerando os elementos essenciais abordados por [Rosenberg 2006].

Este artigo está organizado da seguinte forma: a seção 2 apresenta o *background* que apoia a compreensão dos conceitos usados neste artigo; já a seção 3 apresenta a metodologia usada para o levantamento de cenários e análise dos conflitos com a CNV.

A seção 4 faz uma análise de resultados; por fim, a seção 5 apresenta conclusões e trabalhos futuros.

2. Background

Esta seção apresenta conceitos relacionados a conflitos, sua classificação e formas de administrá-los, além de aprofundar a descrição e etapas da Comunicação Não Violenta.

2.1 Conceito de Conflito, Classificações e Administração

O conflito é compreendido como qualquer tipo de oposição ou interação antagônica originada por diversos motivos, tais como valores diferentes, recursos ou posição social e disputa de poder [Robbins 1974]. Para [Kofman 2004], o “ser humano implica em ter conflitos, o que faz parte da natureza humana, sendo inevitável escapar de nossas necessidades, medos, egoísmos e aborrecimentos”.

O conflito geralmente é encarado como algo negativo que deve ser eliminado; mas, compreender a sua existência pode evitar problemas destrutivos, oferecendo muitas ideias a partir do dinamismo gerado em busca de uma solução. Neste sentido, [Jehn 2003] aponta que existem três tipos de conflitos: relacionamento, tarefa e processo. Cada um deles é detalhado a seguir.

- Conflito de relacionamento: existe quando há incompatibilidades interpessoais entre membros do grupo; geralmente inclui também diferenças de personalidade como diferenças de opinião e preferências.
- Conflito de tarefa: definido como discordâncias entre os membros do grupo sobre as tarefas sendo executadas, isto é, diferenças em ponto de vista, ideias, opiniões e desacordo no grupo sobre o conteúdo e questões da tarefa. Assim, esses tipos de conflito são focados no trabalho ou na tarefa, em contraste a conflitos de relacionamento que se concentram em problemas que não são de tarefa. É identificado também como “conflito de trabalho”, “discordâncias de trabalho” e “problemas de tarefa”.
- Conflito de processo: são sobre os meios para realizar as tarefas, não sobre o conteúdo ou a substância da tarefa em si, mas sobre estratégias para executá-las. Um exemplo desse tipo de conflito pode ser observado quando diferentes pesquisadores discordam sobre a interpretação dos dados e o significado dos resultados; nesse caso eles estão enfrentando conflito de tarefas; se eles discutem sobre quem é responsável por escrever o relatório final e quem fará a apresentação, eles estão tendo um conflito de processo.

Em relação às formas de administração de conflitos, [Robbins 2010] sugere cinco formas de administrar conflitos:

- Competir: quando uma pessoa busca a satisfação de seus próprios interesses, independentemente do impacto que isso terá sobre as outras partes do conflito.
- Colaborar: quando as partes conflitantes desejam satisfazer os interesses de ambas, temos uma situação de cooperação e busca de resultados mutuamente benéficos.

- Evitar: a pessoa reconhece que o conflito existe e tenta suprimi-lo ou livra-se dele.
- Acomodar-se: quando uma das partes procura apaziguar a outra, pode se dispor a colocar os interesses dela antes dos seus, isto é, uma das partes pode se sacrificar para que o relacionamento se mantenha.
- Conceder: quando cada parte em conflito abre mão de algo, acontece um compartilhamento, que desemboca em resultado de compromisso.

Para [Robbins 2010] não existe estilo certo ou errado para gerir conflitos; cada um pode ser apropriado e efetivo, dependendo da situação, do assunto a ser resolvido, e das pessoas envolvidas.

2.2 Comunicação Não-Violenta

A Comunicação Não-Violenta, também conhecida como comunicação compassiva, foi criada pelo psicólogo norte americano Marshall Rosenberg [Rosenberg 2006] a partir das suas experiências pessoais vividas em um bairro violento de Detroit. Seu interesse era compreender alternativas pacíficas de diálogo que amenizassem os conflitos com o bom uso das palavras, mantendo-se o indivíduo no seu estado natural compassivo. Ele passou a aplicar extensivamente a sua abordagem, tendo seu sucesso espalhado por diversos países.

Conforme [Rosenberg 2006], “Embora possamos não considerar ‘violenta’ a maneira de falarmos, nossas palavras geralmente causam dor aos outros e a nós mesmos”. O discurso julgador e moralizador tenta enquadrar o indivíduo dentro dos padrões comportamentais sociais ditados pela sociedade [Barros 2015], prendendo as ideias sobre o certo e errado, o bom e o mal, normal e anormal [Rosenberg 2006]; possibilita, assim, o bloqueio da compaixão tanto pelos outros quanto por nós mesmos.

Por outro lado, o caminho para prática da CNV é resgatar o estado mais natural e humanizado das pessoas para serem empáticas e compassivas na forma de se comunicar nas suas relações sociais. O intuito é reformular a maneira pela qual expressamos e ouvimos o outro, sem pré-julgamentos e comparações. Desse modo, a compaixão inevitavelmente floresce se as pessoas se mantiverem fiéis aos princípios e ao processo da CNV. Os quatro componentes principais do processo da CNV são [Rosenberg 2006]: Observação, Sentimento, Necessidades e Pedidos. Cada um deles é detalhado a seguir:

- 1) **Observação:** no relato observado não deve existir julgamentos e, também, não deve haver emissão de pontos de vista. A expressão deve ser de forma honesta, especificada de acordo com o momento vivencial e contextual. O fato deve ser relatado sem exagerar ou generalizações, ou seja, evitar uso de palavras como sempre, nunca, jamais. A ausência da avaliação ao que foi observado possibilita enxergar com clareza o fato ocorrido, como ilustra o exemplo da Tabela 1.

Table 1. Exemplos de Avaliações x Observações

Avaliações	Observações
Você <i>raramente</i> faz o que eu quero	Nas últimas 3 vezes em que comecei alguma atividade, você disse que não queria fazê-la.

Ele apareceu aqui com <i>frequência</i> .	Ele aparece aqui pelo menos três vezes por semana.
---	--

- 2) **Sentimento:** identificar e expor o que sente em relação ao que foi observado. Assim feito isso, as emoções e a conectividade com os outros se tornam mais acessíveis, contribuindo na resolução do conflito. Os sentimentos informam, já os pensamentos interpretam, como apresenta na Tabela 2.

Table 2. Exemplos de Pensamentos x Sentimentos

Pensamento	Sentimento
Quando usamos a voz passiva. Ex: Estou me sentindo <i>ignorado</i>	Me sinto <i>irritado</i>
Quando usamos “que” Ex: Sinto <i>que</i> isso não faz sentido	Fico com <i>medo</i> quando você diz isso.

- 3) **Necessidades:** reconhecer, honestamente, qual necessidade está ligada ao sentimento observado e que não está sendo atendida; as necessidades possuem um significado especial na CNV: elas são comuns a todas as pessoas e não estão ligadas a nenhuma circunstância ou estratégia específica, no que tange à realização delas. Portanto, “querer ir ao cinema com alguém específico” não é uma necessidade, ao mesmo tempo que um desejo de “passar tempo com uma pessoa em particular” também não é. Provavelmente, a necessidade nesses casos seria a busca pela companhia. Pode-se resolver a necessidade por uma companhia de muitas maneiras, e não apenas com uma pessoa específica, e muito menos indo ao cinema.
- 4) **Pedido:** deve ser especificado com clareza, através de ações concretas (que revelem o que realmente queremos). Porém, deve ser verbalizado de forma sutil, que descarte a ideia de imposição ou ordem. Deve-se expressar aquilo que estamos pedindo e não o que não estamos pedindo. Uma linguagem vaga favorece a confusão. Pedidos devem ser específicos, mensuráveis, atingíveis, relevantes e limitados a um certo tempo. A diferença entre um pedido e uma exigência fica clara quando se vê a reação da pessoa que solicitou o pedido. Se ela não aceitar reagindo de forma agressiva ou com chantagem emocional, ela não está pedindo e sim exigindo.

3. Levantamento de Cenários e Análise dos Conflitos com CNV

Esta seção apresenta como foram realizadas as atividades de Levantamento de Cenários e de Análise de Conflitos com a CNV. O Levantamento consistiu da realização de entrevistas semiestruturadas (Seção 3.1), que foram gravadas e posteriormente transcritas; já a Análise de Conflitos, foi feita pessoalmente com cada entrevistado, com o uso de artefatos de apoio à análise segundo a CNV (Seção 3.2).

A Figura 1 ilustra todo o processo utilizado, que consta das atividades de: Planejamento, Contato, Execução de Entrevistas, Análise de Resultados, Melhoria dos Cenários de Conflitos, e a Análise (Análise dos Conflitos com a CNV). Já na Tabela 3 é possível identificar os eventos de cada atividade realizada, com suas respectivas informações de execução (período, artefatos usados, quantidade de profissionais entrevistados e tempo gasto na execução).

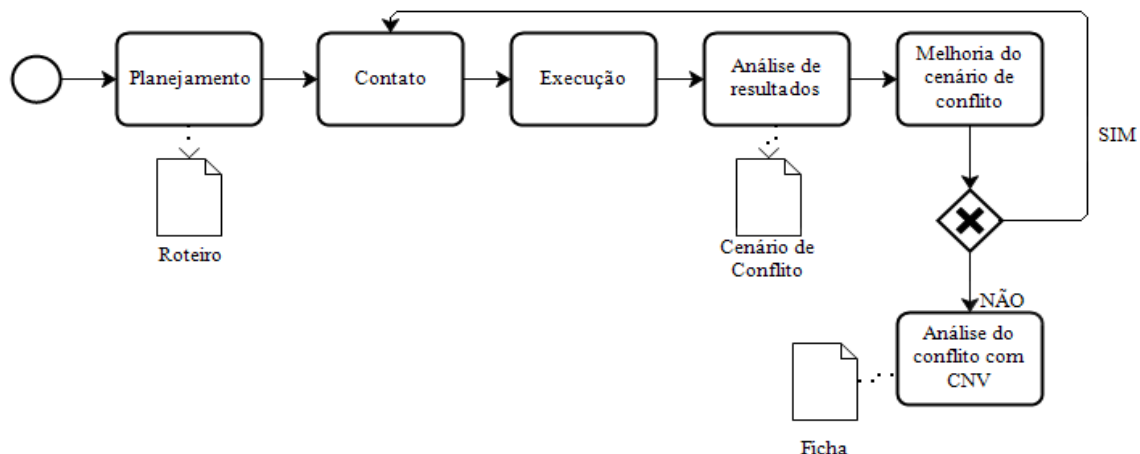


Figure 1. Processo realizado para levantar conflito

Na etapa de Planejamento foi feita uma pesquisa de trabalhos relevantes sobre conflitos, o que contribuiu para a elaboração do roteiro (questões apresentadas na Tabela 10); ainda nessa etapa, foi definido o perfil das pessoas a serem entrevistadas, que incluiu profissionais de empresas diferentes e com pelo menos 10 anos de experiência; esta escolha visou garantir uma maturidade em termos profissionais e conflitos potencialmente interessantes. Em seguida, realizou-se a etapa de Contato, isto é, o agendamento com cada profissional. Posteriormente, na Execução, foram feitas as entrevistas para aquisição dos cenários, individualmente com cada participante, seguindo o roteiro elaborado. Por fim, foi feito o processo de Análise de resultados das entrevistas e extraído os Cenários de conflitos. Em alguns casos, o cenário de conflito foi melhorado através de diferentes interações com o entrevistado, a partir das quais foi amadurecido e melhor detalhado. Ao final, o cenário de conflito foi analisado em termos de conceitos da CNV em conjunto com o respectivo profissional (Atividade Análise de Conflito com a CNV).

Table 3. Dados das atividades realizadas

Atividades		Período de Realização	Artefatos usados	Qtd Profission	Tempo de Execução
Entrevista	Preparação	Nov18	Pesquisa bibliográfica	-	12hrs
	Execução	Dez18/Jan19	Roteiro, Gravador de áudio	9	5hrs 43min
	Follow-up	Jan19	Transcrições; Análise de resultados.	-	16hrs
Análise do Conflito com a CNV	Preparação	Jan19	Cursos, Vídeos de Marshall.	-	15hrs
	Execução	Fev19	Gravador de áudio, Ficha de conflito, Lista de Sentimentos e Necessidades	9	4hrs 30min
	Follow-up	Fev19	Transcrições; Análise do conflito.	-	14hrs

Fonte: Produzido pelo autor.

Ambas as atividades, Entrevistas para levantamento e Análise de Conflitos, foram realizadas individualmente com cada participante. Os locais de cada encontro mudaram de acordo com a disponibilidade de cada entrevistado (uns sendo na universidade, outros nas empresas, por vídeo ou em outros locais).

3.1 Entrevista

Nesta seção são detalhados o Método de Pesquisa, Participantes, e a forma de Coleta de Dados.

[Método de Pesquisa] para cumprir com o objetivo deste trabalho, foi utilizado como método de coleta, uma entrevista semiestruturada. Para [Seaman 1999], esse tipo de entrevista permite maior flexibilidade ao coletar informações imprevistas e são menos dispendiosas; podem ser usadas extensivamente se comparadas com as não estruturadas. As 6 perguntas contidas no questionário são apresentadas na Tabela 4.

Table 4. Roteiro da entrevista

Grupo	ID	Questão
Contextualização	Q.1	Considerando suas experiências profissionais, quais são os problemas comuns enfrentados na fase da ER?
	Q.2	Algum desses problemas gerou situação de conflito? Em que situações ela ocorreu?
Cenário de Conflito	Q.3	O que você observou de efeitos pelo conflito citado?
	Q.4	Quais reflexos gerados para o projeto?
	Q.5	Qual abordagem/estratégia foi utilizada para resolver o conflito?
Resolução de Conflitos	Q.6	O que você considera importante ao se abordar uma situação de conflito?
	Q.7	Quais estratégias obtiveram melhor resultado para resolver conflitos?
	Q.8	Algum conflito não foi bem-sucedido quando tentou resolvê-lo?
Considerações Finais	Q.9	Tem algum outro fato que queira comentar em relação ao assunto?

Fonte: Produzido pelo autor.

[Participantes] para a definição da amostra, foram utilizadas as técnicas de não-probabilidade e amostragem de conveniência [Wohlin 2012]. Portanto, foram selecionados 9 participantes na área de computação que apresentassem experiência na Engenharia de Requisitos e que demonstraram disponibilidade para participação na pesquisa.

[Coleta da Coleta] todas as entrevistas foram realizadas utilizando o roteiro apresentado na Tabela 1, esse roteiro foi validado em uma entrevista piloto, a qual foi considerado nesta amostra, uma vez que não houve alterações bruscas nas perguntas para versão final. Das 9 (nove) entrevistas, 8 (oito) foram realizadas presencialmente,

contemplando a gravação do áudio com a permissão dos entrevistados. A transcrição foi realizada posteriormente para ajudar na análise.

3.2 Análise dos Cenários no contexto da Comunicação Não Violenta

A segunda etapa consistiu na análise dos conflitos relatados utilizando a Comunicação Não Violenta. Isto é, foi realizado um novo encontro com o participante, onde inicialmente a pessoa que entrevistou fez uma explanação sobre o que era CNV e seus principais conceitos; em seguida, o cenário de conflito relatado na entrevista foi lembrado e iniciou-se o processo de análise com CNV. Como apoio a esta atividade, um *template* foi utilizada para ser preenchida com os 4 componentes da CNV, conforme ilustra a Figura 2.

Figure 2. Template preparado e usada para análise do conflito com a CNV

O campo Cenário do Conflito requer o preenchimento da descrição sucinta do conflito a ser analisado. Já no campo Observação a descrição da situação deve ser explorada em busca de um evento específico que aconteceu relacionado ao conflito, além da identificação dos fatos. Os campos Sentimento e Necessidade devem ser preenchidos a partir das listas propostas por Marshall [15]; estas ajudam a aumentar o vocabulário. Em relação ao campo Pedidos, a parte conflitante deve observar as necessidades relatadas e procurar descrever, de forma concreta e clara, um pedido para a outra parte envolvida no conflito ou para si mesmo, buscando atender a essas necessidades.

Na Tabela 5, um conjunto de perguntas chaves são indicadas para orientar a atividade de análise ao conflito com o *template*.

Table 5. Perguntas-guias para template CNV

Momento	Pergunta
Situação	- Pense em uma situação que você queira explorar ou transformar. - Um desafio relacionado a você e outra pessoa em sua vida.
Observação	- Qual foi o evento específico que aconteceu relacionado a esta situação? - Descreva de forma factual algo que alguém (ou você mesmo) fez
Sentimentos	- Quando você pensa nessa situação, qual(is) sentimentos você sente?
Necessidades	- Quando você se conecta com esses sentimentos, quais necessidades percebe que está buscando atender?

Pedidos	- A partir dessa conexão com as suas necessidades, o que você gostaria de pedir a si mesmo? E ao outro? E a outros?
---------	---

Fonte: Produzido pelo autor.

Por fim, utilizou-se um conjunto de figuras com caricaturas de pessoas difíceis de lidar (Sabe-tudo, Chorão, Calado, Talvez e Granada) [Von 2013] para o participante escolher a que tivesse características semelhantes ao perfil da pessoa envolvida no conflito. O *template* foi preenchido na visão do participante da entrevista de acordo com a quantidade de envolvidos no conflito.

4 Resultados

Esta seção reúne os resultados e discussões do estudo realizado. Inicialmente são apresentadas as respostas da entrevista e, em seguida, a análise dos conflitos com a CNV. Para as partes das transcrições dos entrevistados o conteúdo está em *itálico*, entre aspas duplas e com recuo na tabulação.

4.1 Análise das Respostas das Entrevistas

Inicialmente, o questionário focou em obter informações gerais sobre o participante, tais como sua ocupação atual e experiência na área de ER. Os nomes dos entrevistados e suas empresas ficaram no anonimato para impedir a identificação desses indivíduos, como são descritos na Tabela 6.

Na Tabela 6 é possível observar diferentes perfis, com cargos distintos. Esse tipo de amostragem é importante, pois permite possível fazer uma análise sobre diferentes perspectivas dos conflitos que acontecem nas equipes. O tempo de experiência também foi bastante significativo, já que os participantes demonstraram ter uma acumulação de relatos relevantes para contribuir com a pesquisa.

Table 6. Perfil dos entrevistados

Perfil dos entrevistados								
Identificação do Entrevistado								
E1	E2	E3	E4	E5	E6	E7	E8	E9
Atuação profissional								
ER	ER	GP	D	ER	ER	ER	ER	ER
Tempo de experiência								
20	30	30	10	33	24	15	18	33
Tipo de organização								
Pública	Privada	Privada	Privada	Pública	Privada	Privada	Privada	Privada

Legenda: D = Desenvolvedor, GP = Gerente de Projeto, ER = Engenheiro de Requisitos

Fonte: Produzido pelo autor.

Posteriormente, na questão Q1, os participantes foram convidados a responder sobre os problemas comuns enfrentados na fase da ER baseado em suas experiências

profissionais. A Tabela 7 contém uma relação dos principais problemas citados pelos entrevistados.

Table 7. Problemas na Engenharia de Requisitos

Problema	Descrição	Entrevistado
P1: Partes interessadas com dificuldade de definir escopo	Dificuldades em definir um escopo devido a quantidade de partes interessadas envolvidas no projeto.	E1, E2, E3, E4, E5, E6, E9
P2: Comunicação interna com a equipe	Problemas em estabelecer comunicação eficaz entre os membros da equipe.	E1, E3, E9
P3: Documentação incompleta	Problemas com requisitos ambíguos e incompletos.	E1, E4
P4: Suporte insuficiente pelo cliente	Dificuldades do cliente no entendimento do “todo” para o sistema a ser desenvolvido.	E4, E5, E6, E7
P5: Coesão da equipe	Dificuldades em construir o espírito de equipe.	E3, E7
P6: Diferenças hierárquicas	Dificuldades nas relações interpessoais por causa de hierarquia.	E1, E3
P7: Mudanças de requisitos	Dificuldades em manter o combinado para as entregas estabelecidas, falta de estabilidade dos requisitos, causando inclusive estresse para o desenvolvedor.	E4, E6, E8
P8: Prazo de entrega	O prazo de entrega estabelecidos pelo o cliente desafiador para a equipe.	E6
P9: Qualificação em ER;	O cliente compreender o papel do analista no projeto; falta de treinamento sobre o processo que empresa utiliza referente aos requisitos e quais responsabilidades do analista.	E6, E8
P10: Análise de riscos do Projeto	Riscos não identificados prejudicam no andamento do projeto, exemplo: pouca participação do PO ou Stakeholders	E6, E7
P11: Priorização de Requisitos	Dificuldade em definir prioridades dos requisitos	E2, E4, E5, E6, E7
P12: Unificação de ER com outras responsabilidades	Analista com responsabilidade de testes	E8
P13: Equipe distribuída	Dificuldades em manter contato com clientes/PO/outros integrantes distantes para resolver questões referente aos requisitos	E4, E5, E8

Fonte: Produzido pelo autor.

Observa-se a relevância do contato entre analista de requisitos e as partes interessadas, exigindo competências comunicacionais para lidar com diferentes problemas que acontecem, principalmente, na elicitação e priorização de requisitos.

A seguir, mostramos alguns trechos da transcrição dos entrevistados para fins de ilustração:

E1: *"Muitos conflitos às vezes começam pela falta de comunicação, porque você ouve, documenta, o cliente valida de uma forma muito superficial e passa para equipe técnica. A equipe técnica compreende naquele nível de documentação"*

E3: *"Existe um problema na ER que, no final de contas, está ligado a comunicação. Da área de TI com a área de negócio, mas acontece mais na área de negócio. Pois, por ela conhecer muito o negócio dela, acaba não passando algumas informações pra TI, por achar que é óbvia demais."*

E5: *"Hoje o maior problema que temos é com o cliente, ele considera tudo importante e se tudo é importante o prazo fica muito longo"*.

Na questão Q2 solicitamos aos entrevistados que relatassem uma situação de conflito marcante, gerada pelo problema exposto na questão anterior. Essas situações foram detalhadas nas questões Q2-Q5. O intuito foi compreender e analisar como ocorreu o conflito, quais partes estavam envolvidas (relatado pelo entrevistado). Na Tabela 8 são descritos detalhadamente os cenários identificados como sendo os principais e mais completos relatados.

