

Requisitos de *Software* para Sistemas Complexos: Aplicando *Design Science Research* para uma Abordagem de Elicitação de Requisitos com o Método de Análise da Ressonância Funcional (FRAM)

Title: Software Requirements for Complex Systems: Applying Design Science Research to a Requirements Elicitation Approach with the Functional Resonance Analysis Method (FRAM)

Elaine A. Carvalho¹, José O. Gomes¹, Alessandro Jatobá², Mônica F. Silva¹, Paulo Victor R. Carvalho¹

¹Programa de Pós-Graduação em Informática – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ – Brasil

²Centro de Estudos Estratégicos – Fundação Oswaldo Cruz (FIOCRUZ)
Rio de Janeiro, RJ – Brasil

nane.alves@gmail.com, joseorlando@nce.ufrj.br,
alessandro.jatoba@fiocruz.br, monica.silva@ppgi.ufrj.br,
paulov@ien.gov.br

Abstract. *The requirement elicitation task is still considered critical, especially when dealing with complex (non-linear) systems that need technological support with more resilient performance. That is, software capable of dealing with situations of uncertainty. The Functional Resonance Analysis Method (FRAM), based on Resilience Engineering concepts, is used to model the aspects associated with “work as done”, making the unpredictability of tasks more evident. This work, as an extended version of another article published in SBSI 2021, intends to detail aspects related to the application of Design Science Research in the projection of a heuristic model aimed at eliciting software requirements for complex systems as Healthcare supported by a computational tool, the ReqFRAM. Thus, it is intended to detail the steps for this study, the proposed heuristics, and their contributions to Requirements Engineering from the concepts of Resilience Engineering.*

Keywords. *Requirements Elicitation; Software Specification; Complex System; FRAM; Resilience Engineering.*

Resumo. *A tarefa de elicitação de requisitos ainda é considerada crítica, principalmente quando se trata de sistemas complexos (não lineares), que necessitam de suporte tecnológico com desempenho mais resiliente. Ou seja,*

softwares capazes de lidar com situações de incerteza. O Método de Análise da Ressonância Funcional (FRAM), método de modelagem proveniente da Engenharia de Resiliência, é usado para representar os aspectos associados ao “trabalho como feito”, tornando mais evidente as imprevisibilidades das tarefas. Esse trabalho, como uma versão estendida de outro artigo publicado no SBSI 2021, pretende detalhar aspectos relacionados à aplicação da Design Science Research na projeção de um modelo heurístico destinado à elicitação de requisitos de software para sistemas complexos como a Saúde com suporte de uma ferramenta computacional, o ReqFRAM. Assim, espera-se mostrar com mais detalhes os passos do estudo realizado, as heurísticas propostas e suas contribuições para a Engenharia de Requisitos, à luz dos conceitos da Engenharia de Resiliência.

Palavras-Chave. Elicitação de Requisitos; Especificação de Software; Sistemas Complexos; FRAM; Engenharia de Resiliência.

1. Introdução

A incompreensão das necessidades e desejos do cliente, bem como o conhecimento inadequado do domínio do problema, geralmente, estão relacionados às falhas em projetos de desenvolvimento de *software*. Nem sempre é possível garantir que os projetos de desenvolvimento de sistemas de informação atendam a todos os requisitos desejados ou necessários [Mayer *et al.* 2014]. No início do processo de desenvolvimento do *software*, tarefa denominada elicitação de requisitos, os engenheiros de requisitos e desenvolvedores são desafiados a descobrir, entender e especificar os requisitos do usuário. Erros cometidos nas atividades iniciais da Engenharia de Requisitos podem ser responsáveis pelo aumento considerável do custo do projeto de desenvolvimento, resultando até na rejeição do sistema de informação pelos usuários [McAllister 2006].

Os sistemas sociotécnicos têm uma arquitetura de uso intensivo de recursos humanos em larga escala, combinando suporte tecnológico (recursos de *hardware* e *software*) e interfaces humanas [Frost and Mo 2014]. Na concepção da Engenharia de Resiliência, sistemas complexos têm característica de auto-organização (podem mudar para se adaptar à necessidade do ambiente), coexistência (modificação do seu comportamento para responder às interações com outros sistemas dos quais ele não pode operar de forma isolada) e autorreplicação (capacidade de gerar sistemas semelhantes) [Frost and Mo 2014]. Nesses sistemas, o comportamento emergente é muito comum e não é facilmente previsível, dificultando ainda mais a descoberta dos requisitos tecnológicos no desenvolvimento do *software*. Portanto, torna-se crítico considerar essas variabilidades desde o início do desenvolvimento do *software* para que corresponda de forma confiável ao “trabalho como feito” (“Work-As-Done” - WAD) [Hollnagel 2012]. Dependendo da forma como a tarefa de elicitação de requisitos é conduzida, os analistas ou desenvolvedores podem não saber, à priori, a melhor maneira de empregar a tecnologia para lidar com comportamentos e variabilidades que surgirão durante a operação do *software* [Hollnagel 2012]. Portanto, considerar as características próprias desses tipos de sistemas como fatores atuantes no funcionamento dessas tecnologias desde o início do projeto de desenvolvimento – ainda na fase de elicitação de requisitos – torna-se importante para garantir razoavelmente que o *software* desenvolvido corresponda às expectativas dos usuários, tendo um desempenho potencialmente mais resiliente [Hollnagel 2012]. A Engenharia de Resiliência também vem colaborar com a Engenharia