Table 8. Cenários de conflitos

Conflito/ Entrevistado	Descrição do Cenário
C1/E1	Conflito na relação entre líder da equipe e analista de grau sênior. Em um projeto com dificuldades e desgaste entre analista e cliente, foi introduzido um novo líder para facilitar e viabilizar o projeto. Porém, estabeleceu-se novo conflito entre o novo líder e o analista, que tinha um maior conhecimento técnico e não admitia ser liderado por uma pessoa recente ao projeto.
C2/E2	Conflito entre dois setores, onde o setor de contabilidade queria colocar um determinado requisito no sistema e o setor de RH não queria, pois achava que era uma invasão de gerência. O projeto consistia em um sistema de custos, onde o setor de contabilidade precisava de dados sigilosos do outro setor, que não queria disponibilizá-los.
C3/E3	Conflito entre analista e desenvolvedores. Em um projeto foi alocado um analista de requisitos que trabalhava há muito tempo com o pessoal de negócio. Porém ele não conseguia escrever os requisitos em nível que os desenvolvedores conseguissem captá-los; isto gerou muito conflito.
C4/E4	Conflito entre parte interessada e equipe, onde a parte interessada aprovava o que achava interessante sem levar em consideração o usuário final. Além disso, o mesmo ficava vendendo o sistema internamente a um gerente acima dele.
C5/E5	Conflito entre três áreas distintas, onde havia dificuldade de consenso porque era nítido que havia uma disputa de poder, onde uma área queria se sobressair em relação à outra, querendo que os seus requisitos tivessem maior prioridade.
C6/E6	Conflito entre cliente e analista de requisitos (também empresário). A empresa prestadora de serviços aceitou um projeto, principalmente devido à visibilidade do cliente na região, como o interesse para atrair novos clientes. Mas, acabou enfrentando sérios problemas devido às

	frequentes alterações nos requisitos. A principal queixa do empresário e analista de requisitos é que “o cliente quer sempre mais, porque ele acha que aquele valor irrisório que a gente cobra pequenininho e dá direito a isso”. Contudo, chegou ao ponto da demanda de retrabalho, de novas funcionalidades não cobrir as despesas do projeto, manter um funcionário qualificado para o tipo de serviço, custava mais do que o cliente pagava.
C7/E7	Conflito entre desenvolvedor e analista de requisitos. Em um projeto uma entrega foi dada como concluída; uma falha de comunicação! O cliente recebeu, pagou, mas nem tudo estava feito. Neste projeto, o desenvolvedor não tinha questionado o que não entendia dos requisitos, então a implementação era feita como ele achava que era. Devido a esse mau entendimento, foi necessário reestruturar as atividades da equipe, gerando sobrecarga de um outro desenvolvedor, que passou a integrar a equipe. Esse conflito resultou em um grande desgaste da imagem do desenvolvedor na empresa, grande desgaste da equipe perante o cliente, sobrecarga de outro desenvolvedor e atraso no prazo final do projeto.
C8/E8	Conflito na execução das atividades pelo analista de requisitos, também na função de <i>tester</i> . A principal causa foi pela falta de treinamento do novo funcionário sobre o processo a ser utilizado na elicitação de requisitos e caso de testes. A aprendizagem aconteceu pela prática e o irrisório compartilhamento de informação por um outro funcionário que exercia esse papel antes da sua chegada a equipe, o mesmo se manteve na equipe, porém como desenvolvedor. Além dessa demanda de aprendizagem pelos padrões exigidos, o projeto de sistema bancário, tinha o PO em outro estado (apontado como uma dificuldade), por outro lado, cliente estava próximo da equipe, já que o trabalho era <i>in loco</i> com o cliente.
C9/E9	Conflito entre a parte a equipe de desenvolvimento e o departamento de marketing. A empresa operadora telefônica passou por uma grave falha de comunicação interna. Tudo começou quando o analista de requisitos foi pego de surpresa ao assistir uma propaganda na TV da operadora sobre uma nova oferta que iria vir estar disponível na segunda-feira seguinte, esse anúncio foi visto em um sábado à noite. Imediatamente, o analista entrou em contato com a equipe de <i>marketing</i> da empresa para maiores informações e foi confirmado a divulgação. O problema era justamente porque não houve alinhamento entre os setores, tratava-se de uma funcionalidade nova.

Fonte: Produzido pelo autor.

Ainda referente aos conflitos identificados nas entrevistas, as Q3-Q4, buscou-se compreender os efeitos para equipe e as consequências para o projeto. A Tabela 9 apresenta os resultados expostos pelos entrevistados.

De acordo com [Robbins 2010], as consequências do conflito geradas pelo descontentamento podem reduzir a eficácia e a coesão nos grupos. Nos cenários de conflitos apresentados é possível observar que insegurança, desgaste e estresse da equipe foram efeitos frequentes, gerados pelo conflito.

Table 9. Efeitos do conflito e consequências para o projeto

Conflito	Efeitos do conflito	Consequências para o projeto
C1	Equipe desestabilizada; Cliente inseguro.	Atraso para entrega
C2	Alguns membros ficaram intrigados; Interferências do lado pessoal com o profissional.	Atraso para entrega
C3	Problemas de não aceitação do cliente; Expectativas não atingidas pela equipe (esperavam que o projeto iria fluir facilmente pelo fato de ter na equipe um analista da área de negócios); (o analista de requisitos saiu do projeto).	Mudança de escopo Custo; Atraso para entrega
C4	Expectativas não atingidas pela equipe; Expectativas não atendidas pelos usuários; Contratação de novos desenvolvedores.	Mudança de escopo constante; Atraso da entrega;
C5	Desgaste na relação interpessoal dos representantes dos setores que estavam em conflitos	Atraso da entrega; Qualidade do produto
C6	Estresse na equipe de desenvolvimento; Sobrecarga da equipe devido as alterações.	Mudança de escopo
C7	Desgaste com a equipe; desgaste com o cliente; Perda de foco; Imagem na equipe perante o cliente; Sobrecarga de trabalho para um desenvolvedor; Imagem do desenvolvedor perante a equipe;	Impacto na entrega, Custo; Reestruturação das atividades;
C8	Insegurança do cliente; Insegurança da equipe.	Mudança de escopo
C9	Desgaste na equipe; estresse da equipe.	Custo (hora extra para funcionários)

Fonte: Produzido pelo autor.

Para analisar as colocações dos participantes na questão Q5, sobre a resolução do conflito, realizamos a classificação com base em [Robbins 2010], uma vez que os entrevistados expuseram as suas soluções de uma forma informal. Além disso o tipo de conflito foi analisado de acordo com a classificação de [Jehn 2003]. A Tabela 10 apresenta os resultados dessa classificação.

Table 10. Cenários de conflitos e sua classificação

Conflito	Tipo do Conflito	Resolução
C1	Relacionamento	Acomodação
C2	Processo	Competição
C3	Tarefas e Relacionamento	Colaboração
C4	Tarefas e Processo	Colaboração
C5	Relacionamento	Competição
C6	Tarefa	Conceber
C7	Relacionamento e Tarefa	Colaboração
C8	Processo	Acomodação
C9	Tarefa	Competição

Fonte: Produzido pelo autor.

4.2 Análise dos cenários de conflito através da CNV

Através da ficha baseada na CNV, criada para análise (ver Figura. 1), foi possível revisitar junto com o entrevistado o cenário do conflito vivenciado, contemplando as diferentes perspectivas envolvidas no conflito. Para facilitar a compreensão, segue um maior detalhamento do Cenário E3:

Em um projeto foi alocado um analista de requisitos que trabalhava há muito tempo na área de negócio. Isto criou a expectativa que o processo fosse fluir bem. Porém, o novo analista definiu um escopo muito grande para a primeira release (dizia: “quanto mais melhor”) e fez uma especificação muito superficial/genérica dos requisitos. Isto criou dificuldades para a equipe de desenvolvimento, que tentou conversar com ele; este, porém, mostrou-se pouco disponível para fazer o detalhamento em um nível que os desenvolvedores conseguissem captá-los melhor; isto gerou muito conflito. O analista afirmava: “Está claro, está tudo aí escrito”. Um dos desenvolvedores era mais explosivo, o que aumentou a sensação de resistência. A primeira entrega teve problemas; por falta de informação, foi necessário refazê-la. Para o cliente o escopo era o mesmo, ele não considerou que eram necessárias mudanças e sim que tinham sido identificados defeitos. Todo mundo ficou frustrado. A equipe de desenvolvimento ficou desmotivada. Houve mudança no processo de funcionamento entre o analista e os desenvolvedores: passou a ser exigido um retorno dos desenvolvedores ao analista (no prazo de uma semana) apontando os requisitos que não estava claro, o que não era entendido, o que estava em conflito. Se o analista não concordasse, resolvia questões com o gerente; se ainda existissem réplicas faziam uma reunião conjunta. Mas o estrago já estava feito: o cliente não tinha mais confiança na equipe devido aos problemas envolvidos; o processo de solicitação de mudanças dos projetos tornou-se mais enrijecido após esse projeto; o projeto teve custos elevados e com atraso.

A Tabela 11 apresenta o detalhamento da análise de um dos cenários de conflitos (o E3) abordados em conjunto com o entrevistado. Por questões de espaço as outras tabelas referentes à análise dos outros conflitos não são apresentadas.

Table 11. Análise do Conflito do E3

Envolvidos:		Equipe desenvolvedora (P1)		Analista de Requisitos (P2)
Observação:	“Na primeira entrega o cliente identificou que as funcionalidades solicitadas não eram exatamente o que ele queria, isto é, não atendiam às necessidades dele. Por outro lado, a equipe relatou que a documentação estava muito superficial e faltando detalhes de padronização, por exemplo, o cliente reportou suas necessidades, mas elas estavam escritas de uma maneira muito genérica.”			

Sentimentos:	Frustração, Impotência, Surpreso.	Incomodado com a cobrança que chegou para ele, Resiliente, Indignado.
Necessidades:	Gentileza, valorização Conhecer/Ser reconhecido Conexão (no momento mais crítico) Congruência (as pessoas trabalharemos por objetivo comum)	Confiança Reconhecimento (ele vem da área militar, então ele levava a sério a questão de hierárquica, portanto, ele espera ser mais valorizado) Estabilidade (não sabia a quem recorrer) Feedback (da equipe/cliente sobre a documentação)
Pedidos	Olha, a gente tem claramente problemas na documentação que foi gerada por você, você está entendendo tudo, mas a equipe está lendo e não está entendendo. Precisamos conectar isso aí.	Eu vou resolver isso não precisa se preocupar

Fonte: Produzido pelo autor.

Após finalizar essa atividade, o entrevistado do cenário E3, Tabela 3, fez um relato em relação à prática da CNV “*Me fez refletir: Será que se eu estivesse perguntando a ele quais eram as necessidades dele, nós talvez não tivéssemos chegado ao ponto do analista sair do projeto?*”. Esta reflexão, referente à aplicação de CNV, é importante uma vez que a tomada de consciência de uma pessoa pode facilitar a conexão com consigo mesma e com os outros, em busca de relações mais significativas.

Porém, a atividade de análise conjunta com os entrevistados, demonstrou que, apesar das explicações dos conceitos da CNV e aparatos utilizados nas diferentes análises dos cenários, ainda existe dificuldade em expressar o observado livre de julgamentos ou até mesmo nomear o que se estava sentindo e necessitando para realização do pedido.

5. Conclusões

O profissional de requisitos atua como ponte entre a equipe de desenvolvimento e os *stakeholders*, o que lhe requer um conjunto de aptidões interpessoais para lidar com as possíveis situações de conflito em suas interações, tais como tomada de decisão, negociação e competências conversacionais. Apesar da exigência do mercado, essas habilidades nem sempre são desenvolvidas no processo de formação do profissional no meio acadêmico.

Diante deste contexto, o estudo realizado em [Barbosa et al. 2019] apresentou resultados preliminares do levantamento dos cenários de conflitos na ER. Já na dissertação de [Barbosa 2019] foi descrito um conjunto de artefatos instrucionais para análise de conflitos com base na CNV, dentre eles o *template* utilizado neste trabalho. De forma complementar ao [Barbosa et al. 2019], este trabalho apresentou os cenários levantados através de uma entrevista com 9 profissionais de empresas distintas, além disso, um aprofundamento nas situações de conflitos com a Comunicação Não Violenta.

O levantamento de cenários de conflitos realizado contribuiu para o entendimento e identificação da diversidade de conflitos que acontecem nas equipes de desenvolvimento de *software*, assim como os seus efeitos para o projeto. Após a análise dos cenários levantados, buscou-se testar o *template* para apoiar na resolução das situações conflitos com a CNV. A validação contou com um encontro individual com o profissional, seguindo como roteiro a releitura do cenário apontado pelo profissional e a explicação sobre a CNV incluindo as suas etapas.

A partir da validação junto aos profissionais, foi possível observar que a CNV envolve autoconhecimento e empatia, então exige muita prática e o próprio interesse/disposição do indivíduo em reduzir hábitos comuns de fala carregados de julgamentos e avaliações que corroboram com o surgimento de conflitos. Além disso, os profissionais se demonstraram com dificuldades em identificar as emoções e sentimentos despertados ao detalhar as situações de conflitos.

Como trabalhos futuros sugerimos a criação de uma base de cenários de conflitos, incluindo os levantados neste trabalho, colocando-os classificados de forma que possa auxiliar o seu uso para promover uma Aprendizagem Baseada em Problemas (PBL) em disciplinas relacionadas a área da computação. Outra possibilidade, refere-se ao uso de aplicações com o uso de avatares expressando diferentes emoções para potencializar no relato de conflitos e aumento de vocabulário de sentimentos e emoções.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.
- Lenberg, P., Feldt, R., & Wallgren, L. G. (2014). Towards a behavioral software engineering. In Proceedings of the 7th international workshop on cooperative and human aspects of software engineering (pp. 48-55). ACM.
- Pohl, Klaus; Rupp Chris. (2011). Fundamentos da Engenharia de Requisitos. 1 ed., vol. 3, CA.
- Camacho, C. (2005). Gerenciando conflitos em reuniões: uma estratégia para a elicitação de requisitos de software (Doctoral dissertation, Dissertação de Mestrado- Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, PUC-Rio, 168 páginas).
- Karn, J. S., & Cowling, A. J. (2008). Measuring the effect of conflict on software engineering teams. Behavior research methods, 40(2), 582-589.
- Rosenberg, M. B. (2006). Comunicação não-violenta: técnicas para aprimorar relacionamentos pessoais e profissionais. Editora Agora.
- Baesler, E. J., & Lauricella, S. (2014). Teach peace: Assessing instruction of the nonviolent communication and peace course. Journal of Peace Education, 11(1), 46-63.

- Garcia, S., Mercedes, M. et al. (2017). Influência da aplicação da comunicação não violenta na entrega de resultados de uma avaliação de desempenho a funcionários administrativos (tese de bacharel, Quito).
- Nosek, M., Gifford, E. J., & Kober, B. (2014). Nonviolent Communication training increases empathy in baccalaureate nursing students: A mixed method study.
- Robbins, S. P. (1974). Managing organizational conflict: A nontraditional approach. NJ, Prentice-Hall.
- Kofman, F. (2004). Metamanagement: o sucesso além do sucesso. Elsevier.
- Jehn, K. A., & Bendersky, C. (2003). Intragroup conflict in organizations: A contingency perspective on the conflict-outcome relationship. *Research in organizational behavior*, 25, 187-242.
- Barros, I. L. (2015). Comunicação Não-Violenta como perspectiva para a paz. *Ideias e Inovação-Lato Sensu*, 2(3), 67.
- Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, 25(4), 557-572.
- Méndez Fernández, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., ... & Männistö, T. (2016). Naming the pain in requirements engineering: contemporary problems, causes, and effects in practice.
- Von W., Gresse C., Carvalho O., and Battistella. (2013). Ensinar a gerência de equipes em disciplinas de gerência de projetos de software. *Revista Brasileira de Informática na Educação*, 21(1), 15-22.
- Muszkat, M. (2003). Mediação de conflitos: pacificando e prevenindo a violência. Summus Editorial.
- Robbins, Stephen P.(2010). Comportamento Organizacional11ª Ed.. São Paulo - Pearson Prentice Hall.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Springer Science & Business Media.
- Barbosa, A. F., & Lencastre, M. (2019). Identificando cenários de conflitos na Engenharia de Requisitos visando aplicação da Comunicação Não Violenta. 22nd Workshop on Requirements Engineering.
- Barbosa, A F. (2019). Artefatos instrucionais para a análise de conflitos na Engenharia de Requisitos com base na CNV. Diss. Dissertação de Mestrado- Programa de Pós-Graduação Acadêmica em Engenharia de Computação, Escola Politécnica de Pernambuco - Universidade de Pernambuco, POLI- UPE.

Relacionando requisitos de software e competências de recursos humanos através de modelos organizacionais: uma abordagem visando o alinhamento organizacional

**Henrique Prado de Sá Sousa¹, Eduardo Kinder Almentero²,
Julio Cesar Sampaio do Prado Leite³**

¹Departamento de Informática – Universidade Federal do Estado do Rio de Janeiro
(UNIRIO)
Rio de Janeiro, RJ, Brasil

²Departamento de Computação – Universidade Federal Rural do Rio de Janeiro
(UFRRJ)
Seropédica, RJ, Brasil

³Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio)
Rio de Janeiro, RJ, Brasil

hsousa@uniriotec.br, almentero@ufrrj.br, www.inf.puc-rio.br/~julio

Abstract. *Organizational alignment is a quality related to organizational performance. Many actions are implemented to help alignment, especially when targeting strategies, but achieving it is still a complex task. Human resources are crosscutting components in an organization, so they are key elements for successful alignment because they act on tasks and make important decisions for the organization. In turn, IT offers computational support for process execution. Therefore, there is an intercession between these areas in everyday tasks and their alignment can be beneficial when oriented towards the organizational goals. Organizations have requirements that can be achieved by people and/or technology. Alignment failures in the HR structure can be mitigated by IT, expanding the chances of success. In this paper, we investigate the relationships between human and software requirements to better prospect the ways of aligning them. For this, we elicit software requirements using as input an example of an organizational model incremented with the modeling of human competences, that are required for the organizational actors. Posteriorly, we cross human competence information with software requirements to identify and evaluate the correlations to find relevant elements to help in decision-making in IT and HR actions.*

Resumo. *O alinhamento organizacional é uma qualidade relacionada ao desempenho organizacional. Muitas ações são implementadas para auxiliar no alinhamento, especialmente quando voltado para o alcance de estratégias, entretanto, alcançá-lo prossegue sendo uma tarefa difícil. Os recursos humanos, por serem componentes transversais em uma organização, são elementos chave para o sucesso do alinhamento porque atuam em tarefas e tomam decisões importantes para a organização. Por sua vez, a TI oferece*

suporte computacional para a realização de processos. Portanto há uma intercessão entre estas áreas nas tarefas cotidianas e seu alinhamento pode ser benéfico quando orientados para os objetivos da organização. As organizações possuem requisitos que podem ser satisfeitos por pessoas e/ou tecnologia. As falhas de alinhamento na estrutura de RH podem ser mitigadas pela TI ampliando as possibilidades de sucesso da organização. Neste artigo, investigamos as relações entre requisitos humanos e de software para melhor prospectar as formas de alinhamento das duas áreas. Para isso, elicitamos requisitos de software usando como insumo um exemplo de modelo organizacional, incrementado com a modelagem de competências humanas que são requeridas aos atores organizacionais. Posteriormente cruzamos as informações de competências humanas com os requisitos de software para identificar e avaliar as correlações visando encontrar elementos relevantes para auxiliar na decisão em ações de TI e de RH.

1 Introdução

As competências presentes em uma organização representam o seu potencial de realização e possibilitam o alcance de seus objetivos [Pralhad e Hamel 1990]. A gestão de competências normalmente é um papel da área de Recursos Humanos (RH), que busca administrá-las de forma alinhada aos objetivos estratégicos, ampliando a capacidade de sucesso e sobrevivência, especialmente em ambientes competitivos [Penrose 1959].

É compreensível que, fundamentalmente, a área de RH vislumbre o conceito de competências partindo dos indivíduos que compõem o capital humano organizacional [Fleury e Fleury 2001]. De fato, as pessoas desempenham papéis fundamentais na organização, especialmente, papéis decisórios e tarefas sensíveis, as quais demandam competências específicas que auxiliam a obtenção de resultados satisfatórios.

No entanto, os processos organizacionais vêm sido enriquecidos com tecnologia há décadas, impelindo a reengenharias de processos com mudanças profundas, em especial, com a absorção de responsabilidades por parte de softwares, o que altera a estrutura de competências da organização [Mascarenhas et al. 2005].

Além de transformar a forma de trabalho de áreas consolidadas como a de RH, a área de TI obtém seu papel estratégico na organização, sendo corresponsável pela administração de competências organizacionais que são entregues a partir da estrutura tecnológica, em especial os softwares [Coltman et al. 2015], [Stirna et al. 2012].

Com o avanço tecnológico nos processos organizacionais, algumas competências tendem a ser compartilhadas e/ou absorvidas por softwares, entretanto, a inserção de elementos tecnológicos pode desdobrar em novas demandas de competências para os recursos humanos, os quais atuarão em conjunto com a tecnologia.

Em meio às transformações constantes estão os riscos inerentes, os quais podem ser reduzidos a partir de mecanismos teóricos que auxiliem a projetar mudanças com menor risco possível. Um dos maiores desafios das organizações é manter o alinhamento de sua estrutura em determinado grau de sinergia para obter o máximo desempenho. O maior desafio consiste em manter diferentes áreas atuando em conjunto

de forma harmônica, o que sobressalta a importância do contexto social, da comunicação e da atuação de papéis estratégicos ao alinhamento.

Há de se observar tanto o alinhamento de cada parte, como o alinhamento do conjunto, especialmente em favor dos objetivos organizacionais. Neste interim, delinear as competências necessárias para satisfazer as demandas organizacionais é uma tarefa importante. Para auxiliar o profissional no estudo e avaliação do complexo domínio organizacional, a modelagem é de suma importância, pois auxilia não somente no registro e organização de informações, mas serve de instrumento de entendimento do domínio, na análise de cenários, definição de estratégias, e auxílio em decisões. Portanto, podem ser utilizados como insumo em projetos com diferentes finalidades, incluindo projetos de software.

Diversas propostas de linguagens para modelagem organizacional foram desenvolvidas (ex. [Amyot e Mussbacher 2002], [Braubach et al. 2010], [Jander et al. 1993], [OMG 2011], [Scheer 2000], [Singh e Woo 2009] , [Yu 1995]), entretanto, não exploram conceitos específicos provenientes dos diferentes domínios organizacionais, como os utilizados neste trabalho, provenientes do domínio de RH. Comumente, a modelagem organizacional representa os recursos humanos como “atores” os quais são responsáveis por tarefas e objetivos. No entanto, o contexto de RH é mais complexo, pois envolve atributos do indivíduo que são capazes de afetar os resultados de um processo, o que as torna, por consequência, informações relevantes no contexto estratégico.

[Yu 1995] e [Singh e Woo 2009] exploram características humanas visando o alinhamento, mas não aprofundam no domínio específico de RH para representar com maior fidelidade as questões que são abordadas pela área no contexto do alinhamento estratégico. A não inclusão de informações deste domínio implica na ausência de notação para a sua modelagem, o que consequentemente gera a impossibilidade de relacionamento com os outros elementos do modelo. Isso impede a correlação destas informações com outras que são relevantes na elicitação de requisitos de software, o que, consequentemente, afetará, dentre outras, as qualidades de completeza e detalhamento dos requisitos.

O objetivo deste artigo é realizar uma análise do potencial de uso das informações sobre competências de recursos humanos presentes em modelos organizacionais como insumo para a engenharia de requisitos. Para isso, utilizaremos um exemplo a fim de evidenciar as interseções de competências humanas e requisitos de software. Posteriormente avaliamos os benefícios dessas informações visando a automatização de tarefas e apoio computacional.

O presente artigo está dividido nas seguintes seções: a seção 2 apresenta conceitos básicos relacionados à visão organizacional; a seção 3 descreve resumidamente a teoria de competências e apresenta como a linguagem GPI-RH permite a modelagem de competências através de um padrão de modelagem; a seção 4, apresenta os modelos utilizados neste trabalho como insumo para a definição de requisitos e posterior cruzamento com as informações de competências. As informações são apresentadas em tabelas que auxiliam a visualização da correlação dos dados, seguida de uma discussão dos resultados; na seção 5 são apresentadas as conclusões e trabalhos futuros.

2 O contexto organizacional

Organizações podem ser entendidas como um conjunto de perspectivas normalmente formalizadas como departamentos ou gerências. O alinhamento organizacional é um conceito qualitativo resultante da contribuição de cada perspectiva visando os objetivos organizacionais. Definir e avaliar o alinhamento organizacional é uma atividade complexa devido ao grande número de variáveis que compõem o seu contexto.

A divisão da organização em departamentos auxilia a compartilhar as responsabilidades, entretanto, também pode contribuir com o distanciamento entre as partes que estabelecem estratégias para melhorar-se e cumprir apenas os objetivos que lhe são diretamente atribuídos. Desta forma, o alinhamento organizacional pode ser dividido em duas abordagens: alinhamento vertical, definido pelas ações que são realizadas por um departamento visando satisfazer os objetivos organizacionais; e alinhamento horizontal, definido por ações que visam alinhar os departamentos para que atuem em conjunto. O resultado do trabalho conjunto de todos os departamentos define o desempenho organizacional, portanto as organizações investem em ações de alinhamento [Salimian et al. 2012].

Segundo Santos (1998) “... traçar estratégias é um processo de entrelaçamento de tudo que é necessário para administrar uma organização”. Portanto, na prática, o alinhamento aos objetivos estratégicos de uma organização é difícil de alcançar levando organizações a falhar na implementação de suas estratégias [Kaplan e Norton 2001].

Neste trabalho buscamos definir uma forma para alinhar as áreas de RH e de TI. Do ponto de vista de RH, as competências humanas são requisitos para a satisfação dos objetivos através da realização de tarefas em processos da organização. As estratégias orientadas para competência consistem em manter a estrutura humana capaz de entregar as competências para satisfazer os objetivos estratégicos da organização. Entretanto, o desalinhamento de competências é comum sendo necessário identificá-los para corrigir [Kaplan e Norton 2003], [Becker e Gerhart 1996].

Neste contexto, os softwares organizacionais são ferramentais que podem suprir a ausência de competências de RH transformando-as em competências de TI. Do ponto de vista da TI, em especial da engenharia de requisitos, todo o domínio é fonte de informação [Leite 1993] para a elicitação de requisitos o que destaca o vasto campo multidisciplinar no qual a TI se debruça para apoiar processos de diferentes naturezas.

Os requisitos de RH são demandas organizacionais, as quais podem ser insumos para o desenvolvimento de softwares de forma estratégica para mitigar dificuldades na estrutura de RH.

A Figura 1 resume os conceitos abordados nesta sessão:



Figura 1 – Alinhamento de perspectivas organizacionais, adaptado de [Sousa e Leite 2017a]

3 A modelagem de competências em modelos organizacionais

Os recursos humanos estão presentes em todos os níveis da hierarquia organizacional e ocupam cargos comumente determinados por um perfil profissional o qual se julga capacitado a cumprir com as responsabilidades que lhe serão atribuídas. Esses perfis comumente estão relacionados a determinados papéis pré-estabelecidos e são definidos por um conjunto de quesitos, os quais se espera que o recurso humano possua, de tal forma que esteja apto a desempenhar o papel organizacional.

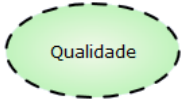
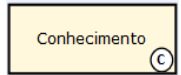
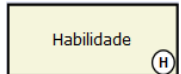
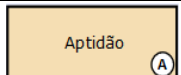
As competências são resultados alcançados através da execução de tarefas sob determinado desempenho [Fleury e Fleury 2004]. Essas tarefas demandam um conjunto de Conhecimentos, Habilidades e Aptidões (conhecido como CHA ou KSA (Knowledge, Skill, Ability) em inglês) para que possam ser realizadas a contento. Na literatura especializada, as competências são diretamente definidas em termos de CHA [Stevens 1994] [Tripathi e Agrawal 2014], embora existam outras abordagens (ex. [Carbone et al. 2005], [Draganidis e Mentzas 2006]).

As competências organizacionais ficam delimitadas pela capacidade de seus recursos, porém, observa-se que a união de diferentes competências é capaz de gerar outras competências. Através dos resultados do trabalho de múltiplas equipes em múltiplos processos, a organização pode alcançar competências altamente agregadas [Fleury e Fleury 2004] que formam seu portfólio [Prahalad e Hamel 2006], habilitando-a a atuar em determinado nicho do mercado. Portanto, existem competências em diferentes níveis de abstração.

Nos modelos organizacionais, os recursos humanos podem ser definidos em termos dos elementos CHA que possuem, enquanto as tarefas organizacionais são definidas em termos de CHA que requerem. Desta forma, pode ser modelado um perfil de determinado recurso humano (definido como Perfil Real) e um perfil de requisitos para as tarefas organizacionais (definido como Perfil Requerido). Por consequência, é possível construir Perfis Requeridos de determinados papéis, bem como de processos, processos abstratos, competências e até objetivos organizacionais. Este desdobramento dos mapas de CHA permite avaliar o potencial alinhamento de recursos humanos através do cruzamento dos Perfis Reais e dos Perfis Requeridos [Sousa e Leite 2017a].

Para representar esses elementos, utilizamos a linguagem GPI-RH¹ [Sousa e Leite 2017b] que introduz no âmbito da modelagem organizacional, conceitos específicos do domínio de RH. A linguagem utiliza-se de padrões do domínio, os quais são relacionados especialmente ao alinhamento organizacional. Neste trabalho utilizamos o Padrão de Competências [Sousa e Leite 2017b] para modelar a relação de competência e os respectivos mapas de CHA. A Tabela 1 apresenta a notação do padrão, ilustrado na Figura 2.

Tabela 1. Notação para o Padrão de Competência

Conceito/Representação	Definição
	Representa uma qualidade que pode ser satisfeita em diferentes níveis, dependendo da visão do avaliador.
	Representa determinado conhecimento necessário para a organização ou um atributo de um recurso humano.
	Representa determinada habilidade necessária para a organização ou um atributo de um recurso humano.
	Representa determinada aptidão necessária para a organização ou um atributo de um recurso humano.
	Representa uma contribuição positiva.
	Representa uma relação “E”, denotando uma decomposição.

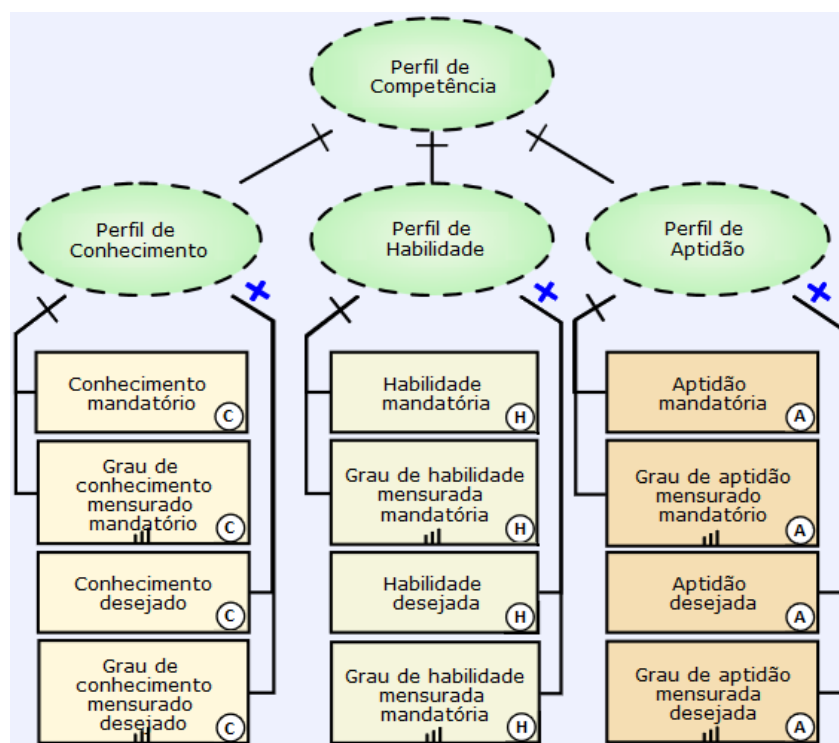


Figura 2. Padrão de Perfil de competências

¹ A ferramenta GPI pode ser acessada em http://ger.inf.puc-rio.br:8080/oryx/GPI_editor.xhtml

Os perfis são conceitos qualitativos que podem pertencer a determinado recurso humano ou é demandado para determinado ator, tarefa, processo ou competência, podendo tomar diferentes graus baseado em suas operacionalizações. Essas operacionalizações são os conhecimentos, habilidades e aptidões, que mapeiam os respectivos perfis, sendo específicos ao determinado elemento para o qual o padrão de competência é instanciado.

Cada operacionalização CHA pode ser definida como mandatórias ou desejáveis. As operacionalizações mandatórias são necessárias para fazer (make) minimamente a competência existir, portanto, estão ligados pelo relacionamento AND. As operacionalizações desejáveis são opcionais e contribuem positivamente para o perfil, se presente em algum nível. Portanto, eles estão ligados por um relacionamento HELP (+). Não são esperados relacionamentos do tipo HURT (-) em um perfil que normalmente é de avaliação positiva, entretanto, não há impedimentos de uso.

Ainda existem as operacionalizações que são obrigatórias ou desejadas, e que são mensuráveis. Elas são marcadas com três barras pequenas, na parte inferior do elemento do gráfico, e devem obrigatoriamente estar relacionadas a indicadores. As operacionalizações do CHA são expressas em uma notação diferente das descritas no NFR Framework [Chung et al. 2000], para se adequar melhor ao formato utilizado na linguagem GPI-RH.

3 Mapeando requisitos de software e competências de recursos humanos

Os requisitos de um software descrevem o seu comportamento, isto é, o que ele deve fazer e como isto deve ser feito para satisfazer as necessidades dos interessados. Durante a apreciação de modelos organizacionais com viés de elicitação de requisitos é comum o enfoque nas tarefas descritas, pois estas descrevem um comportamento operacional. Entretanto, as competências registradas em alguns destes modelos (no uso da linguagem GPI-RH) também são uma importante fonte de informação, pois detalham as características necessárias e/ou recomendadas aos atores para que os resultados sejam entregues de forma satisfatória.

Uma vez que a automatização do processo implica em transferir para o software a responsabilidade pela execução de parte ou de todas as tarefas do processo, é importante garantir a implementação adequada das competências necessárias para entrega satisfatória dos resultados. Caso contrário, é possível que o software não apresente as características necessárias para atender às necessidades que motivaram seu desenvolvimento.

Ademais, acreditamos que a consideração das competências no processo de Engenharia de Requisitos possibilita um entendimento maior das necessidades relacionadas às tarefas do processo organizacional analisado. Ao elucubrar as competências no contexto de requisitos estamos ampliando os pontos de vista da análise que é feita tradicionalmente. Ao invés de considerar apenas o que e como deve ser feito, estamos considerando também as características necessárias aos atores do processo, para que a chance da entrega corresponder ao esperado seja maior.

Desta forma, ao automatizar o processo através de software, considerando as competências, estamos tornando o novo ator do processo, o software, mais apto a entregar aquilo que se espera dele. A fim de preencher a lacuna explicitada, propomos neste trabalho uma abordagem baseada no mapeamento de competências em requisitos,

criando um rastro que deve ser mantido ao longo do processo de Engenharia de Requisitos. Este rastro possibilita identificar se há impacto da competência nos requisitos, além de determinar este impacto.

Considerando um processo tradicional de Engenharia de Requisitos [Leite 1993], posicionamos a abordagem aqui proposta na etapa de elicitação requisitos (Figura 3), assumindo que serão utilizados como fonte de informações modelos organizacionais que permitam a representação de competências, como o GPI-RH. Os eventuais impactos identificados devem ser considerados para as etapas posteriores, de modelagem e análise. É importante que ressaltar que este posicionamento é apenas uma sugestão. É perfeitamente razoável, caso a especificação de requisitos já esteja concluída, por exemplo, realizar o mapeamento utilizando modelos de requisitos.

O processo de mapeamento adotado neste trabalho utiliza como insumo um modelo organizacional contendo um processo relacionado às competências humanas (Figura 4), as quais são mapeadas através de um modelo padrão de perfil de competências (Figura 5 e Figura 6). Posteriormente identificamos requisitos através da interpretação do processo organizacional (Tabela 2). A lista de requisitos está relacionada às tarefas de processo, as quais demandam do recurso humano determinada configuração de CHA para serem realizadas. As tarefas de um processo fomentam a relação indireta entre os requisitos de software e o perfil de competências (**Tabela 3**).

Foi utilizado no exemplo o processo “Requisição de Análise de Crédito”, obtido através de uma simplificação do processo descrito em [Diirr et al. 2010]. O processo é ilustrado na Figura 4 (os conceitos representados na figura se encontram descritos em vermelho); O modelo também possui outros conceitos mapeados, como as competências, objetivos organizacionais, e o recurso de “agrupamento”, o qual delinea o conjunto de tarefas necessárias para implementar uma competência. A “entrega” da competência satisfaz/contribui com os objetivos relacionados. Neste processo, todas as tarefas são realizadas manualmente e considera-se a possibilidade de automatizá-las.

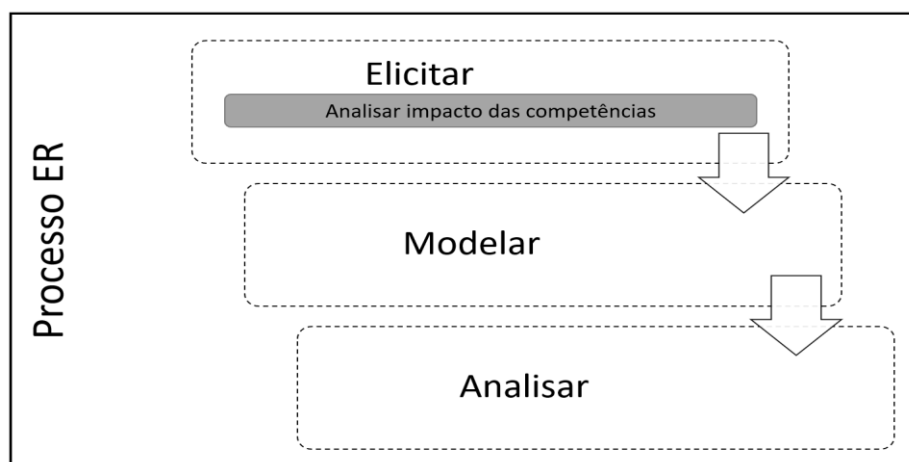


Figura 3. Abordagem inserida no processo ER

“Requisição de Análise de Crédito” descreve um curso de tarefas usual do domínio financeiro para concessão de empréstimo e se inicia quando uma proposta de crédito é recebida pelo Analista de crédito. Ele verifica a existência de cadastro do cliente. Caso o cadastro não exista, o Analista realiza o cadastro. Caso o cadastro exista e esteja desatualizado, o Analista atualiza o cadastro. Se o cadastro existir e estiver

atualizado, o Analista verifica se o cliente possui crédito suficiente para cobrir a sua solicitação. Se não houver crédito suficiente, a proposta de crédito é cancelada, senão, o valor da proposta é abatido do valor do crédito. Posteriormente, o Analista calcula as taxas de contrato e juros de acordo com o perfil do cliente e então redige a proposta de contrato. O Gerente de empréstimo analisa proposta de contrato visando identificar características que apontem um contrato de risco em potencial. Caso o contrato seja de risco, ele é cancelado, porém, se for possível alterá-lo de forma a torná-lo aceitável o Gerente realiza os ajustes. Se não houver riscos substanciais ele é aprovado. Ao final, o Gerente de contrato comunica o resultado da proposta de crédito.

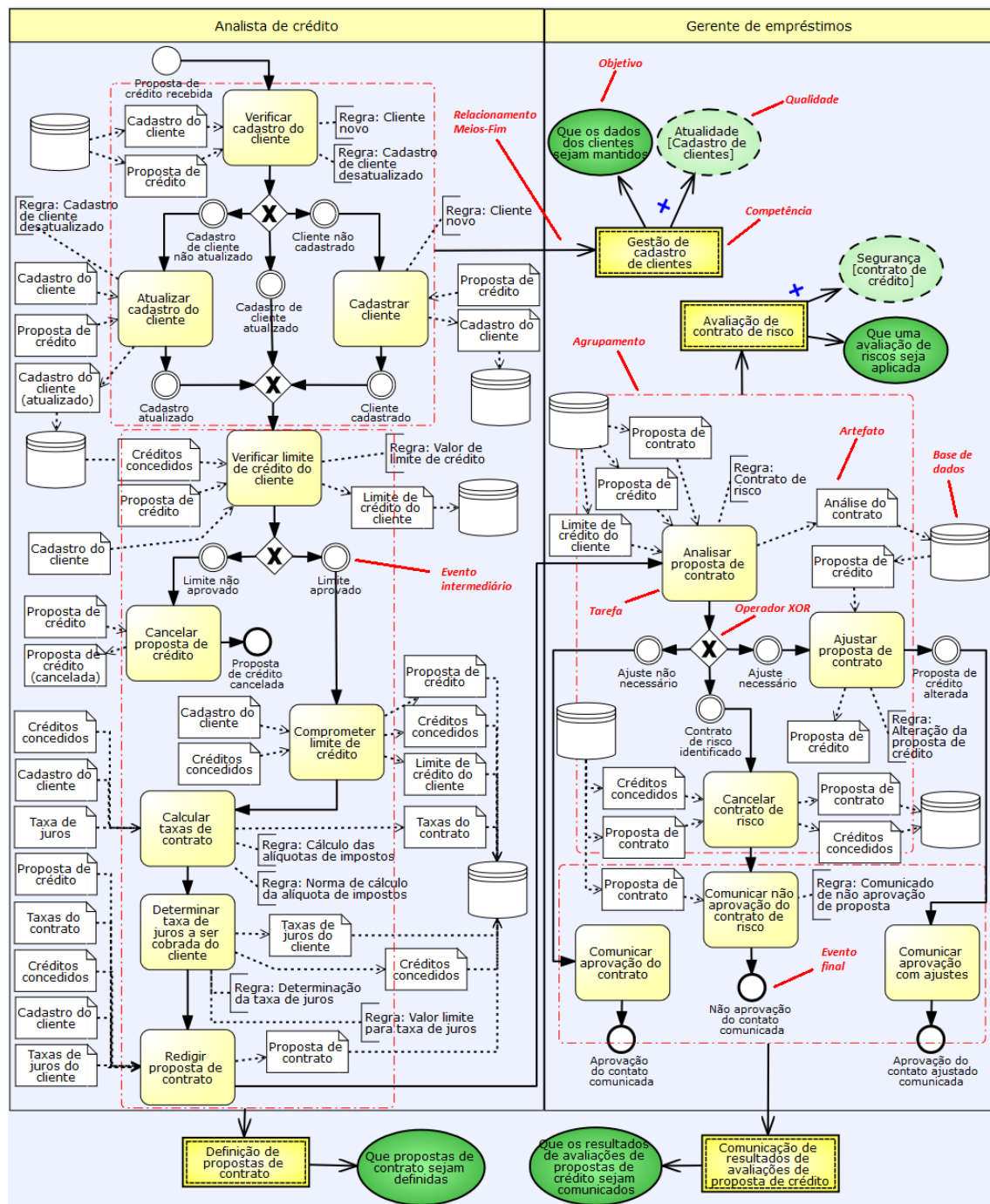


Figura 4. Processo de Avaliação de Proposta de Crédito

Através de um exercício simples de análise das tarefas do modelo, extraímos uma lista de potenciais requisitos do software (Tabela 2). O mapeamento pode ser realizado com requisitos em outros níveis de detalhamento. Neste exemplo, a fim de facilitar o entendimento, utilizamos requisitos em um nível alto de abstração.

Tabela 2. Lista de requisitos

Nº	Nome	Descrição
1	Calcular as taxas associadas ao contrato	O sistema deve calcular as alíquotas de impostos que incidam sobre o contrato de crédito de acordo com a legislação vigente.
2	Calcular juros referentes ao contrato	O sistema deve calcular os juros que serão cobrados do cliente utilizando as tabelas de juros definidas pela instituição.
3	Avaliar proposta de contrato	O sistema deve aplicar os critérios estabelecidos pela instituição para determinar se o contrato será aprovado ou reprovado.
4	Sugerir ajuste de contrato	Caso o contrato seja reprovado, o sistema deve sugerir as correções necessárias.

Analisando o processo é possível identificar, por exemplo, que para executar as tarefas “Analisar proposta de contrato”, “Ajustar proposta de contrato” e “Cancelar contrato de risco” o ator “Gerente de Empréstimo” deve possuir a competência descrita pelo perfil de competência (Figura 5 e Figura 6) “Avaliação de contrato de risco”. Este perfil detalha a competência necessária decompondo-a em perfil de conhecimento, perfil de habilidade, e perfil de aptidão. Assim, verificamos que as operacionalizações “Flexibilidade”, “Perspectiva (pontos de vista)” e “Criteriosidade”, são, por exemplo, algumas das operacionalizações obrigatórias para este perfil de competência.

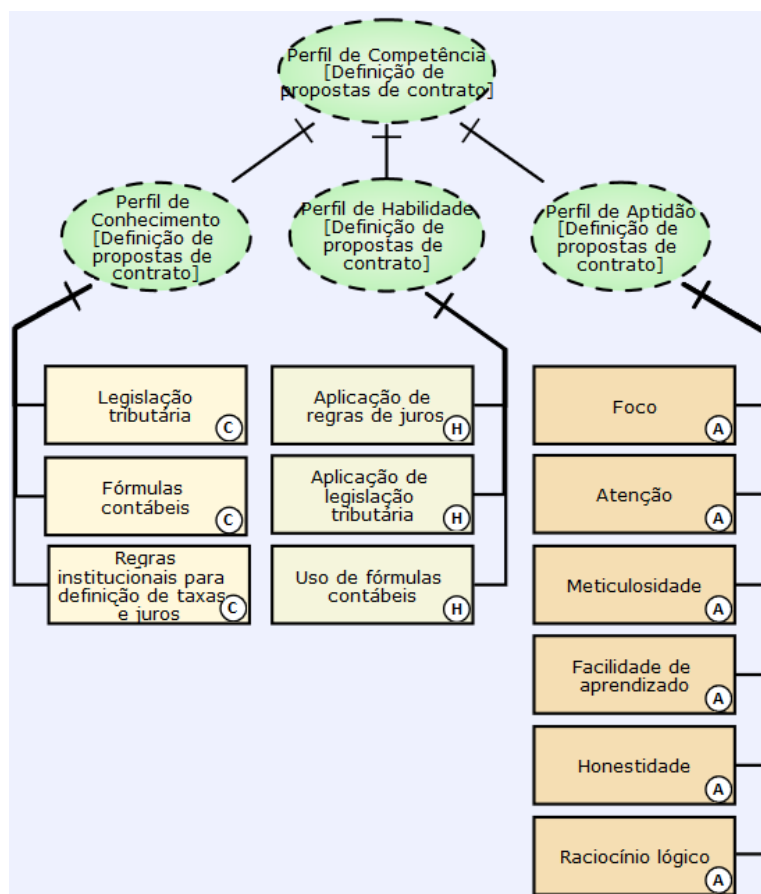


Figura 5. Modelo de Perfil de Competência "Definição de propostas de contrato"

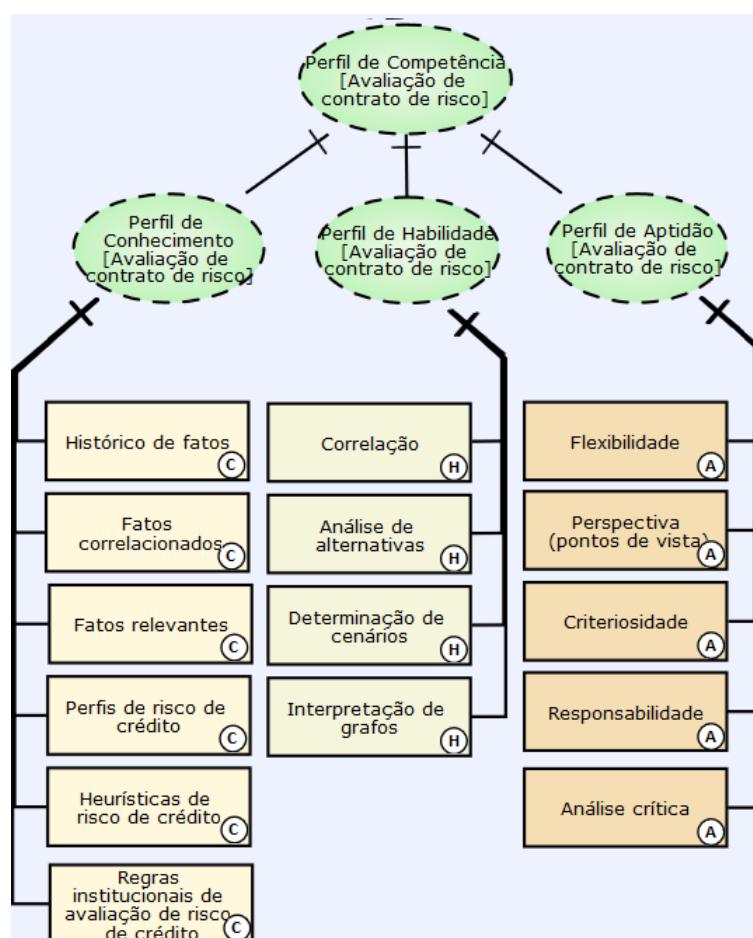


Figura 6. Modelo de Perfil de Competência "Avaliação de contrato de risco"

A partir dos modelos de perfil de competência (Figura 5 e Figura 6) e da lista de requisitos é possível realizar o primeiro passo do mapeamento entre competências, CHA e requisitos do software.

Para realizar este passo é importante observar a relação entre os requisitos, as tarefas e as competências presentes no modelo organizacional. No exemplo utilizado verificamos que o requisito “Calcular as taxas associadas ao contrato” está relacionado a competência “Calcular as taxas associadas ao contrato” e, portanto, este mapeamento foi realizado na Tabela 3. O mesmo raciocínio foi aplicado no requisito “Sugerir ajuste de contrato”, o qual foi relacionado a competência “Avaliação de contrato de risco”.

O segundo passo do mapeamento consiste na avaliação das CHA afim de verificar aquelas que estão relacionadas aos requisitos mapeados as competências. No caso do requisito “Calcular as taxas associadas ao contrato” a operacionalização de Perfil de Conhecimento “Legislação tributária” foi considerada desnecessária e, por isso, foi removida. Por outro lado, as operacionalizações de Perfil de Conhecimento “Fórmulas contábeis” e “Regras institucionais para definição de taxas e juros” foram consideradas necessárias e, por isso, permaneceram no mapeamento.

Esta mesma lógica foi aplicada às operacionalizações do Perfil de Habilidade, em que a operacionalização “Aplicação de regras de juros” foi removida, enquanto as operacionalizações “Uso de fórmulas contábeis” e “Aplicação de legislação tributária”

foram mantidas. Já no caso das do Perfil de aptidões, foi removida a operacionalização “Meticulosidade”. O mesmo raciocínio foi utilizado para os demais requisitos da lista e com isso completamos a **Tabela 3**.

Tabela 3. Mapeamento entre requisitos e CHA

Competência	Requisito(s)	Conhecimentos	Habilidades	Aptidões
Definição de propostas de contrato	Calcular as taxas associadas ao contrato	- Fórmulas contábeis - Regras institucionais para definição de taxas e juros.	- Aplicação de regras de juros - Uso de fórmulas contábeis.	- Foco - Atenção - Facilidade de aprendizado - Honestidade - Raciocínio lógico.
Definição de propostas de contrato	Calcular juros referentes ao contrato	- Legislação tributária - Fórmulas contábeis	- Uso de fórmulas contábeis. - Aplicação de legislação tributária	- Foco - Atenção - Facilidade de aprendizado - Meticulosidade - Honestidade - Raciocínio lógico.
Avaliação de contrato de risco	Avaliar proposta de contrato	- Histórico de fatos - Fatos Correlacionados - Fatos relevantes - Perfis de risco de crédito - Heurísticas de risco de crédito - Regras institucionais de avaliação de risco de crédito	- Correlação - Análise de Alternativas - Determinação de cenários - Interpretação de grafos	- Flexibilidade - Perspectiva (pontos de vista) - Criteriosidade - Responsabilidade - Análise crítica
Avaliação de contrato de risco	Sugerir ajuste de contrato	- Histórico de fatos - Fatos Correlacionados - Heurísticas de risco de crédito - Regras institucionais de avaliação de risco de crédito	- Correlação	- Perspectiva (pontos de vista) - Criteriosidade - Responsabilidade

Uma vez mapeadas as relações entre os requisitos extraídos do processo e as competências é possível avaliar o impacto das operacionalizações de Perfil de Conhecimento, Habilidade e Aptidão. Durante a análise destes impactos, identificamos casos similares recorrentes, que levaram a identificação de três padrões:

- Não há impacto;
- O impacto incorre na modificação de um requisito funcional ou não funcional ou adição de um novo;
- O impacto incorre na modificação do próprio processo que está sendo automatizado.

As operacionalizações “Foco”, “Atenção”, “Honestidade” e “Raciocínio lógico” do Perfil de Aptidão, por exemplo, que estão relacionadas com o requisito “Calcular taxas associadas ao contrato” são absorvidas uma vez que o processo for automatizado, pois as características intrínsecas a um sistema computacional dispensam a necessidade de operacionalizar essas características humanas.

A operacionalização de Perfil de Aptidão “Facilidade de aprendizado”, por exemplo, relacionada ao requisito “Calcular juros referentes ao contrato” é necessária devido a volatilidade das regras de juros da instituição financeira, a qual deve adequar-se

constantemente a realidade do mercado. Uma alteração nestas regras requer um novo entendimento, para que o resultado seja entregue de maneira satisfatória. Neste caso, a operacionalização de Perfil de Aptidão terá impacto nos requisitos do software, uma vez que este deve ser flexível (RNF) para acomodar eventuais mudanças nas regras de cálculo de juros.

Por fim temos o caso em que a relação entre Perfil de Competência e requisito tem impacto no próprio processo que está sendo automatizado. A operacionalização de Perfil de aptidão “Responsabilidade”, que está relacionada ao requisito “Avaliação de contrato de risco”, está ligada a identificação da autoria da tomada de decisão. Neste caso, uma vez que a tomada de decisão seja delegada ao software, o próprio processo precisará ser adaptado para que a autoria possa ser atribuída a um agente humano responsável. Isto pode ser feito, por exemplo, através da inclusão de uma nova tarefa em que a alternativa escolhida pelo software durante a tomada de decisão seja aprovada ou não por um responsável (humano).

4 Conclusão

Neste trabalho defendemos a ideia de que relacionar as competências dos recursos humanos aos requisitos extraídos de modelos organizacionais, em tempo de desenho, possibilita uma análise mais profunda e detalhada das necessidades que o software deve atender.

Demonstramos através de um exemplo modelado na linguagem GPI a interação entre informações de diferentes domínios (TI e RH), que se afetam mutuamente na inserção de softwares em processos organizacionais.

Destacamos que esse estudo é importante, uma vez que a engenharia de requisitos não se atém somente ao software, mas perpassa por todo conhecimento do domínio, o qual, no âmbito organizacional, possui vasto campo multidisciplinar. Há a necessidade de se identificar as interfaces entre as diversas perspectivas organizacionais para que possibilite a definição recursos que auxiliem sua correlação e posterior análise.

Outros trabalhos já abordaram o conceito de competências organizacionais [Stirna et al. 2012] e de características humanas [Yu 1995] [Singh e Woo 2009], em modelos organizacionais, entretanto, não correlacionaram conceitos específicos do domínio, visando ampliar o conhecimento para contribuir na construção de softwares mais alinhados sob a perspectiva de RH.

O cruzamento de informações pode auxiliar em projetos de software, por exemplo, permitindo a priorização de requisitos a partir de demandas identificadas pela área de RH. Essas demandas poderiam ser identificadas durante a gestão de competências, com evidências de dificuldades em satisfazer determinadas tarefas de rotina, o que poderia justificar o investimento em apoio ou automatização computacional, em contrapartida a novas contratações ou consultorias. Uma das formas de se identificar requisitos poderia ser através do uso de indicadores críticos de RH que explicitam o desempenho insuficiente em tarefas específicas. Esses requisitos podem se tornar prioritários em um projeto de software.

Este trabalho se encontra delimitado a um estudo preliminar, baseado em exemplos simples. Entretanto, a investigação se justifica para evidenciar a relação dos conceitos abordados a partir da modelagem organizacional e do uso dos modelos como

fonte de informação para elicitación de requisitos. Em trabalhos futuros, aprofundaremos o estudo a partir de modelos organizacionais mais detalhados para melhor delimitar as relações e identificar potenciais padrões. A partir disso, será possível identificar melhor como apoiar a questão do alinhamento vertical entre a TI e o RH e propor com mais precisão a absorção dos elementos presentes nos mapas CHA por softwares. Outro estudo é o cruzamento do resultado de aplicações dos métodos de identificação de requisitos a partir dos modelos de processos [Azevedo et al.2009] com informações de RH presentes nos modelos organizacionais o que permitirá a correlação automatizada e geração de relatórios através da ferramenta de modelagem.

Referências

- Amyot, D., Mussbacher, G.: URN: Towards a New Standard for the Visual Description of Requirements. In: Proceedings of International Workshop on System Analysis and Modeling, SAM'2002. pp.21-37, (2002).
- Azevedo, L. G., Sousa, H. P., Souza, J. F., Santoro, F., & Baiao, F. (2009). Identificação automática de serviços candidatos a partir de modelos de processos de negócio. In *Conferencia IADIS Ibero Americana WWW/INTERNET*.
- Becker, B.; Gerhart, B.; "The Impact of Human Resource Management on Organizational Performance: Progress and Prospects"; The Academy of Management Journal, Vol. 39, N°4, pp. 779-801, 1996.
- Braubach, L., Pokahr, A., Jander, K., Lamersdorf, W., Burmeister, B.: Go4Flex: Goaloriented Process Modelling, Proc. 4th International Symposium on Intelligent Distributed Computing, (2010).
- Carbone, P., Brandão, H.P., Leite, J.B.D., Vilhena, R.H.M.P.: Gestão por competências e gestão do conhecimento, Rio de Janeiro: Ed. Fundação Getúlio Vargas, (2005).Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.
- Chung, L., Nixon, B., Yu, E., Mylopoulos, J., "Non-Functional Requirements in Software Engineering"; Kluwer Academic Publishers – Massachusetts, USA, (2000).
- Coltman, T., Tallon, P., Sharma, R.H., Queiroz, M.: Strategic IT alignment: twenty-five years, Journal of Information Technology, <https://doi.org/10.1057/jit.2014.35>, (2015).
- Diirr, T., Souza, A, Azevedo, L. G., Santoro, F.: Analyze Credit Request Model, Technical Report of Applied Informatics Department – Federal University of State of Rio de Janeiro (UNIRIO), (2010).
- Draganidis, F.; Mentzas, G.: Competency based management: a review of systems and approaches, Information management & computer security, 14(1), 51-64, (2006).
- Fleury, Maria Tereza Leme, and Afonso Fleury. "Construindo o conceito de competência." Revista de administração contemporânea 5.SPE (2001): 183-196.
- Fleury, M. T. L., Fleury, A. C. C.: Alinhando estratégia e competências, RAE-Revista de Administração de Empresas, vol. 44, n. 1, (2004).
- Jander, K., Braubach, L., Pokahr, A., Lamersdorf, W.: Goal-oriented Processes with GPMN, International Journal of Artificial Intelligence Tools 20(6):1021-1041, (2011).

- Kaplan, R.H.S., Norton, D. P. *Organização Orientada para a Estratégia*. Campus, 416 p., 2001.
- Kaplan, R.H. S.; Norton.; “Strategy Maps: Converting intangible assets into tangible outcomes”; Boston: Harvard Business School, 2003. Leite, J.C.S.P., *Livro Vivo: Engenharia de Requisitos*. (1993).
- Mascarenhas, A.O., Vasconcelos, F.C., Vasconcelos, I.F.G.: Impactos da tecnologia na gestão de pessoas: um estudo de caso, *Revista de Administração contemporânea* 9.1, pp.125-147, (2005).
- OMG.: *Business Process Model and Notation (BPMN), Version 2.0*, (2011).
- Penrose, E.T.: *The Theory of Growth of the Firm*, 1st edn, London: Basil Blackwell, (1959).
- Prahalad, C.K., Hamel, G.: The core competence of the corporation, *Strategische unternehmensplanung—strategische unternehmensführung*. Springer, Berlin, Heidelberg, pp. 275-292, (2006).
- Salimian, H., Khalili, S., Nazemi, J., & Alborzi, M. (2012). Alignment in the organizations strategy window (concentration on business strategy and operations strategy). *African Journal of Business Management*, 6(51), 12016-12022.
- Santos, F. C. A.; “Dimensões competitivas da estratégia de recursos humanos: Importância para a gestão de negócios em empresas manufatureiras”; Programa de Pós-Graduação da FGV/EAESP, São Paulo, 1998.
- Scheer, A.W., *ARIS – Business Process Modeling*, Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 2000, 218 p. 3. ed., Bibliografia: ISBN, 3-540-65835-1, (2000).
- Singh, S. N., Woo, C.: "Investigating business-IT alignment through multi-disciplinary goal concepts." *Requirements Engineering* 14.3 (2009): 177-207.
- Sousa, H.P.S., Leite, J.C.S.P.: Toward an organizational alignment modeling language: The Human Resource competency perspective, In: 2017 IEEE 19th Conference on Business Informatics (CBI). Vol. 1. IEEE, (2017a).
- Sousa, H.P.S., Leite, J.C.S.P.: Requirement Patterns for Organizational Modeling, In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, (2017b).
- Stevens, M.J. The Knowledge, skill, and ability requirements for teamwork: implications for human resource management. *Journal of Management*, Volume 20, Issue 2, Summer 1994, Pages 503-530, 1994. Stirna, J., Grabis, J., Henkel, M., Zdravkovic, J.: Capability Driven Development – An Approach to Support Evolving Organizations, In: Sandkuhl K., Seigerroth U., Stirna J. (eds) *The Practice of Enterprise Modeling. PoEM 2012*. Springer, Berlin, Heidelberg, (2012).
- Tripathi, K., Agrawal, M.: Competency Based Management, In *Organizational. Global Journal of Finance and Management*, 6(4), 349-356, (2014).
- Yu, E.: *Modelling Strategic Relationships for Process Reengineering*, Phd Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada, (1995).

Requirements Communication in Safety-Critical Systems: an extended literature overview

Jéssyka Vilela ¹, Jaelson Castro ¹, Luiz Martins ², Tony Gorschek ³, Camilo Almendra ^{1,4}

¹Universidade Federal de Pernambuco (UFPE), Brazil

²Universidade Federal de São Paulo (UNIFESP), Brazil

³Blekinge Institute of Technology (BTH), Sweden

⁴Universidade Federal do Ceará (UFC), Brazil

{jffv, jbc}@cin.ufpe.br, legmartins@unifesp.br, tony.gorschek@bth.se,
camilo.almendra@ufc.br

Abstract. Context: *Safety-critical systems (SCS) must be carefully planned since inadequate or misunderstood requirements have been recognized as the major cause of safety-related catastrophes. Objective:* *We investigate the integration and requirements communication in the requirements engineering (RE) process among different parties when developing SCS. Method:* *We used a Systematic Mapping Study as the basis for our work. Results:* *We analyze several aspects such as challenges, domain, requirements activity, languages, tools, stakeholders involved, communication format, and safety standards. Conclusions:* *This information contributes to setting up possible collaborative networks and as a reference when developing new research projects.*

1. Introduction

Safety-critical systems are mainly controlled by software nowadays [Sikora et al. 2012][Hatchiff et al. 2014]. New generations of medical devices, means of transportation (aircraft, automated trains and cars), nuclear power generating stations, banking and investment systems, as well as a growing number of automated systems rely on software to enable new functions, provide pre-existing functions more efficiently, and reduce time to service a user need as well as the effort and competence required by people providing services.

There are many cases in the literature [Leveson 2011] where inadequate or misunderstood requirements were the major cause (not coding or implementation) of a significant proportion of accidents and safety-related catastrophes. Therefore, these systems must be carefully specified, demanding more rigorous RE approaches [Leveson 2011].

RE focuses on good specification practices but has yet to find working solutions for effective requirements communication. Furthermore, the competences of requirements engineers and safety engineers normally work independently of each other and have inherently different tools and engineering practices - resulting in a lack of coordination that can compromise the quality of safety analysis and safety specifications [Vilela et al. 2017].

In this work, we investigate the approaches proposed to improve the integration of requirements communication in the RE process among different parties when developing

SCS. We adopted the systematic mapping study as a research method. We believe the results of such novel study will benefit both researchers and practitioners. The review will provide researchers with important research gaps regarding the requirements communication between safety and RE. For the industrial readership, the review will provide practitioners with useful information about the state-of-the-art and advances so far. This information contributes to setting up possible collaborative networks and as a reference when developing new research projects.

This paper is organized as follows. Section 2 presents background and related work. The research methodology adopted to conduct the mapping study is presented in Section 3. The results and the analysis related to our research questions are presented in Section 4. Our conclusions are presented in Section 5.

2. Background and Related Work

SCS are those software or system operations that, if not performed, performed out of sequence, or performed incorrectly could result in improper control functions, or lack of control functions required for proper system operation. Such problems can directly or indirectly cause or allow a hazardous condition to exist [Leveson 2011].

In order to set the scope and make clear the adopted definition of requirements communication used in this mapping study, and to ensure consistency throughout this paper, we discuss this concept in the next section.

2.1. Requirements Communication

Requirements communication is a traversal process of exchanging information [Glinz and Fricker 2015] about the requirements among all stakeholders [Bjarnason et al. 2011] involved in the system lifecycle. This concept does not only comprise the communication itself but the specification and analysis of all artifacts involved in the RE process. Since changes occur throughout the project, requirements communication must also continue during the entire life cycle [Bjarnason et al. 2011].

This process aims to achieve a shared understanding [Glinz and Fricker 2015] of the system's requirements to increase completeness and correctness of the requirements specifications. It encompasses all the activities needed to inform the stakeholders of the content, meaning and status of requirements. The elicited requirements need to be communicated, and changes to those requirements negotiated and communicated among all affected roles, e.g. requirements engineers, developers, and testers [Bjarnason et al. 2011].

2.2. Related Work

The communication of requirements among different parties in the development of SCS is critical for the quality of the system. This occurs since requirements should be understood in the same way by different roles in the development. We argue that the requirements engineers and safety engineers should collaborate, exchange information and work jointly and in an iterative way. However, they usually work independently of each other and have inherently different tools and engineering practices - resulting in a lack of coordination that can compromise the quality of safety analysis [Wang et al. 2018], and therefore, the quality of safety specifications.

Communication problems in software development were investigated by some authors such as Brady et al. [Brady et al. 2007], Pernstal [Pernstal 2015], Rasmussen and Lundell [Rasmussen and Lundell 2012], Wang et al. [Wang et al. 2018] as well as Nakamura et al. [Nakamura et al. 2016]. Although these works explore several challenges related to the integration of RE and safety, little has been done to date to perform an extensive identification and mapping, in a comprehensive manner, the state-of-the-art on the communication of requirements among different parties in the development activities/process when developing SCS. Hence, to the best of our knowledge, this is the first mapping study with such specific focus. In the next section, we detail our research protocol.

3. Research Methodology

In this section, we present the design and the execution of the mapping study. The research methodology used was based on the guidelines and template proposed by Kitchenham and Charters [Kitchenham and Charters 2007].

The focus of this review is the integration between RE and safety engineering and the requirements communication among different parties during the RE process. We included only English primary studies, published in any year until February 2018, that address in their objectives the communication in the RE process among different parties when developing SCS, related Requirements and Safety in the context of RE process, or covered Design in the relationship with Requirements and Safety.

We excluded Secondary studies, Short-papers (≤ 3 pages), Duplicated studies, Studies clearly irrelevant to the research, taking into account the research questions, Gray literature, Redundant paper of same authorship, Publications whose text was not available (through search engines or by contacting the authors), and Studies whose focus was not the communication in the RE among different parties when developing SCS or safety requirements specification.

Our study was guided by the research questions presented in Table 1. The search strategy included two types of search to find studies relevant to the scope of the review. The first type was an automatic search, using a string validated by experts on RE and SCS. The second strategy was the manual inclusion of papers well-known about requirements communication.

We developed a review protocol in which the main elements are as follows: the *se-*

Tabela 1. Research questions.

Research Question
RQ1. What challenges have been identified pertaining to the communication among engineers during the RE process when developing SCS?
RQ2. Which approaches have been proposed to improve the communication in the RE process among engineers when developing SCS?
RQ2.1. What are the types of these approaches?
RQ2.2. For which domains were these approaches proposed?
RQ2.3. What RE activities were supported by these approaches?
RQ2.4. Which requirements specification languages are used by these approaches?
RQ2.5. Which tools are used for the requirements specification?
RQ2.6. For which stakeholder were they proposed?
RQ2.7. What are the communication formats used?
RQ2.8. For what safety standards have the approaches been proposed?

lected resources chosen were Science Direct, SpringerLink, ACM Digital Library, IEEE Xplore, Scopus, and Compendex; the **search method** used web search engines; the **population** was composed of peer-reviewed publications reporting approaches to improve the communication in the RE process among parties when developing SCS; the aim of the **intervention** was to collect empirical evidence in relation to the research questions.

We developed the search string by specifying the main terms of the phenomena under investigation (SCS and requirements communication). After several iterations, we defined the search string below to search within keywords, title, abstract and full text of the publications:

- (1) (“*safety critical system*” OR “*safety critical systems*” OR “*safety-critical system*” OR “*safety-critical systems*”) AND
- (2) (“*requirements engineering*” OR “*requirements engineer*” OR “*requirements team*” OR “*requirements specification*”) AND
- (3) (“*safety requirements*” OR “*safety engineering*” OR “*safety engineer*” OR “*safety team*” OR “*safety analysis*” OR “*safety specification*”) AND
- (4) (“*communication*” OR “*integration*” OR “*interaction*” OR “*collaboration*” OR “*alignment*” OR “*understanding*” OR “*relationship*” OR “*share*” OR “*sharing*” OR “*combination*” OR “*interrelation*” OR “*interplay*” OR “*interdependency*”)

Figure 1 depicts the steps of the selection process showing the number of studies in each step. The data were extracted from the 60 primary studies using an extraction form fully aided by the StArt tool.

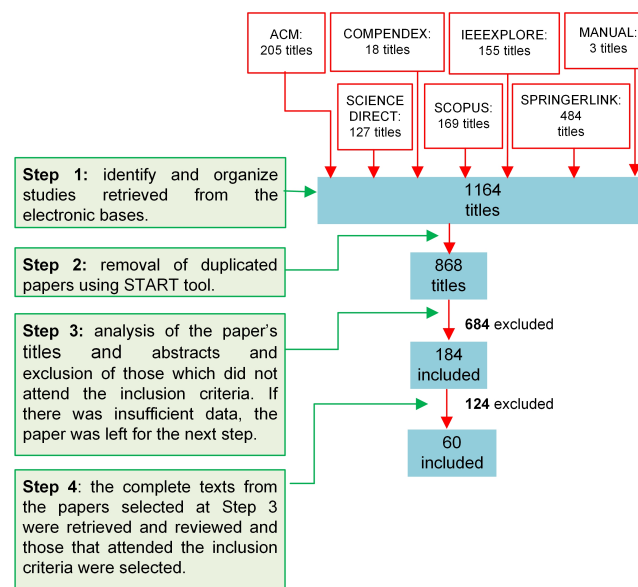


Figura 1. Paper selection flowchart.

The quality assessment (QA) of selected studies was achieved by a scoring technique to evaluate selected studies in terms of credibility, completeness and relevance. All papers were evaluated against a set of 20 quality criteria whose assessment instrument we developed and used in a previous work [Vilela et al. 2017] and described in the supplementary material.

4. Results and Analysis

A total of 60 studies satisfied the inclusion criteria and their data were extracted. The quality scores of the selected studies are presented in Table S1 on supplementary material¹. The mean of quality was 83.12%, hence, the overall quality of the selected studies is acceptable.

4.1. Overview of the Studies

The reviewed papers were published between 1994 and February 2018. From a temporal point of view (Figure 2), we can notice that the number of studies about requirements communication in SCS is low over the years. Despite the apparently increasing number of studies on this topic (peak in 2009-2011), this result corroborates the statement that the collaboration of safety analysis and RE has been somewhat neglected [Leveson 2011]. It is also worth noting that, as the search process of this review was performed in February 2018, a slight decrease in the number of publications would be expected in 2018 because some papers might have been in press.

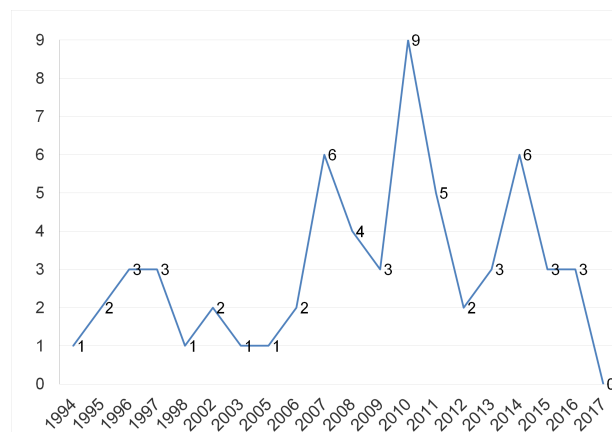


Figure 2. Temporal view of the studies.

Figure 3 presents a bubble plot distributed over three dimensions regarding three characteristics of the studies: evaluation method, research type and application context (academic, industrial or both). The left part in this figure denotes the relationship between the research type of the studies and their evaluation method. The number in a bubble represents the number of studies that present both characteristics. On the other hand, in the right part of this figure, the number in a bubble represents the number on a specific research type in a certain application context.

The results of each research question are presented and discussed in the next sections.

4.2. RQ1: What challenges have been identified pertaining to the communication among engineers during the RE process when developing SCS?

The selected studies point out many challenges as listed in Section 2 supplement material. In this section, we also discuss the details the elicited challenges.

¹Available at: www.cin.ufpe.br/~jffv/papers/wer2019

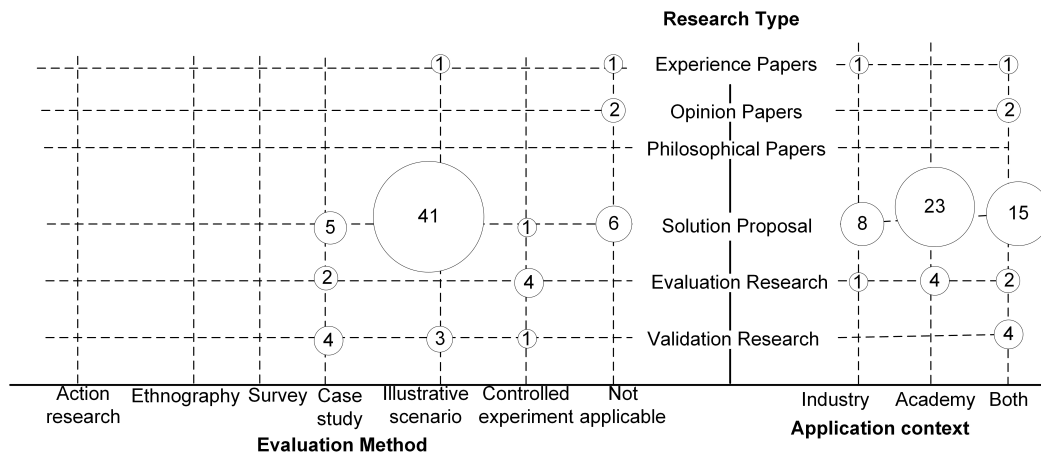


Figura 3. Bubble plot with application context, research type and evaluation method dimensions.

Many challenges of requirements communication are related to the concept of shared understanding [Glinz and Fricker 2015]. Shared understanding among a group of people has two facets: explicit shared understanding is about interpreting explicit specifications, such as requirements, design documents, and manuals, in the same way by all group members. Implicit shared understanding denotes the common understanding of non-specified knowledge, assumptions, opinions, and values. The shared context provided by implicit shared understanding reduces the need for explicit communication and, at the same time, lowers the risk of misunderstandings.

4.3. RQ2: Which approaches have been proposed to improve the communication in the RE process among engineers when developing safety-critical systems?

This research question was divided into eight sub research questions (RQ2.1 to RQ2.8) aiming to analyze many aspects of requirements communication of SCS.

4.4. RQ2.1: What are the types of these approaches?

The contribution types are reported considering the classification presented in the selected studies. The final list of contribution types are presented in Figure 4. Note that, similarly to other research questions, this question also allows a study to be included in more than one category.

The results suggest that the majority of studies are concerned with proposing some kind of *Model*, *Tool*, *Process*, *Approach*, *Method*, and *Template*. These different types of contributions may be an indication that not all artifacts types are equally suited for all activities in software and RE. Moreover, several persons with various roles and different requests use artifacts based on their individual work throughout the project [Liskin 2015]. This high number of approaches in these categories may also suggest that how requirements engineers should perform the safety activities is still an open research question.

4.5. RQ2.2: For which domains were these approaches proposed?

Figure 5 shows the distribution of the studies by application domain. 78.3% of the studies were classified as domain-independent, the remainder of the studies were developed in

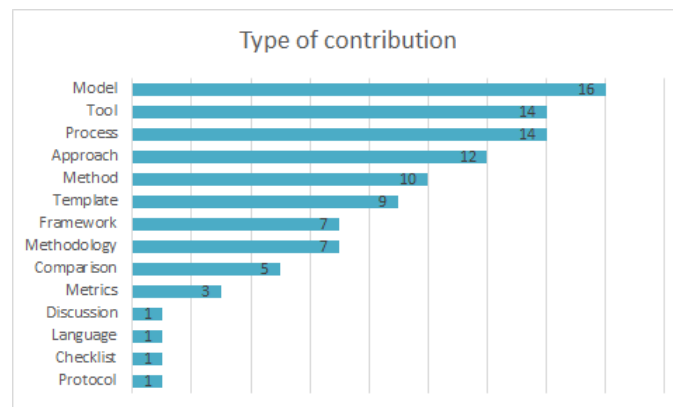


Figura 4. Type of contributions on requirements communication of SCS.

the following application domains: robotics, automotive, avionics, medical, railway, and mechatronics.

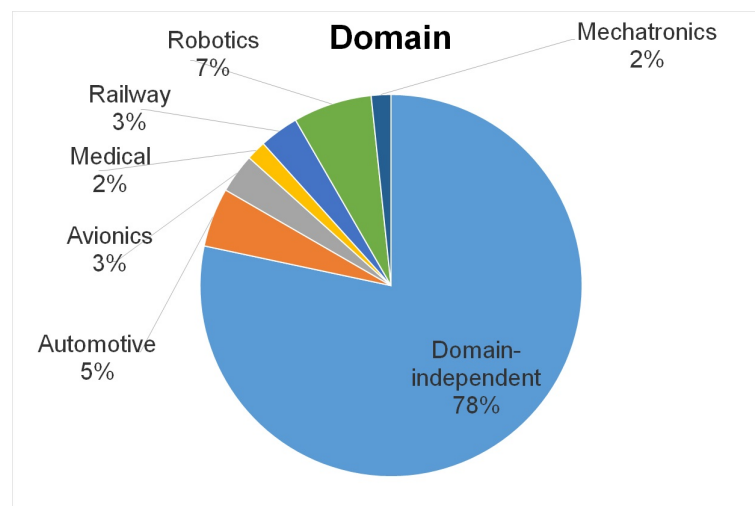


Figura 5. Application Domain.

In 47 studies (78.3%), the authors mentioned or gave some indication that the proposed approaches could be used in different safety-critical domains. We adopted the strategy used by Martins and Gorschek [Martins and Gorschek 2016] of classifying them as domain-independent. From these results, we can suggest that there is some tendency of improving requirements communication with a common body of knowledge in safety analysis. In a previous work [Vilela et al. 2018], we have proposed a domain independent metamodel called Safe-RE which is a Safety Requirements Metamodel Based on Industry Safety Standards.

Considering specific application domains, robotics was the domain where more approaches for requirements communication were proposed (7%). It was a surprise since we are expecting more contributions in traditional fields such as avionics or automotive. This outcome suggests that safety is a concern in different areas and requires the collaboration of multiple professionals [Leveson 2011][Martins and Gorschek 2016][Sikora et al. 2012][Hatcliff et al. 2014].

Analyzing the publication year of the 13 domain-specific approaches, we noticed that 69.23% are recent contributions (published in 2010 or after). This may suggest that the model-driven architecture and the domain specific languages as well as the frameworks for model-driven development might be influencing the approaches for considering domain-specific concerns.

Many standards require that a safety analysis is performed when developing or modifying a SCS. Hence, another factor influencing these results may be for what safety standards have the approaches been proposed (see RQ2.8) since many standards are domain-specific.

4.6. RQ2.3: What RE activities were supported by these approaches?

We categorized these activities according to the main steps of a RE process: elicitation, analysis and negotiation, specification, validation and management² (see Table 2). In summary, the results suggest that all RE activities are covered by the studies.

Tabela 2. RE activities supported by the approaches in requirements communication.

Activity	Count	%
Elicitation	13	21.67%
Analysis and Negotiation	31	51.67%
Specification	38	63.33%
Validation	28	46.67%
Management	12	20%

In summary, the results suggest that the studies cover all RE activities. The *Specification* activity is addressed by more than 60% of the studies. In fact, to some extent, this result was expected, since SCS are submitted to certification processes, and many of them must have to be compliant with some safety standard.

Analysis and Negotiation activities also encompass a high number of studies (51.67%). This result is also interesting because we argue that RE and safety engineers should collaborate in this activity to conduct better safety analysis and better safety requirements specifications. Furthermore, 28 studies (46.67% of papers included) covered both Specification and Analysis and Negotiation activities in the same paper, indicating that almost all studies, except [Chandrasekaran et al. 2009, Stålhane and Sindre 2007, Stålhane et al. 2010], that covered the Analysis and Negotiation are also concerned with the documentation of such analysis.

A significant number of studies addressed the Validation activity. This supports the need for certification that requires the construction of safety cases. Some studies (20%) covered the Management activity, indicating that it is important to manage the requirements and apply traceability mechanisms to improve the requirements communication.

There are also studies that aim to provide guidance for Elicitation activities. For instance, there are approaches that aim to represent safety knowledge to drive elicitation tasks. It is worth noting that the majority of the papers which addressed Elicitation activity also included Analysis and Negotiation and Specification activities.

²In this paper, we considered requirements management as a requirements activity.

Tabela 3. Requirements specification languages per domain.

RE language	Count
Natural Language	27
Use Cases Description	20
UML	16
Block diagram design language, and State machine design language, SysML	8 (each)
Context Diagram	6
Logics, and Formal methods	5 (each)
Mathematical notations	3
Problem Frames, and Event Time Diagram (ETD)	2(each)
KAOS, RSMML language, SpecTRM- RL modeling language, System Diagrams, HIVE requirements language, Goal Model of ATRIUM, Alloy, VDM++, Structured Analysis and Design Technique (SADT), Event-B, EAST-ADL, Bayesian Belief Networks (BBNs), ALTARICA, AUTOSAR, User Requirements Notation (URN), Requirements Definition and Analysis Language (RDAL), and Architecture Analysis and Design Language (AADL)	1 (each)

A research direction to enable continuous communication could be by making sure cross-competence teams work jointly on the most dominant group of activities such as elicitation and specification or specification and validation. However, assuring these competencies is a challenge to managers since other variables can influence the collaborative work.

Finally, among all 60 studies, only five papers address all activities of the RE process and only one study [Heimdahl 2007] proposed an entire RE process. This may be an indication that a holistic approach to improving the requirements communication that supports all activities of the RE process is needed. This may be one of the reasons for so many problems and challenges faced (see RQ1) in the development of SCS.

4.7. RQ2.4: Which requirements specification languages are used by these approaches?

The languages used by the studies to specify the requirements are listed in Table 3. Furthermore, five papers did not cite any language. We identified a great variety of requirements specification languages adopted by the approaches. *Textual requirements*, written in natural language, are the most frequent type of RE specification language addressed by the studies. This category included studies on all analyzed domains, but the majority of studies are domain-independent as can be seen in Table 3. It also encompasses studies that specify requirements documents using a requirements document template. In the supplement material (Section 3.2), we discuss the pros and cons of the identified languages. In the supplement material (Section 3.2), we discuss the pros and cons of the identified languages.

Although *Textual requirements* category is the dominant documentation style for requirements, it does not imply that natural language is considered a satisfactory specification technique [Sikora et al. 2012]. The qualitative study conducted by Sikora et al. [Sikora et al. 2012] showed that many embedded systems practitioners are dissatisfied with the use of natural language for requirements specification. In their study, practitioners revealed that they consider tedious and error-prone to deal with large bodies of natural language requirements.

For example, Sikora et al. [Sikora et al. 2012] frequently noted that checking the consistency of the natural language requirements specification must be done manually by means of inspections, which leads to an enormous effort. Furthermore, the requirements

specification must precisely define the physical process to be controlled as well as the requirements for the controller since embedded systems perform control tasks. Therefore, according to the practitioners, there is a need for approaches that support the specification of controller requirements because the use of natural language allows too many interpretations, whereas controller specifications are written as pseudo-code preempt the controller design [Sikora et al. 2012].

The natural language is wonderfully expressive, but frequently ambiguous [Whitehead 2007]. Furthermore, natural language is, generally, unsuitable to the capture of the rigorous arguments needed for a safety case. Recent work on Goal Structuring Notation (GSN) [Heimdahl 2007] is a direction that promises to assist in the construction of rigorous arguments. However, it should be not the only specification language used in the development of large systems such as the SCS.

The *Use Case Description* category is also popular in the selected studies. It encompasses 17 papers and is adopted mainly in domain-independent approaches. The *UML* category includes studies that adopted any diagram defined by this language, for instance, use case diagrams, activity diagrams, sequence diagrams, and class diagrams (statecharts were included in the state machine design language).

UML is a language used by many academic and industrial works. *UML* is also mainly adopted in domain-independent approaches. However, it imposes several constraints for modeling systems that are not limited only to software [Scholz and Thramboulidis 2013]. Being a software-oriented language, representing other aspects of systems (e.g., hardware, mechanics, and electronics), is more complicated than in a system-oriented language such as SysML. The development of SCS requires the modeling of the entire system, not just their software-related aspects, as requested by many international safety standards [Biggs et al. 2014].

Block diagram design language category is another language frequently used since many studies target at increasing the reliability of system components or the dependability rather than the safety of a system. However, although reliability is important for achieving safety, making a system more reliable is not sufficient if it has unsafe functions [Medikonda and Panchumarthy 2009]; and threats to safety are not limited to failing components. According to Biggs et al. [Biggs et al. 2014], when targeting dependability, it is important to model properties such as performance, possible faults, and maintenance, but they do not provide for important safety information such as hazards (other than faults) and potential harm.

State machine design language category is adopted by eight studies and is only adopted in domain-independent approaches. This language can model the behavior of a large number of problems through states, the possible input events, and the possible actions or output events that result from state transition.

The *Systems Modeling Language (SysML)* is used in seven studies in domain-independent approaches and one study in the Mechatronics domain. This outcome is expected since this language is a general-purpose visual modeling language for systems engineering applications. SysML is defined as a dialect of the UML standard and supports the specification, analysis, design, verification, and validation of a broad range of systems and systems-of-systems. This system may include hardware, software, information, pro-

cesses, personnel, and facilities.

In the study of Sikora et al. [Sikora et al. 2012], UML/SysML models are the most commonly used types of models in RE for embedded systems. They are used to describe structural aspects of the system and its environments, such as a power plant, a pumping station, a vehicle, or an airplane. Although GSN has been used for the construction of safety cases in some segments, the authors [Sikora et al. 2012] describe that goal models are rarely used.

An interesting question pointed out by Sikora et al. [Sikora et al. 2012] is that even when a model type has a potential benefit for the development project, it tends to be used less frequently, if its use is not mandatory, e.g., by project constraints or intra-organizational regulations.

Considering specific-domain approaches, we noticed that in the automotive and mechatronics domains, there is a variety of RE languages adopted. This may suggest that there is no consensus, and the researchers are trying to find which language is the most adequate for this domain.

These results seem to indicate the prevalence in the selected studies of informal and semi-formal approaches over formal ones in order to document and communicate requirements, design decisions, and relevant information among the project teams and actors in the development and certification of SCS as also concluded by [Martins and Gorschek 2016].

Many specification languages were cited by only one selected study. This suggests that many of them are not acceptable to all stakeholders involved in the RE process and, as a consequence, such languages not get used. According to Heimdahl [Heimdahl 2007], without a modeling language acceptable to all stakeholders, the language will not get used, and all research into formal techniques will not make them into software engineering practice. This result shows that a good specification language for the SCS domain should have a well-defined graphic notation to avoid misunderstandings as well as formal reasoning, preferably embedded in the language, to improve the consistency checks and the quality of the requirements specifications.

4.8. RQ2.5: Which tools are used for the requirements specification?

We believe that requirements communication is also improved by the use of shared tools, hence this question maps the ones used to develop the requirements specification of SCS. Table 4 lists the tools mentioned more than once in the selected studies.

Tabela 4. Tools used in the requirements specification.

Tool	Count	%
It does not cite	36	60%
A proposed one	10	16.67%
Sparx Systems Enterprise Architect	6	10%
ARTi-SAN Studio, DOORS, SystemWeaver, mCRL2, Rodin platform	2 (each)	3.33%
IBM Rational Software Architect, IBM Rational Rhapsody, IBM Rational Harmony for Embedded Real-Time Development tool, HIVE (Hierarchical Verification Environment) tool, Siemens Teamcenter Systems Engineering and Requirements Management, Elektra, Spreadsheet tool, Visio, SafeSlice, EATOP, Artop, Supremica, TCT, NBC, UPPAAL, UML4PF, Papyrus UML, ERRSYS, SRSV, OSATE, and jUCMNav	1 (each)	1.67%

The majority of the studies (36 studies - 60%) did not mention

any tool for requirements specification. On the other hand, some approaches used or recommended different tools such as Markovski and Mortel-Fronczak [Markovski and van de Mortel-Fronczak 2012] as well as Pernstål et al. [Pernstål et al. 2015]. This lack of tools is a substantial issue since they can contribute to the requirements communication and should consider safety concerns to improve shared understanding.

Other works (ten studies - 16.67%) report that they developed a tool to support their proposals but they did not present their names. The results might indicate that the tools are not adapted for SCS or to enable communication in large teams. Most tools are expensive per license and this forces companies to buy few licenses, limiting access to the central repository and thus hindering communication. Perhaps, the use of no tool (or using internal ones like excel) is a reaction towards the expensive licenses.

Table 4 present the tools cited in the included studies. However, many of them did not explicitly discuss how the tools support the communication throughout the RE activities (*elicitation, analysis, specification, validation, management*). As we described in Section 4.10, some studies adopt analysis tools as a form to improve the requirements communication. This category includes tools shared by stakeholders involved in the requirements specification and safety analysis as well as tools that support some kind of safety analysis.

4.9. RQ2.6: For which stakeholder were they proposed?

The stakeholders mentioned in the selected studies are listed in Table 5. The majority of the approaches were designed to be used by safety engineers and developers.

Tabela 5. Stakeholders involved in the approaches.

Stakeholder	Count	%
Safety Engineer	29	48.33%
Developer	23	38.33%
Software Engineer	19	31.67%
Requirements Engineer	18	30%
Design Engineer	10	16.67%
Architect	9	15%
Customer	7	11.67%
System Engineer	6	10%
Certification authorities and Project manager	3 (each)	5%
Human factors expert, Manufacturing (MAN), and Product development (PD)	2 (each)	3.33%
Supplier, Test engineer, Quality Manager, Cognitive engineer, Operator, Constraints Engineer, Domain Engineer, and Reliability Engineer	1 (each)	1.67%

The results presented in Table 5 suggest that, as expected, *safety engineers* are the stakeholders for which most studies have been proposed. The next most cited stakeholders in the selected studies were *Developer*, *Software Engineer*, and *Requirements Engineer*. This outcome might indicate that there is some confusion in the selected studies, perhaps not in the industry, of their roles and the division of attributions is not clearly defined.

These results show that the proposed approaches embrace different types of stakeholders involved in the development of SCS. Such approaches could help them in different but complementary activities, such as safety analysis and safety requirements specification as also concluded by Martins and Gorschek [Martins and Gorschek 2016]. Furthermore, the main responsibility of safety analysis is from safety engineers. However, it is advisable when it is shared with requirements engineers and stakeholders.

Moreover, there is a tendency of sharing the responsibility of safety analysis conduction by all these stakeholders mentioned above.

4.10. RQ2.7: What are the communication formats used?

We based our analysis on the work of Jim Whitehead [Whitehead 2007] that classifies the collaboration tools as *Model-based*, *artifacts-based*, *Process support*, *Awareness*, and *Collaboration infrastructure* in a roadmap about collaboration in software engineering. We complemented such classification with *Analysis tools*, and *Face-to-face verbal communication* categories according to the formats presented in the selected studies. Table 6 lists the communication formats used in the approaches.

Tabela 6. Communication format used in the approaches.

Communication Format	Count	%
Model-based collaboration	42	70%
Process support	26	43.33%
Artifacts-based	21	35%
Analysis tools	19	31.67%
Face-to-face verbal communication	4	6.67%
Collaboration infrastructure	3	5%
Awareness	2	3.33%

The *model-based collaboration* was used by 70% of the selected studies (42 studies). Hence, there is a tendency, in the selected studies, of using models to improve the requirements communication in SCS. Model-based specifications are consistent and less ambiguous than informal specification documents, forcing the stakeholders to make clear all aspects of the system early in the design process. Therefore, models provide a shared meaning that engineers use when coordinating their work, as when stakeholders consult a requirements specification to determine how to design a portion of the system or to perform the safety analysis.

The *Process support* was the second communication format most adopted by the approaches (26 studies - 43.33%). It consists of collaborating through on a predefined structure for the sequence of steps to be performed, the roles stakeholders must fulfill, and the artifacts that must be created. This communication format serves to reduce the amount of coordination required to initiate a project and to define the typical sequence of steps that should be followed in the development and the roles and artifacts that should be produced.

As we already discussed in Section 4.4, the development of a safety-critical system involves the creation of multiple *artifacts*. We considered as an artifact all documents that were not based on models. Each type has its own semantics and creating it is an inherently collaborative activity. Several stakeholders contribute to each of these artifacts, working to understand what each other has done, eliminate errors, and add their contributions [Whitehead 2007]. This communication format was used by 21 studies (35%).

Analysis tools were adopted as a communication format by 19 studies (31.67%). This category comprehends tools shared by stakeholders involved in the requirements specification and safety analysis as well as tools that support some kind of safety analysis.

The *Face-to-face verbal communication*, used by [Paige et al. 2008][Pernstål et al. 2015][Fricker et al. 2010][Fricker et al. 2008] (four

studies - 6.67%), includes meetings, informal conversations in hallways, doorways, and offices. However, the conversations are not formally structured, despite being the concern the development of a formal system, a piece of software, [Whitehead 2007].

The *Collaboration infrastructure* category comprises the technologies developed for the integration of software tools. Main forms of tool integration include data integration, ensuring that tools can exchange data, and control integration, ensuring that tools are aware of the activities of other tools, and can take action based on that knowledge [Whitehead 2007]. Only the studies [Beckers et al. 2013][Briones et al. 2007][Heimdahl 2007] (three studies - 5%) explicitly discussed this communication format.

The *Awareness* communication format was only explicitly used in [Medikonda and Panchumathy 2009][Schedl and Winkelbauer 2008] (two studies - 3.33%). It consists of providing information about the current activities of other stakeholders. By increasing this awareness of the activities, the stakeholders are able to perform coordination activities sooner, and potentially avoid conflicts [Whitehead 2007]. Furthermore, it is important to highlight that the other communication formats also contribute to improving the awareness but it was not the specific objective of the studies.

4.11. RQ2.8: For what safety standards have the approaches been proposed?

The safety standards presented in the approaches are exhibited in Table 7. Table 7 shows that the great majority of the approaches of requirements communication (46 studies - 76.67%) *does do not follow* any safety standard.

Tabela 7. Safety standards adopted.

Safety Standard	Year	Domain	Count	%
No			46	76.67%
IEC 61508	2010	Generic	4	6.67%
ISO 26262	2011	Automotive	3	5%
DO-178B	1992	Avionics	1	1.67%
ISO/IEC 15504	2003	Generic	1	1.67%
ISO 12207	1995	Generic	1	1.67%
ISO 12100	2010	Machinery	1	1.67%
IEC/SC65A	1992	Generic	1	1.67%
Australian Defence Standard Def (Aust) 5679	1998	Generic	1	1.67%
ANSI/RIA R15.06-1999	1999	Robotics	1	1.67%
ISO/IEC 9126	2001	Generic	1	1.67%
IEC 1508	1995	Generic	1	1.67%
IEC 61499	2011	Generic	1	1.67%
IEC 61131	1993	Generic	1	1.67%
EIA-632	1994	Generic	1	1.67%

This finding is worrying since SCS should be certified by regulatory bodies and this requires submitting relevant system safety information to appropriate authorities [Zoughbi et al. 2011]. Hence, the communication with certification authorities with regards to system certification requires providing proof that appropriate standards were followed during the development process [Zoughbi et al. 2011].

From the approaches which based their concepts in part on the definitions given by the international standards for safety, we identified fourteen standards (see Table 7). The date of the safety standard release varies between 1992 and 2011. 64.29% of the followed safety standards are developed for general purposes such as defining the safety life cycle,

the requirements for evaluation of the software development process, the terminology and guidelines.

5. Conclusions

The RE activities are critical to avoid the introduction of defects and misunderstandings among engineers and developers when developing SCS. Communication among work-groups that develop interdependent pieces of a system is crucial for a successful outcome of software development projects [Pernstal 2015]. This is an important question in the development of SCS considering that many safety problems occur due to errors and misunderstandings in safety requirements specifications.

Our mapping study draws on 60 studies, selected out of 1164, through a multi-stage process. An important feature of the review is that it does not restrict itself to a particular domain or safety standard. This broad scope in the search gives us deeper insights into the state-of-the-art about how the requirements communication is conducted in the RE process. Currently, we are working on the analysis of safety standards and the comparison of the results of the state-of-art with the state-of-practice is in progress.

5.1. Threats to validity

We adopted the classification of threats to validity well adopted in the literature which corresponds to Internal, External, Construct and Conclusion categories. *Construct validity*: For all concepts, we used many synonyms to ensure high coverage of potentially-relevant studies from a database search. *Internal validity*: In order to minimize selection and extraction mistakes, the selection process was performed in an iterative way. It is also worth noting that the all authors are lecturers and experienced researchers with expertise in RE, Software Engineering or SCS. *External validity*: In order to mitigate external threats, the search process was defined after several trial searches and validated with the consensus of the authors. *Conclusion validity*: The research protocol was carefully designed and discussed by the authors to minimize the risk of exclusion of relevant studies. It is worth highlighting that we did not restrict the time period of published studies to obtain the maximum coverage possible.

The results of this mapping study showed that although there are some approaches to improve the requirements communication of SCS, several problems still remain since many studies do not support the real needs of the industry. Therefore, this mapping study has generated several promising research directions:

- (1) How safety analysis techniques can be improved to evaluate shared understanding (RQ2.1)?
- (2) To what extent do the domain-independent approaches cover the needs of domain-specific critical systems (RQ2.2)?
- (3) Why the approaches do not cover the entire RE process? (RQ2.3)
- (4) To what extent do the tools used in the requirements specification are capable of improving requirements communication (RQ2.5)?
- (5) Which is the most effective communication format in requirements communication of safety-critical systems (RQ2.7)?
- (6) Why do the approaches not follow the guidelines of safety standards (RQ2.8)?

6. ACKNOWLEDGMENTS

We would like to acknowledge that this work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) from Brazil, and by KKS foundation through the S.E.R.T. Research Profile project at the Software Engineering Research Lab, Blekinge Institute of Technology, Sweden.

Referências

- Bjarnason, E., Wnuk, K., and Regnell, B. (2011). Requirements are slipping through the gaps? a case study on causes & effects of communication gaps in large-scale software development. In *International Requirements Engineering Conference (RE)*, pages 37–46. IEEE.
- Brady, A., Seigel, M., Vosecky, T., and Wallace, C. (2007). Addressing communication issues in software development: A case study approach. In *Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on*, pages 301–308. IEEE.
- Glinz, M. and Fricker, S. A. (2015). On shared understanding in software engineering: an essay. *Computer Science-Research and Development*, 30(3-4):363–376.
- Hatcliff, J., Wassying, A., Kelly, T., Comar, C., and Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. In *Proceedings of the on Future of Software Engineering*, pages 182–200. ACM.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Leveson, N. (2011). *Engineering a safer world: Systems thinking applied to safety*. Mit Press.
- Liskin, O. (2015). How artifacts support and impede requirements communication. In *Requirements Engineering: Foundation for Software Quality*, pages 132–147. Springer.
- Martins, L. E. G. and Gorschek, T. (2016). Requirements engineering for safety-critical systems: A systematic literature review. *Information and Software Technology*, 75:71–89.
- Nakamura, H., Umeki, H., and Kato, T. (2016). Importance of communication and knowledge of disasters in community-based disaster-prevention meetings. *Safety Science*.
- Pernstal, J. (2015). *Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development*. PhD thesis, Department of Computer Science and Engineering, CHALMERS UNIVERSITY OF TECHNOLOGY.
- Rasmussen, J. and Lundell, Å. K. (2012). Understanding ”communication gaps” among personnel in high-risk workplaces from a dialogical perspective. *Safety science*, 50(1):39–47.
- Sikora, E., Tenbergen, B., and Pohl, K. (2012). Industry needs and research directions in requirements engineering for embedded systems. *Requirements Engineering*, 17(1):57–78.

- Vilela, J., Castro, J., Martins, L. E. G., and Gorschek, T. (2017). Integration between requirements engineering and safety analysis: A systematic literature review. *Journal of Systems and Software*, 125:68–92.
- Vilela, J., Castro, J., Martins, L. E. G., and Gorschek, T. (2018). Safe-re: a safety requirements metamodel based on industry safety standards. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, pages 196–201. ACM.
- Wang, Y., Graziotin, D., Kriso, S., and Wagner, S. (2018). Communication channels in safety analysis: An industrial exploratory case study. *arXiv preprint arXiv:1804.08909*.
- Whitehead, J. (2007). Collaboration in software engineering: A roadmap. *FOSE*, 7(2007):214–225.

Systematic Literature Review References

- Beckers, K., Heisel, M., Frese, T., and Hatebur, D. (2013). A structured and model-based hazard analysis and risk assessment method for automotive systems. In *Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on*, pages 238–247. IEEE.
- Biggs, G., Sakamoto, T., and Kotoku, T. (2014). A profile and tool for modelling safety information with design information in sysml. *Software & Systems Modeling*, pages 1–32.
- Briones, J. F., De Miguel, M. Á., Silva, J. P., and Alonso, A. (2007). Application of safety analyses in model driven development. In *Software Technologies for Embedded and Ubiquitous Systems*, pages 93–104. Springer.
- Chandrasekaran, S., Madhumathy, T., Aparna, M., and Shilpa Jain, R. (2009). A safety enhancement model of software system for railways. In *Systems Safety 2009. Incorporating the SaRS Annual Conference, 4th IET International Conference on*, pages 1–6.
- Fricker, S., Gorschek, T., Byman, C., and Schmidle, A. (2010). Handshaking with implementation proposals: Negotiating requirements understanding. *IEEE software*, (2):72–80.
- Fricker, S., Gorschek, T., and Glinz, M. (2008). Goal-oriented requirements communication in new product development. In *Software Product Management, 2008. IWSPM '08. Second International Workshop on*, pages 27–34.
- Hatcliff, J., Wassyn, A., Kelly, T., Comar, C., and Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. In *Proceedings of the on Future of Software Engineering*, pages 182–200. ACM.
- Heimdahl, M. P. E. (2007). Safety and software intensive systems: Challenges old and new. In *Future of Software Engineering*, pages 137–152. IEEE Computer Society.
- Markovski, J. and van de Mortel-Fronczak, J. (2012). Modeling for safety in a synthesis-centric systems engineering framework. In *Computer Safety, Reliability, and Security*, pages 36–49. Springer.
- Medikonda, B. S. and Panchumorthy, S. R. (2009). A framework for software safety in safety-critical systems. *SIGSOFT Softw. Eng. Notes*, 34(2):1–9.

- Paige, R. F., Charalambous, R., Ge, X., and Brooke, P. J. (2008). Towards agile engineering of high-integrity systems. In *Computer Safety, Reliability, and Security*, pages 30–43. Springer.
- Pernstål, J., Gorschek, T., Feldt, R., and Florén, D. (2015). Requirements communication and balancing in large-scale software-intensive product development. *Information and Software Technology*, 67:44–64.
- Schedl, G. and Winkelbauer, W. (2008). Practical ways of improving product safety in industry. In *Improvements In system Safety*, pages 177–193. Springer.
- Scholz, S. and Thramboulidis, K. (2013). Integration of model-based engineering with system safety analysis. *International Journal of Industrial and Systems Engineering*, 15(2):193–215.
- Sikora, E., Tenbergen, B., and Pohl, K. (2012). Industry needs and research directions in requirements engineering for embedded systems. *Requirements Engineering*, 17(1):57–78.
- Stålhane, T. and Sindre, G. (2007). A comparison of two approaches to safety analysis based on use cases. In *Conceptual Modeling-ER 2007*, pages 423–437. Springer.
- Stålhane, T., Sindre, G., and Du Bousquet, L. (2010). Comparing safety analysis based on sequence diagrams and textual use cases. In *Advanced Information Systems Engineering*, pages 165–179. Springer.
- Zoughbi, G., Briand, L., and Labiche, Y. (2011). Modeling safety and airworthiness (rtca do-178b) information: conceptual model and uml profile. *Software & Systems Modeling*, 10(3):337–367.

Desafios e Práticas da Engenharia de Requisitos no Contexto de Fábrica de Software com foco na Documentação e Gestão do Conhecimento.

Luana Souza¹, Erica Miranda¹, Márcia Lucena¹, Apuena Gomes²

¹Departamento de Informática e Matemática Aplicada – Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

²Instituto Metr pole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

luana.tms@ppgsc.ufrn.br, eecmiranda@gmail.com,
marcia.lucena@dimap.ufrn.br, apuena@imd.ufrn.br

Abstract. *The difficulties in drafting requirements documentation and knowledge management among the requirements engineers in the software factories were the motivation for an adapted organizational ethnography. The overall objective was to investigate challenges in two teams of requirements engineers in a factory, which impact the production and maintenance of a documentation. The identified challenges were grouped into three categories and after triangulation, practices were indicated, based on the literature, to overcome the challenges encountered. The identified challenges, the benefit in the use of the suggested practices and the ethnographic process itself are the contributions of this work.*

Resumo. *As dificuldades na elaboração da documentação de requisitos e a gestão do conhecimento entre os engenheiros de requisitos nas fábricas de software foram a motivação para realização de uma etnografia organizacional adaptada. O objetivo geral foi investigar desafios em duas equipes de engenheiros de requisitos de uma fábrica, que impactem na produção e na manutenção de uma documentação. Os desafios identificados foram agrupados em três categorias e após a triangulação, práticas foram indicadas, baseadas na literatura, para suplantam os desafios encontrados. Os desafios identificados, o benefício no uso das práticas sugeridas e o próprio processo etnográfico são as contribuições deste trabalho.*

1. Introdução

A documentação na Engenharia de Requisitos (ER) estabelece uma comunicação entre todos os envolvidos e nivela o conhecimento destes sobre o projeto. Nesse sentido, a documentação revela-se como um ativo específico para transferir e compartilhar conhecimento, trazendo consigo todos os desafios atrelados à esta tarefa. Em sabendo que os requisitos costumam ser uma das fontes de falhas em projetos de software, por diferentes motivos dentre eles, uma documentação inadequada ou insuficiente. A documentação precisa ser consistente, não-ambígua, ter uma estrutura clara, permitir ser modificada e estendida, além de completa e rastreável, como defendeu Pohl e Rupp (2015). Deste modo, a documentação seria mais um veículo de comunicação, considerando, que segundo Chau e Maurer (2004), a gestão do conhecimento (transferir e compartilhar conhecimento) é uma

tarefa difícil e, no passado, já foram utilizados processos e representações mais estruturadas e formalizadas.

Para entender melhor o conceito de gestão do conhecimento, é necessário entender a diferença entre dado, informação e conhecimento. Segundo Rossini e Palmisano (2003), dado é o elemento que representa a menor parte da informação, a informação é o dado configurado de forma adequada ao entendimento e à utilização, enquanto, o conhecimento é a capacidade adquirida para interpretar e atuar sobre um conjunto de Informações, adquirida a partir das relações que o indivíduo estabelece sobre o conjunto combinado com outros conjuntos que já lhe é conhecido (ex: experiência, visões, valores, crenças), possibilitando a compreensão e a conclusões sobre e a partir dele. Assim, a gestão do conhecimento, para Terra (2000), é uma análise de políticas, processos e ferramentas com o propósito de compreender o processo de geração, identificação, armazenamento, disseminação, compartilhamento e aplicação do conhecimento organizacional para produzir ganhos econômicos para a empresa.

Apesar do mérito dessa abordagem, com a tendência dos processos ágeis de software, processos e representações foram substituídos e a transferência de conhecimento face-a-face começaram a ser utilizadas. Sendo que essa prática (face-a-face) depende muito de interações sociais e de uma boa rede de relacionamento dentro da equipe. Isto fez com que as limitações da prática ágil comessem a serem vistas: i) a aplicação/software depende muito da comunicação face-a-face, tornando-se de difícil aplicação para grandes equipes, ii) a localização de integrantes (em caso de ausências) pode ser difícil e a confiança em um compartilhamento informal de informações pode apresentar desafios, iii) a prática só facilita o aprendizado dentro da equipe, mas não de outras equipes. Nesse sentido, a proposta foi entender como os dois times de requisitos estavam trabalhando para compartilhar o conhecimento em equipes que utilizam as práticas Scrum.

O Scrum, segundo Bellenzier *et al.* (2015), é um método que se baseia em boas práticas de gerenciamento de projetos, utilizando ciclos iterativos, curtos e incrementais, com envolvimento constante do time e visibilidade para o cliente. Para Sutherland e Schwaber (2013), as equipes Scrum têm como características a auto-organização, a multifuncionalidade, não reconhecendo títulos para equipe de desenvolvimento ou mesmo subequipes independente dos domínios a serem tratados. Os membros da equipe podem ter habilidades em alguma área específica, mas a responsabilidade é da equipe de desenvolvimento como um todo. Quanto à documentação no Scrum, esta difere-se das metodologias tradicionais, pois evolui a cada iteração, mantendo-se flexível e visível.

Neste contexto, foi planejada uma etnografia organizacional para entender como uma fábrica de software lida com essa questão, levantando desafios e propondo soluções a partir de práticas utilizadas no Scrum e na Gestão do Conhecimento. As questões de pesquisa deste estudo foram: 1. Quais são os desafios enfrentados por engenheiros de requisitos na elaboração da documentação em equipe ágil? 2. Como esses desafios impactam no trabalho dos engenheiros de requisitos em equipe ágil?

Este trabalho está organizado como segue. A Seção 2 trata da metodologia de pesquisa utilizada neste estudo, enquanto que a Seção 3 apresenta a execução do planejamento elaborado para a etnografia organizacional. Na Seção 4, são mostrados os resultados obtidos com este estudo e proposições de soluções. Para a Seção 5, são expostas as discussões considerando os resultados, as limitações e ameaças à validade. Por fim, os

trabalhos relacionados e as considerações finais são apresentados nas Seções 6 e 7, respectivamente.

2. Metodologia de Pesquisa

Segundo Angrosino (2009), a pesquisa qualitativa tem o objetivo de entender, descrever e explicar fenômenos sociais de diferentes formas, tais como: i) Analisando experiências de indivíduos ou grupos; ii) Examinando interações e comunicações, que estejam se desenvolvendo; iii) Investigando documentos (textos, imagens, filmes ou música) ou traços semelhantes de experiência ou interações.

Como Neyland (2008) explica, a etnografia significa:

“Etnografia é uma metodologia de pesquisa desenvolvida originalmente no campo da antropologia, que é agora utilizada em uma variedade de trabalhos (em, por exemplo, antropologia, sociologia, teoria da administração, estudos organizacionais e estudos culturais). Envolve a observação e a participação em grupos específicos (como grupos indígenas locais, consultores de gestão, estudantes de medicina e assim por diante). Esta observação e participação visa envolver-se com questões de como um determinado grupo opera, o que significa ser um membro de um grupo específico e como as mudanças podem afetar esse grupo.” [Neyland 2008]

Para Ybema *et al.* (2009), existe uma distinção da etnografia organizacional para outras formas de etnografia. Uma dessas diferenças é o cenário, ou "campo", que é o foco da análise, e define a etnografia organizacional. Dessa forma, a etnografia organizacional em vez de tentar compreender toda a organização, estaria mais orientada a seguir uma pessoa ou uma prática organizacional específica. Assim, nesta pesquisa, foi escolhida a metodologia etnográfica que utiliza as seguintes etapas: (1) observações em campo; (2) entrevistas; (3) arqueologia dos sistemas, - para conhecimento dos sistemas, documentações, dos grupos e perfis, com objetivo de entender sua dinâmica e interações, utilizadas em duas equipes de engenheiros de requisitos; e (4) triangulação - que estrutura e combina os eventos observados.

Desta forma, foi desenvolvido um protocolo para a etnografia organizacional, onde foram definidas as fases do estudo, as melhores semanas para as interações com os participantes dentro das suas agendas, reunir os artefatos necessários, solicitar os acessos aos sistemas, aos documentos e qualquer instrumento utilizado pela instituição participante. Além disto, foi feita uma reunião com o diretor de tecnologia da informação (*Chief Information Officer* - *CIO*) da instituição para explicar, e pedir autorização para a realização do estudo com duas das equipes desta instituição.

2.1. Caracterização das Fases e Artefatos Utilizados

Para conhecer a dinâmica de trabalho das equipes com ênfase nas atividades documentais relacionadas aos sistemas de trabalho, artefatos, processos, comunicação, interação e cultura das equipes, para que fosse possível o reconhecimento de desafios e a proposição de melhorias. Para isto, o estudo foi dividido em etapas: i) entrevista com os líderes das equipes; ii) observação em campo; iii) entrevistas com os engenheiros requisitos; iv)

entrevista com o diretor de sistemas; v) arqueologia dos sistemas e acessos cedidos aos sistemas utilizados rotineiramente.

Na primeira etapa, as entrevistas com os líderes, foram feitas antes do período de observação das equipes, com o intuito de conhecer um pouco sobre os participantes das equipes, como era feito o gerenciamento das atividades, as experiências vividas pelos líderes, e as dificuldades da liderança. Para isto, foi elaborado um roteiro e solicitada a permissão de gravação.

Na segunda etapa, a observação em campo, era composta por: i) eventos programados e não programados; ii) variáveis ambientais, que pudessem influenciar o trabalho da equipe; iii) verificação de disponibilidade de informação e de material de trabalho suficiente; iv) verificação do uso diário das ferramentas e possíveis dificuldades; v) observação e identificação dos processos de comunicação entre a equipe e as demais equipes; vi) observação de momentos de atividade de alta concentração; vii) níveis de ruído e interrupções; e viii) observação e identificação de dúvidas recorrentes dos públicos alvo.

Já a terceira etapa, foram feitas entrevistas com os engenheiros de requisitos, a partir das observações feitas em campo e das dificuldades relatadas na literatura. Tal como nas entrevistas anteriores, foram adotados os mesmos procedimentos. Como era sabido que estavam em uma fase de transição de tecnologias, arquitetura e, também, das ferramentas utilizadas pelas equipes, a entrevista com o diretor de sistemas foi importante para entendimento das decisões e suas razões. Assim como, nas outras entrevistas foram adotadas as mesmas estratégias quanto às perguntas, permissão e gravação das respostas. Para em seguida, iniciar a arqueologia dos sistemas, definida para quinta etapa, foi realizada a partir da observação de documentos e sistemas utilizados pelos engenheiros de requisitos na segunda etapa.

2.2. Participantes

A instituição participante pode ser considerada de grande porte, segundo o Banco Nacional de Desenvolvimento - BNDES¹. Esta é responsável pela construção, manutenção e evolução de sete sistemas e de diferentes naturezas. Os sistemas servem a própria comunidade interna e mais de 50 outras entidades parceiras em todo o Brasil, em um processo chamado de transferência de tecnologia, no qual equipes de tecnologia da informação - TI (instituições externas) são os verdadeiros clientes da instituição observada e, após treinamento e terem uma canal de comunicação estabelecido e disponível por 24h/7d na semana, assumem, internamente, a responsabilidade de prestar serviços de implantação e manutenção, nas suas instituições. Essas são de diferentes segmentos, que ao assinar o termo ou o contrato, podem adotar os diferentes sistemas de acordo com sua necessidade, e também optar por quais módulos querem que sejam implantados dos sistemas escolhidos.

Existem várias formas de compartilhamento de informações e de comunicação entre a instituição, a comunidade e os clientes. Foram implantados dois sistemas para comunicação, esclarecimento de dúvidas e priorização das demandas por todo ecossistema envolvido, e criado um comitê interno para definição a partir da priorização do ecossistema, regras pré-estabelecidas e capacidade de atendimento da instituição.

¹ <https://www.bndes.gov.br/>

Os sistemas e equipes escolhidas para esta etnografia têm a responsabilidade de manter e atualizar dois dos maiores sistemas da fábrica de software. No momento da escolha das equipes, eram estas equipes que tinham maior necessidade de apoio nas atividades de documentação, foco principal deste trabalho. Neste estudo, foram envolvidas 11 pessoas, sendo que oito pessoas tinham cargo e desempenhavam, especificamente, a função de engenheiros de requisitos, duas pessoas, respondiam pelo papel de líderes de equipe, e uma pessoa responsável era pela diretoria de sistemas. Ressalta-se que, embora as equipes tivessem declarado que adotavam o Scrum, foram observados que os tamanhos das equipes e projetos eram maiores do que esperados, os cargos e, conseqüentemente, os papéis eram bem definidos, assim como as atividades e as responsabilidades. Por conta disto, a opção foi trabalhar apenas com os engenheiros de requisitos, sendo cinco da equipe 1 e três da equipe 2.

Com a intenção de manter a privacidade e o sigilo, os participantes tiveram seus perfis basicamente descritos, como a seguir: i) **Diretor de Sistemas**: idade 40 anos, possui pós-graduação (doutorado) na área de computação, oito anos de experiência em desenvolvimento de software, gestão e coordenação de atividades de arquitetura de sistemas; ii) **Líderes de Equipes**: idades entre 35 e 37, com três anos de experiência na função, e ambos possuem pós-graduação (áreas de computação e design); iii) **Engenheiros de Requisitos**: idades entre 24 à 38 anos, com formação na área de conhecimento da Ciência da Computação.

3. Realização da Etnografia Organizacional

Esta pesquisa etnográfica foi realizada utilizando três formas de coletas de dados. Sendo detalhadas nos subseções seguintes (3.1, 3.2 e 3.3).

3.1. Observação das Equipes

As equipes foram observadas durante o período de 80 horas, durante este período foram permitidas apenas anotações, e não a filmagem ou fotografias: atividades programadas e não programadas; disponibilidade de material de trabalho suficiente; uso diário das ferramentas e possíveis dificuldades; identificação dos processos de comunicação entre a equipe e as demais equipes; observação de momentos de atividade de alta concentração; níveis de ruído e interrupções; e identificação de dúvidas frequentes das equipes. Foram observados também que:

- 1) O ambiente físico era compartilhado por toda equipe de cada projeto observado; não havia agrupamentos físicos por funções ou atividades; o ambiente era refrigerado e bem iluminado; as mesas eram pequenas e individuais; e os integrantes ficavam lado a lado ou frente a frente;
- 2) Quanto ao histórico da documentação de requisitos disponível, os projetos avaliados possuíam aproximadamente 14 anos de existência; durante anos, estes produtos não foram documentados de forma adequada ao longo do tempo; posteriormente, a instituição, adotou a utilização de uma ferramenta Wiki² interna com o intuito de documentar o detalhamento necessário sobre os softwares; com certo tempo de uso da Wiki, sua efetividade e a rastreabilidade passou a ser questionada dentro da instituição, principalmente, pelas

² <https://pt.wikipedia.org/wiki/Wiki>

dificuldades de versionamento dos artefatos; recentemente, adotou-se então a plataforma Git³ para o adequado armazenamento da documentação e demais artefatos por versão disponibilizada; e, neste momento, os engenheiros passaram a trabalhar com uma documentação representada por: especificação de caso de uso, glossário e mensagens de sistema;

3) Os fluxos de atividades das equipes possuíam dois fluxos de atividades: a) “Sustentação”, no qual era feita a manutenção dos sistemas, reparando erros e realizando pequenos ajustes; e b) “Evolução”, onde eram desenvolvidas e implantadas as novas versões e novas funcionalidades; no fluxo de sustentação, as atividades eram recebidas através de um sistema de gerenciamento de atividades cadastradas pelo setor de suporte de tecnologia interno da instituição a partir de uma solicitação dos usuários; no fluxo de evolução, as atividades são definidas de acordo com os novos requisitos de cada sistema; e estas atividades são programadas e divididas por sprints;

4) Quanto às atividades coletivas observadas, em uma das equipes foi possível acompanhar uma reunião de sprint de um módulo, onde a equipe fez uma pequena retrospectiva da sprint anterior, onde foram discutidos: a) o que feito, b) o que ficou pendente, c) o que deu certo, d) o que deu errado, e) o que era possível melhorar; e também foi possível acompanhar reuniões com diversos usuários internos e reuniões dos engenheiros de requisitos com a diretoria de sistemas para fazer o acompanhamento da produção da nova documentação de requisitos.

3.2. Entrevistas

As entrevistas com os líderes de equipe foram feitas antes do período de observação, com o intuito de conhecer um pouco das equipes, do gerenciamento das atividades, da experiência e das dificuldades da liderança. As entrevistas com os engenheiros de requisitos foram feitas em um período posterior ao da observação em campo, no intuito de conhecê-los melhor, e tentar fazer com que eles pudessem ficar um pouco mais à vontade para contribuir com as questões levantadas. Já a entrevista com o diretor de sistemas ocorreu após as entrevistas dos engenheiros, com o intuito de entender suas decisões, motivações, experiências e dificuldades. Os roteiros das entrevistas foram elaborados para cada perfil (engenheiro de requisitos, líder de equipe e diretor de sistemas), embora existiam perguntas em comum, e todas questões eram abertas.

A seguir, são apresentadas apenas as respostas pormenorizadas das entrevistas com os engenheiros de requisitos, visto que as entrevistas com os líderes de equipe e a diretoria de sistemas eram necessárias para explicar a pesquisa elaborada, pedir autorizações, entender as razões e as decisões tomadas por pessoas nessas funções ou cargos. O foco e os maiores desafios estavam relacionados com as atividades, os perfis, as experiências e as relações entre os outros stakeholders e cada um dos engenheiros de requisitos observados, na primeira fase da pesquisa.

Ao todo foram entrevistados oito engenheiros de requisitos, os percentuais demonstrados podem dar o total de 100% se cada participante respondeu 1 item, ou pode dar mais 100% se cada participante respondeu mais de um item. As respostas das entrevistas foram abertas e não estimuladas, sendo categorizadas e demonstradas em números absolutos e percentuais para o entendimento do cenário geral, como a seguir:

³ <https://pt.wikipedia.org/wiki/Git>

- 1) Atualização Profissional: ao serem questionados sobre como buscavam a atualização profissional, todos os engenheiros de requisitos lembraram de citar o curso sobre engenharia de requisitos que está sendo oferecido pela instituição. Contudo, apenas (5) 62,5% destes citaram buscar atualização por conta própria.
- 2) Técnicas de Elicitação: a principal técnica de elicitação - citada por (3) 37,5% dos engenheiros - foi a técnica de entrevista. O resultado geral sobre as técnicas de elicitação sugeriram que os entrevistados conhecem até oito opções distintas (Entrevistas, Protótipos, Brainstorms, Diagramas BPM, Arqueologia de Sistemas, Engenharia Reversa, Questionários e Apresentações). Entretanto, foi visto que o domínio de algumas destas técnicas estava concentrado em poucas pessoas. Técnicas de Documentação: foram citados: diagramas, atas de reunião e casos de testes. Os diagramas foram indicados por (4) 50% dos entrevistados como a principal técnica de documentação. Destes diagramas, o de negócio (BPM) foi citada por (3) 75%.
- 3) Técnica de validação: a “validação com o cliente” foi citada por (3) 37,5% e “validação sem o cliente” por (3) 62,5%. Notou-se que as pequenas alterações foram feitas sem cliente, enquanto as grandes, preferencialmente, feitas em sua presença.
- 4) Estratégias de gerenciamento de requisitos foram citadas: i) quadro de tarefas (1) 12,5%, ii) estudos de impacto (1) 12,5%; e iii) (6) 75% dos engenheiros não citou uso de estratégias para gerenciar requisitos.
- 5) Importância do engenheiro de requisitos foram citadas características, como: domínio do negócio, capacidade de criar soluções, ligação entre cliente e equipe de desenvolvimento e a própria importância dos requisitos. Um dado surpreendente foi o fato de (3) 37,5% que não souberam responder a esta questão.
- 6) Artefatos prioritários: a especificação de caso de uso representará (7) 87,5%, diagrama de estados (1) 12,5%, tarefas (1) 12,5%, mas para (1) 12,5% “não existia artefato prioritário”. A observação etnográfica revelou que a comunicação das equipes depende formalmente da criação e do conteúdo descrito nas tarefas e - informalmente - da presença e da disponibilidade dos engenheiros de requisitos para esclarecimento de dúvidas. Apesar disto, a tarefa não foi reconhecida como um artefato prioritário.
- 7) Estilo de escrita: este deveria ser simples, mais detalhado, com imagens, todavia (2) 25% não responderam. Foi informado que havia um documento de orientação, mas não foi apresentado por nenhum dos entrevistados.
- 8) Público alvo da documentação: para (7) 87,5% eram os responsáveis pela implantação (por exemplo, um analista de tecnologia da informação) e os interessados nas instituições parceiras; alguns ainda citam as equipes internas de: suporte (1) 12,5%, desenvolvimento (1) 25%, testes (1) 12,5%, mas ainda assim (1) 12,5% não soube responder.
- 9) Processo de gerenciamento de requisitos: nesta questão, foram citadas: o gerenciamento coletivo (4) 50%, o gerenciamento individual (2) 25%, contudo foi explicitado que para (2) 25% que não há gerenciamento. No gerenciamento coletivo, foram citadas as ferramentas: Wiki (2) 25% e GitHub (3) 37,5%. A Wiki era utilizada antes do novo modelo de especificação, e neste para as especificações de casos de uso era pretendido utilizar o GitHub para versionamento da documentação.
- 10) Políticas e modelos: foram citadas: a existência de política; a existência de modelo; a não existência de política; como também a não existência de modelo. Dentro da política foi citado o trabalho home office para a construção das especificações em um período de um dia, enquanto em modelo foi citada a existência do guia para elaboração da especificação. O guia para construção dos casos de uso foi confundido com política.

11) Base de conhecimento: as respostas remeteram a: “não existe” (2) 25%, “existe, mas não faz reuso” (3) 37,5%, “existe, reuso” (1) 12,5% e “não respondeu” (2) 25%. Estratégias de recuperação de informações: as indicadas foram: as baseadas em conhecimento tácito (6) 75% e baseada em conhecimento explícito (3) 37,5%. No caso das estratégias explícitas, as respostas foram: tarefas (2) 25%, e-mails (1) 12,5% e legislação (2) 25%. No caso das estratégias tácitas, as respostas foram: consultar a clientes (4) 50%, não consultar clientes (1) 12,5%, baseada em conhecimento tácito próprio (5) 62,5% e baseada em conhecimento tácito da equipe (5) 62,5%).

12) Evidências legais: quando necessárias, foram: e-mails (5) 62,5%, evidência inclusa no sistema (6) 75%, solicitações de suporte (1) 12,5%), registro na tarefa (3) 37,5%.

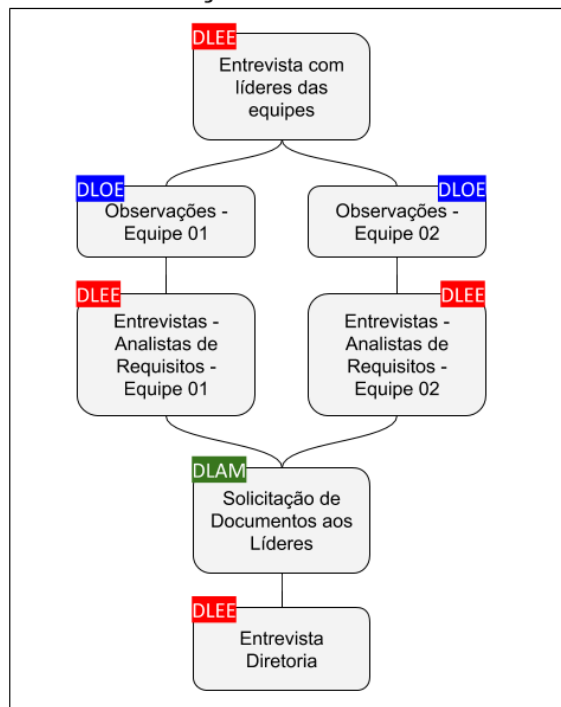
13) Oportunidades para melhorar o trabalho: havia muita expectativa ou desapontamento, quando questionados sobre o assunto, e suas respostas foram: incluir mais pessoas na equipe (2) 25%, trabalhar somente com análise e especificação dos requisitos (2) 25%, mais participação em eventos, cursos e reuniões na área de requisitos (1) 12,5%, trabalhar sem interrupções (2) 25%, possuir mais conhecimento na área de requisitos (5) 62,5%, fazer um melhor gerenciamento do tempo (3) 37,5%.

3.3. Arqueologia dos Sistemas

Na etapa de análise de materiais de arquivo, foram solicitados todos os documentos utilizados pelos engenheiros de requisitos durante a etapa de observação. As equipes não possuíam repositórios comuns para armazenar os arquivos, apesar de utilizarem ferramentas com este propósito, como Google Drive⁴ e Trello⁵. A organização não seguia nenhum padrão, embora tivessem sido comentados em diferentes momentos a existência de modelos e políticas de “boas práticas”. Também, foi visto que não existia um ambiente visual (físico ou virtual), onde as equipes pudessem ver as tarefas do grupo. No planejamento da Diretoria, havia a proposta de customização e a implantação de uma nova ferramenta integrada para gerenciamento de projetos.

⁴ <https://drive.google.com>

⁵ <https://trello.com>

Fase de execução**Legenda**

DLOE	Desafios levantados nas observações da etnografia
DLEE	Desafios levantados nas entrevistas da etnografia
DLAM	Desafios levantados na análise de materiais

Figura 1. Esquematização da fase de execução da etnografia organizacional para levantamento de desafios da engenharia de requisitos na fábrica de software

A Figura 1 ilustra como foi planejada a execução da etnografia organizacional na fábrica de software junto às duas equipes para levantamento dos desafios da engenharia de requisitos enfrentados por estes profissionais no seu cotidiano.

4. Resultados

Este estudo etnográfico demonstrou a partir da triangulação dos desafios encontrados nas observações, entrevistas e arqueologia dos sistemas, que os desafios encontrados impactavam diretamente na equipe de requisitos. Assim, optou-se por cada item (desafio) receber um identificador individual, incluindo a etapa no qual foi encontrado, desta forma: “Desafios Levantados nas Observações da Etnografia” passaram a ser identificados como “DLOE” (Tabela 1); “Desafios Levantados nas Entrevistas da Etnografia”, como “DLEE” (Tabela 2); e por fim os “Desafios Levantados nas Análises de Materiais”, consequentemente “DLAM” (Tabela 3). Cada desafio pode fazer parte de uma ou mais categorias. Estas categorias foram emergiram após a coleta de todos os dados, e relacionaram-se a partir de diferentes interações e iterações utilizando a metodologia Grounded Theory adaptada de Charmaz e Mitchell (2001) para uma possível solução, como foram demonstrados nas tabelas a seguir.

4.1. Desafios Encontrados

Os desafios demonstrados, na Tabela 1, foram encontrados na etapa de observação, quando as equipes de engenheiros de requisitos tiveram todas as suas atividades observadas e em seguidas registradas em um diário de pesquisa. Desse registro foi possível submergir os desafios e as categorias associadas.

Tabela 1. DLOE - Desafios Levantados nas Observações da Etnografia.

ID	Título	Categoria
01	Estabelecer prazos de entrega é uma dificuldade	Metodologia ágil
02	Presença de gargalos no processo produtivo	Metodologia ágil
03	Documentar problemas e decisões tomadas, motivações	Documentação
04	Aumentar a produtividade dos analistas de testes	Compartilhamento de conhecimento
05	Remanejar a equipe durante sprints, podendo comprometer a produtividade	Compartilhamento de conhecimento
06	Perduram dúvidas sobre quantidade de tarefas por sprints	Metodologia ágil
07	Documentação utilizada não atende as versões do sistema	Documentação
08	Dependência da disponibilidade dos engenheiros de requisitos para continuidade do processo produtivo	Documentação/Compartilhamento de conhecimento
09	Incerteza sobre efetividade da mudança da documentação	Documentação
10	Equipes muito grandes (média de 30 pessoas)	Metodologia ágil
11	Duração do tempo destinado a reuniões	Metodologia ágil
12	Permanecem dúvidas sobre como estimar o tempo das tarefas	Metodologia ágil

Antes do período de observações foram feitas entrevistas com os líderes das equipes, com o objetivo de conhecer o modo de trabalho das equipes e do líder. Após o período de observações, foram feitas entrevistas com os engenheiros de requisitos, a fim de confrontar achados encontrados nas observações e entender sua visão sobre o trabalho com os requisitos. Após todas as etapas também foi feita uma entrevista com a diretoria de sistemas, no intuito de conhecer seus direcionamentos e decisões. Todas as entrevistas foram gravadas e depois transcritas, dessa forma foi possível a identificação dos desafios (demonstrados na Tabela 2) e suas categorias.

Tabela 2. DLEE - Desafios Levantados nas Entrevistas da Etnografia.

ID	Título	Categoria
01	Documentação sem versionamento	Documentação
02	Parte do conhecimento do negócio não está acessível em formato	Compartilhamento de

	explícito	conhecimento/ Metodologia ágil
03	Demanda indo direto para desenvolvimento (casos pontuais)	Documentação/ Metodologia ágil
04	As dependências entre os módulos são complexas e de difíceis de rastreamento	Documentação/ Compartilhamento de conhecimento
05	Desconhecimento de técnicas de gerenciamento de requisitos	Documentação
06	Ausência de base de conhecimento compartilhada	Compartilhamento de conhecimento
07	A definição dos artefatos prioritários não é clara	Documentação
08	A documentação não é o primeiro artefato produzido na sprint	Metodologia ágil
09	Documentação incompleta, não rastreável e desatualizadas	Documentação
10	Limitações para formação de equipes multidisciplinares	Metodologia ágil
11	Documentação focada apenas na necessidade do cliente	Documentação
12	Dependência operacional do engenheiro de requisitos	Documentação
13	Criação de tarefas sem um modelo mínimo	Documentação
14	Customização externa do produto não gera valor para a instituição	Compartilhamento de conhecimento

A análise de materiais de arquivo foi feita com base na inspeção dos materiais que foram utilizados pelos engenheiros de requisitos no período de observação. Dessa forma foi possível a identificação dos desafios demonstrados, na Tabela 3.

Tabela 3. DLAM - Desafios Levantados nas Análise de Materiais.

ID	Título	Categoria
01	Dificuldades em estabelecer prazos de entrega	Documentação
02	Frequentes gargalos no processo produtivo	Compartilhamento de conhecimento
03	Documentar problemas e decisões tomadas, motivações	Compartilhamento de conhecimento/ Metodologia ágil

Considerando os desafios encontrados, e aqui apresentados, foi utilizado o Diagrama de Venn para melhor visualização de algumas semelhanças percebidas entre estes nas três etapas de coletas de dados (observação, entrevistas e análise de materiais). Quando o desafio é encontrado em mais de uma forma de coleta, isso reforça a sua existência. Desta forma, a Figura 2 apresenta o diagrama de Venn, onde é possível ver as interseções entre os desafios, representados por seus identificadores. O primeiro círculo (azul) representa os desafios encontrados nas observações, o segundo círculo (vermelho) representa os desafios encontrados nas entrevistas e o terceiro círculo (verde) representa os desafios encontrados na análise de materiais.

Nas interseções, pode ser percebida a semelhança entre os desafios. Nas **etapas de observações e entrevistas**: 1) DLOE-2/DLOE-4/DLEE-8: tratam de desafios relacionados a especialização e multidisciplinaridade das equipes; 2) DLOE-6/DLEE-1: tratam de desafios relacionados ao versionamento da documentação; 3) DLOE-7/DLEE-2/DLEE10: tratam de desafios relacionados a dependência do analista de requisitos. Nas **etapas de entrevistas e análise de materiais**: 4) DLEE-4/DLAM-2: tratam de desafios relacionados à ausência de uma base de conhecimento compartilhada. Por fim, o desafio que teve a interseção entre as três **etapas observações, entrevistas e análise de materiais**: 5) DLOE-8/DLEE-9/DLAM-1: tratam de desafios relacionados à adequação da documentação de requisitos para atender à necessidade informacional de seu público-alvo.

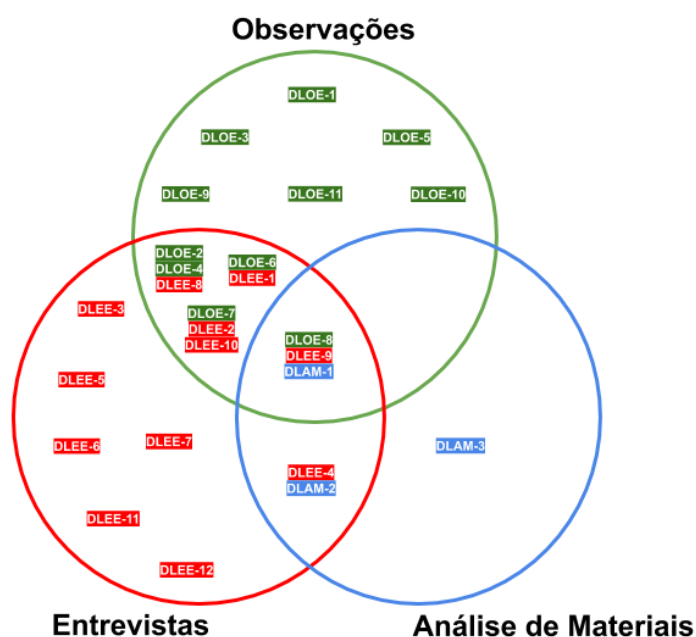


Figura 2. Diagrama de Venn - semelhanças entre desafios

A fim de encaminhar uma solução para os desafios relacionados à adequação da documentação, evidenciados nos itens DLOE-8/DLEE-9/DLAM-1, foram elaborados surveys com o público-alvo da documentação, que podem ser vistos em Souza (2019).

4.2. Proposições de soluções

As soluções propostas foram fundamentadas na literatura acadêmica teórica e empírica, nacional e internacional relacionada às áreas como, por exemplo, Engenharia de Requisitos, Gestão do Conhecimento. Nesta mesma literatura, para problemas identificados e semelhantes aos vividos nas fábricas de software brasileiras, inclusive na estudada, soluções foram testadas e avaliadas, minimamente. Entretanto, neste estudo, constatou-se que muitas pessoas/profissionais desconhecem as alternativas já apontadas. Essas soluções dividem-se em três grandes categorias: i. documentação; ii. compartilhamento do conhecimento; iii. metodologias ágeis.

Em relação a superar as necessidades do público interessado e aos desafios da construção, utilização, e do propósito da **documentação (i)** são sugeridos, considerando os estudos realizados por Barbosa (2018), de Farias Dantas (2017), Nonaka e Takeuchi (1997), Pereira (2018), Santana (2018), Sommerville (2011) e Thakurta (2017): I. registro histórico da interação com o cliente, por este mostrar-se importante em contextos de maior rotatividade das partes e sua produção poderia ser otimizada à medida que o redator responsável praticasse sua capacidade de síntese; II. adoção de ferramenta de versionamento de modo que o código e a especificação fossem preservados ao longo do tempo; III. questionários feitos com público interno e externo identificaram quais eram suas demandas informacionais específicas, o que permitiria a definição de um modelo de especificação de caso de uso com as informações de quem as consomem; IV. nenhuma alteração deveria ser realizada nos produtos sem a realização de uma análise de negócio; V. mapeamento visual de dependências entre módulos, regras e sistemas, que à época era apenas tácito; VI. avaliação sistemática do conteúdo, que foi abordado na capacitação interna, para garantir que foram incluídas estratégias para o gerenciamento de requisitos; VII. estabelecimento de prioridade dentre os artefatos com base na necessidade informacional dos usuários; VIII. adoção de um modelo para criação de tarefas e sua vinculação ao artefato de especificação equivalente.

Enquanto que para superar os desafios para o **compartilhamento de conhecimento (ii)** podem ser propostos, considerando os estudos realizados por Bilessimo (2017), Desidério (2015), Gaspar (2016), Kopf (2018), Mack (2001), Nonaka e Takeuchi (1997), Parker (2014) e Santana (2018): I. estudo da demanda informacional dos analistas de testes e considerar a possibilidade de treinamento/capacitação sobre testes automatizados; II. construção de um mapa de conhecimento e promoção de atividades de transferência do conhecimento; III. estímulo e criação de meios para o compartilhamento de conhecimento a partir da prioridade da construção de conhecimento explícito; IV. construção e manutenção de uma base de conhecimento. Estas iniciativas poderiam contribuir para a redução da dependência da presença dos engenheiros de requisitos para o trabalho da equipe, assim como rodízio entre os projetos e os integrantes, ou oportunidades de iniciar novos projetos; V. incentivo a divulgação e criação de meios para o compartilhamento de customizações feitas por parceiros, e considerar o valor dessas customizações na negociação de contratos futuros; VI. compartilhamento das informações sobre as atividades atuais de cada membro da equipe poderia permitir maior colaboração entre os integrantes da equipe e atuação dos líderes das equipes.

Para vencer os desafios da adoção e utilização de **metodologias ágeis (iii)** podem ser sugeridos, considerando os estudos realizados por Bang (2007), Cleland-Huang (2017), Cohn (2015), Kopf (2018), Leffingwell (2010), Rising (2000), Schwalbe e Sutherland

(2017), Taibi (2017) e Wang (2018): I. adoção de várias equipes menores para um sistema, multidisciplinares e sprints curtas para permitir maior consciência no momento de planejar, reduzir o custo de comunicação, acompanhar melhor as atividades e os impedimentos, além de possibilitar o aumento da produtividade média; II. orientação às equipes sobre a relação da efetividade do planejamento de sprints e a construção de uma memória de grupo; III. adoção de diferentes técnicas de estimativas de itens de backlog, que permitiriam aprimorar estimar o esforço relativo entre os itens, as técnicas aplicadas para relacionar o esforço e o tempo investido em cada tarefa); IV. os requisitos poderiam ser considerados itens de backlog, e a criação destes poderiam ser planejadas, como meta da sprint. Também, seria possível criar sprints, onde toda a equipe estivesse focada na especificação de requisitos. Uma outra abordagem seria iniciar a sprint com um requisito superficialmente detalhado (como uma história de usuário), para depois realizar o aprofundamento durante o desenvolvimento do item; V. ponderação por parte dos líderes sobre a transformação das funções em papéis dentro das equipes; VI. avaliação da estratégia de contratação e de formação de especialistas, com objetivo de compor times multidisciplinares; VII. como sugerido anteriormente, com o compartilhamento das informações por diferentes meios em tempo real dos membros e dos projetos possibilitaria maior colaboração entre os integrantes da equipe e gerenciamento pelos líderes.

5. Discussão dos Resultados

Considerando o estudo realizado e a literatura pesquisada, as soluções podem ser utilizadas por outras instituições, visto que os desafios enfrentados são semelhantes ou de mesma natureza, como podem ser vistos em Souza (2019). Os profissionais estão sobrecarregados, muitas vezes desatualizados, sem ânimo ou estímulo para buscar melhorias para o processo, não contam com políticas, planos ou modelos, que possam se apoiar. O retrabalho mostra a constante má gestão do conhecimento ou falta de compartilhamento das informações entre equipes, clientes (que também são profissionais de TI) e usuários, Fernández *et al.* (2017), Wohrab *et al.* (2018). Isto gera conflito, desinformação e maior custo em todos os níveis (do operacional até para chegar ao usuário final).

A aplicação das metodologias de acordo com um protocolo, previamente, estabelecido possibilitou relacionar e confrontar observações, entrevistas, materiais e documentos, por exemplo. Existiram muitas divergências entre o que foi relatado e observado. Dirimidas a partir da triangulação, que criou relações, e analisá-las possibilitaram dimensionar e categorizar os desafios. Assim, a busca por soluções e melhorias para todo esse ecossistema dentre os próprios interessados e pesquisadores permitiu maior envolvimento das partes, a proposição de soluções práticas e factíveis a todos envolvidos. Como também, o alinhamento ou compensação promovidos pelas diferentes áreas relacionadas Engenharia de Requisitos e Gestão do Conhecimento; bem como aplicação de diferentes metodologias de pesquisa.

Algumas **limitações e ameaças à validade** não puderam ser mitigadas por completo: a primeira delas foram as experiências e os conhecimentos anteriores das pesquisadoras, que podem ter interferido nos desafios encontrados na etnografia. Refletindo sobre cada uma das etapas desta pesquisa foram identificadas: para a etapa de observações - i) a presença física das pesquisadoras no ambiente das equipes pode ter ocasionado mudanças no comportamento dos observados; ii) a ideia inicial do estudo era observar a equipe documentando seus requisitos para encontrar possíveis desafios, mas, com o início da etnografia, foi sabido que as equipes não estavam criando e manipulando a especificação no

ambiente de trabalho, por causa da dificuldade em fazê-lo em um ambiente compartilhado em consequência das inúmeras interrupções; iii) o registro do contato dos engenheiros de requisitos com os materiais utilizados durante o período de observação foi realizada apenas em parte do horário de expediente; para a etapa das entrevistas: i) no roteiro de entrevistas foi utilizado um vocabulário (preservação de informação, recuperação de informação, base de conhecimento, por exemplo) que talvez tenha, inicialmente, dificultado o entendimento de algumas perguntas pelos entrevistados; ii) as perguntas do roteiro de entrevista eram abertas, para que houvesse maior exploração das respostas e identificação do envolvimento com o restante da equipe, entretanto, como a interpretação ficava a cargo entrevistado, algumas respostas ficaram com o foco diferente do desejado.

6. Trabalhos Relacionados

Foram encontrados dois trabalhos relacionados a esta pesquisa. O primeiro deles foi um que realizou uma etnografia de caráter organizacional também com o objetivo de consolidar práticas da Engenharia de Requisitos, Wohrab *et al.* (2018). Este estudo acompanhou um projeto de três anos em uma empresa industrial de grande porte, que tentou consolidar práticas da Engenharia de Requisitos, e personalizar uma solução de ferramenta às necessidades da empresa, mantendo a autonomia de unidades individuais. Foram apresentados os desafios da empresa, e compartilhado as estratégias de mitigação com base nas lições aprendidas. Diferente deste estudo, Wohrab *et al.* (2018) buscaram personalizar uma ferramenta para os desafios em Engenharia de Requisitos.

Já o segundo trabalho, data de 2017, e é mais um resultado das pesquisas elaboradas pelo grupo NaPiRE. Neste trabalho, Fernandez et al. (2017) tinham como objetivo de conhecer os desafios da Engenharia de Requisitos, em diversas indústrias. Dos desafios encontrados foi elaborada uma lista pré-elaborada, que foram escolhidos pelas empresas. A diferença deste estudo está que o instrumento utilizado foi um survey, onde os participantes escolhiam quais eram os desafios enfrentados.

7. Considerações Finais

A etnografia organizacional realizada permitiu identificar diferentes desafios relacionados à Engenharia de Requisitos em uma fábrica de software de grande porte, que não divergiram de outras fábricas de software ou da literatura pesquisada. A literatura já indicava que equipes grandes - como a observada nesta pesquisa - tendem a possuir dificuldades para comunicar-se e compartilhar conhecimento. Então, as soluções foram classificadas em três categorias, que emergiram da aplicação metodologia de Grounded Theory adaptada: documentação, compartilhamento de conhecimento, e metodologias ágeis.

Sendo que as categorias de documentação e de compartilhamento de conhecimento em equipes, que adotam metodologias ágeis de desenvolvimento, eram o foco deste estudo. Como a documentação tem o objetivo de compartilhar o conhecimento - como um retrato do conhecimento atual -, e assim formalizar decisões. A decisão foi investigar como isto estava sendo feito na instituição escolhida, mas também como poderia ser melhorada, e potencializada dentro da metodologia (Scrum), parcialmente, utilizada pelas equipes escolhidas. Além de descobrir desafios, este estudo sugeriu possíveis soluções com aplicação de práticas advindas da gestão do conhecimento e da engenharia de requisitos em comunhão com o que é preconizado com a Scrum (metodologia ágil).

Trabalhos futuros, que já constam no planejamento, são: o treinamento, a customização e as apresentações das soluções e o melhoramento das práticas de Scrum propostas. Além de uma investigação junto aos clientes sobre necessidades e dificuldades.

Agradecimentos

As autoras agradecem o financiamento parcial desta pesquisa da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001; o apoio dos participantes deste estudo realizado; e os responsáveis e os participantes do Laboratório de Especificação e Teste de Software (UFRN/LETS).

As autoras também agradecem o convite dos editores do periódico Cadernos do IME - Série Informática; e destacam que o presente artigo é uma versão ampliada e revisada de um trabalho anterior publicado no XXII Workshop sobre Engenharia de Requisitos (WER19), intitulado “Aplicação da Etnografia no Contexto de Fábrica de Software na Perspectiva da Engenharia de Requisitos”, Souza *et al.* (2019).

Referências

- Angrosino, M. (2009) “Etnografia e Observação Participante”. Coleção Pesquisa Qualitativa coordenada por Uwe Flick (eds.). Porto Alegre: Artmed.
- Bang, T. J. (2007) “An Agile Approach to Requirement Specification”. Em: International Conference on Extreme Programming and Agile Processes in Software Engineering. Springer, Berlin, Heidelberg. p. 193-197.
- Barbosa, G. S. e da Silva, E. O. (2018) “Geração de Informações Gerenciais para Sistemas de Controle de Versão: um Estudo de Caso Utilizado o GitHub”. Caderno de Estudos em Sistemas de Informação, v. 5, n. 1.
- Bellenzier, M., Audy, J., Prikladnicki, R. e Luciano, E. (2015) “How the Scrum Adoption Relates to Productivity of Software Development Teams?”. Em: 2015 6th Brazilian Workshop on Agile Methods (WBMA). p. 1-5. IEEE.
- Bilessimo, S. M. S., Esteves, P. C. L., Barros, A. F. F., e Vargas, C. M. (2017) “Práticas de Gestão de Conhecimento: um Estudo de Caso em uma Instituição de Ensino Federal”. Revista Gestão Universitária na América Latina-GUAL, v. 10, n. 3, p. 78-96.
- Charmaz, K. e Mitchell, R. G. (2001) “Grounded Theory in Ethnography. Handbook of Ethnography”, v. 160, p. 174.
- Chau, T. e Maurer, F. (2004) “Knowledge Sharing in Agile Software Teams”. Em: Logic versus approximation. p. 173-183. Springer, Berlin, Heidelberg.
- Cleland-Huang, J. (2017) “Linhas de Produtos de Engenharia Reversa em Ambientes Ágeis: Lições Aprendidas e Desafios”. Em: Proceedings dos Sistemas Internacionais 21 e Software Product Line Conference - Volume Um . ACM. p. 5-5.
- Cohn, M. (2005) “Agile Estimating and Planning”. Pearson Education.
- De Farias Dantas, N. C. e Carvalho, L. M. (2017) “Necessidades de Informação de Gestores Públicos: o Caso de um Centro Acadêmico Universitário”. Informação em Pauta, v. 2, p. 160-180.

- Desidério, P. H. M. e Popadiuk, S. (2015) “Redes de Inovação Aberta e Compartilhamento do Conhecimento: Aplicações em Pequenas Empresas”. RAI Revista de Administração e Inovação, v. 12, n. 2, p. 110-129.
- Fernandéz, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., et al. (2017) “Naming the Pain in Requirements Engineering”. Empirical Software Engineering, v. 22, n. 5, p. 2298-2338.
- Gaspar, M. A., Santos, S. A., Donaire, D., Kuniyoshi, M. S., e Prearo, L. C. (2016) “Gestão do Conhecimento em Empresas Atuantes na Indústria de Software no Brasil: um Estudo das Práticas e Ferramentas Utilizadas”. Inf. & Soc.:Est., João Pessoa, v.26, n.1, p. 151-16.
- Koff, M., Sauermann, V. e Frey, F. (2018) “Implement Communities of Practice in an Agile IT Environment”. Proceedings of the 23rd European Conference on Pattern Languages of Programs - EuroPLOP.
- Leffingwell, D. (2010) “Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise”. Addison-Wesley Professional.
- Mack, R., Ravin, Y., e Byrd, R. J. (2001) “Knowledge Portals and the Emerging Digital Knowledge Workplace”. IBM systems journal, v. 40, n. 4, p. 925-955.
- Neyland, D. (2008) “Organizational Ethnography”. Los Angeles, London, New Delhi, Singapore: Sage Publications.
- Nonaka, I. e Takeychi, H. (1997) “Criação de Conhecimento na Empresa: Como as Empresas Japonesas Geram a Dinâmica da Inovação”. Rio de Janeiro: Elsevier.
- Pohl, K., e Rupp, C. (2015) “Fundamentos de Engenharia de Requisitos: um Guia de Estudo para o Profissional - Certificado para Exame de Engenharia de Requisitos - Nível-base Compatível com IREB”. Rocky Nook, Inc.
- Parker, M. E. F. e Del Monte, Y. F. (2014) The Agile Management of Development Projects of Software Combining Scrum, Kanban and Expert Consultation. In: IFIP International Conference on Open Source Systems. Springer, Berlin, Heidelberg. p. 176-180.
- Pereira, F. C. M., Jordão, R. V. D., Netto, M. C. C., e da Cunha Duarte, L. (2018) “Estudo de usuário: necessidades informacionais de empresas contratantes de pesquisas de mercado”. Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação, v. 23, n. 51, p. 108-122.
- Rising, L. e Janoff, N. S. (2000) “The Scrum Software Development Process for Small Teams”. IEEE software, v. 17, n. 4, p. 26-32.
- Rossini, A. e Palmisano, Â. (2003) “Administração de Sistemas de Informação e a Gestão do Conhecimento”. São Paulo: Thomson.
- Santana, Y. D., e Garcia, G. P. C. (2018) “Os Usuários da Informação no Arquivo: uma Aproximação a seus Estudos desde as Ciências da Informação”. e-Ciencias de la Información.
- Schwalbe, K e Sutherland, J. (2019) “Scrum Guide 2017”. Disponível em: <<https://www.scrum.org/>>. Acessado em: 07/02/2019.
- Sommerville, I. (2011) Engenharia de Software, 9ª edição. Pearson.

Souza, L.T.M. (2019) “Documentação de Requisitos e Compartilhamento de Conhecimento: uma Proposta a partir de um Estudo Etnográfico”. Dissertação de mestrado. Universidade Federal do Rio Grande do Norte, Pós-graduação em Sistemas e Computação.

Souza, L., Miranda, E., Lucena, M., e Gomes, A. (2019) "Aplicação da Etnografia no Contexto de Fábrica de Software na Perspectiva da Engenharia de Requisitos". Em 22th Workshop on Requirements Engineering (WER2019).

Sutherland, J. e Schwaber, K. (2013) “O Guia do Scrum. O Guia Definitivo para o Scrum: as Regras do Jogo”. Scrum org , v. 268.

Taibi, D., Lenarduzzi, V., Ahmad, M. O., e Liukkunen, K. (2017) “Comparing Communication Effort Within the Scrum, Scrum with Kanban, XP, and Banana Development Processes”. Em: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. ACM. p. 258-263.

Terra, J. (2000) “Gestão do Conhecimento: o Grande Desafio Empresarial”. Rio de Janeiro: Negócios.

Thakurta, R. (2017) “Understanding Requirement Prioritization Artifacts: a Systematic Mapping Study”. Requirements Engineering, v. 22, n. 4, p. 491-526.

Wang, Z. (2018) “Teamworking Strategies of Scrum Team: A Multi-Agent based Simulation”. Em: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. ACM. p. 404-408.

Wohrab, R., Pelliccione, P., Knauss, E. e Gregory, S. (2018) “The Problem of Consolidating RE Practices at Scale: An Ethnographic Study”. Em: International Working Conference on Requirements Engineering: Foundation for Software Quality. p. 155-170. Springer, Cham.

Ybema, S., Yanow, D., Wells, H. e Kamsteeg, F. (eds.). (2009) “Organizational Ethnography: Studying the Complexities of Everyday Life”. Los Angeles, London, New Delhi, Singapore, Washington Dc: Sage Publications.

CADERNOS DO IME

Série Informática

Normas para publicação de trabalhos

A revista Cadernos do IME - Série Informática é um veículo para divulgação de trabalhos acadêmicos, científicos ou profissionais nas áreas de informática, ciência da computação e correlatas, sob responsabilidade do Departamento de Informática e Ciência da Computação (DICC) do Instituto de Matemática e Estatística (IME) da Universidade do Estado do Rio de Janeiro (UERJ).

A revista é semestral, sendo publicada em junho e dezembro. São aceitos trabalhos inéditos e que apresentem elementos de originalidade, em português, inglês ou espanhol.

Os Cadernos do IME aceitam os seguintes tipos de contribuições:

- Artigo técnico, trabalho de pesquisa com elementos de originalidade;
- Tutorial, texto introdutório a áreas específicas da ciência da computação, com revisão de literatura e organização dos conceitos relacionados;
- Resenha / Revisão, revisão de literatura e análise crítica de produtos da área de ciência da computação e informática;
- Resumo, texto destinado à informação sobre trabalhos de graduação e de pós-graduação;
- Comunicação, texto destinado à divulgação de opiniões, pesquisas em andamento, lançamentos e divulgação de eventos.

Os artigos e tutoriais não devem ultrapassar 20 páginas e as demais formas de publicação devem se restringir a no máximo 10 páginas.

Todos os trabalhos enviados para publicação nos Cadernos do IME – Série Informática, independente de tipo de contribuição, devem seguir o modelo para publicação de artigos da SBC – Sociedade Brasileira de Computação.

Os arquivos digitais no formato fonte (.DOC) e PDF deverão ser encaminhados aos editores da série.

A submissão é eletrônica e em fluxo contínuo no endereço <http://www.e-publicacoes.uerj.br/index.php/cadinf>. As chamadas de trabalho para volumes especiais estão disponíveis <http://www.ime.uerj.br/cadernos/cadinf/>.

Os conteúdos e pontos de vista expressos nos trabalhos publicados são de inteira responsabilidade dos autores.



Ruy Garcia Marques
Reitor

Maria Georgina Muniz Washington
Vice-Reitora

Luis Antonio Campinho Pereira da Mota
Diretor do Centro de Tecnologia e Ciências

Geraldo Magela da Silva
Diretor do Instituto de Matemática e Estatística

Sérgio Luiz Silva
Vice-Diretor do Instituto de Matemática e Estatística

Mauricio Alejandro Antonucci Vilches
Departamento de Análise Matemática

Ricardo Galdo Camelier
Departamento de Estruturas Matemáticas

Jaime Velasco Câmara da Silva
Departamento de Geometria e Representação Gráfica

Alexandre da Costa Sena
Departamento de Informática e Ciência da Computação

José Fabiano Serra Costa
Departamento de Estatística

Marcus Vinicius Tovar Costa
Departamento de Matemática Aplicada