de Software nesse contexto, fornecendo conceitos, métodos e técnicas potenciais para a descoberta de requisitos para sistemas sociotécnicos complexos.

O método de Análise de Ressonância Funcional (Functional Resonance Analysis Method - FRAM) [Hollnagel 2012] é utilizado justamente para modelar as variabilidades dos sistemas complexos, sendo bastante adequado à captura das características essenciais do WAD (suas funções, seu desempenho conjunto e suas interdependências, suas variabilidades ocasionadas por inúmeros fatores de natureza sociotécnica) [Clay-Williams *et al.* 2015].

Apesar da existência dos novos métodos de modelagem ligados à Engenharia de Resiliência [Hollnagel *et al.* 2006] como FRAM [Hollnagel 2012], STAMP [Levenson 2004] e diversos outros métodos para estruturação de problemas em sistemas complexos [Rasmussen 1997], ainda são poucas as iniciativas que sistematizam as contribuições desses métodos ou modelos para a Engenharia de Software, mais especificamente para a Engenharia de Requisitos [Baxter and Sommerville 2011]. Esse trabalho, como uma versão estendida do artigo Alves Carvalho *et al.* (2021) publicado no Simpósio Brasileiro de Sistemas de Informação (SBSI) 2021, apresenta com mais detalhes a aplicação da *Design Science Research* para a construção de um modelo sistematizado por heurísticas e orientado ao modelo FRAM para identificar requisitos de *software* para os sistemas sociotécnicos complexos como a Saúde.

1.1. Objetivo e Problema da Pesquisa

Muitos são os trabalhos que buscam sistematizar as contribuições da modelagem de negócios como ponto de partida para a elicitação de requisitos [Bitencourt *et al.* 2016]. Apesar da ampla aplicação, a modelagem de processos tradicional em muitos casos não considera as variabilidades (não linearidades) presentes nos sistemas sociotécnicos complexos, que impactam no funcionamento dos *softwares* de apoio, que, muitas vezes, não estão preparados para atender às necessidades desse ambiente variável e com necessidade de adaptação [Hollnagel 2012]. Já no âmbito da Engenharia de Requisitos, é possível encontrar trabalhos que tratam os processos a nível de execução [Do Prado Leite *et al.* 2016] e abordagens orientadas a objetivos [Yu *et al.* 2008] que também consideram a variabilidade.

No entanto, esses trabalhos não abordam os aspectos sociotécnicos adotados na perspectiva da Engenharia de Resiliência. Ainda são poucas as iniciativas que sistematizam as contribuições do método FRAM ao representar o “trabalho como feito” e suas variabilidades para a Engenharia de *Software* (ES) [Baxter and Sommerville 2011]. Já na perspectiva da Engenharia de Requisitos, a forma tradicional de realizar a tarefa de elicitação de requisitos nem sempre permite entender corretamente a complexidade do contexto ou antecipar os possíveis erros oriundos de situações inesperadas, ou seja, da variabilidade com que o dispositivo de TI precisará lidar. Na perspectiva do contexto sociotécnico complexo, o *design* de dispositivos tecnológicos específicos pode levar a incidentes inesperados e de graves proporções se as variabilidades forem desconsideradas [Katina *et al.* 2014]. Pesquisas que contribuam para o aprimoramento da tarefa de elicitação de requisitos aplicada aos sistemas complexos (coleta de informações, experiências e preferências do usuário) se tornam importantes no desenvolvimento adequado dos produtos tecnológicos [Azevedo and Campos 2008].

Assim, o problema da pesquisa pode ser enunciado como a dificuldade de identificar requisitos de *software* para sistemas sociotécnicos complexos que incorporem o tratamento da variabilidade. Já a questão da pesquisa pode ser definida da seguinte maneira: “como o modelo FRAM (suas informações, principalmente, a variabilidade) pode ser utilizado na tarefa de elicitação de requisitos para identificar funcionalidades que incorporem ao *software* características que proporcionem um comportamento considerado mais resiliente nos sistemas sociotécnicos complexos?” Portanto, o objetivo principal é desenvolver um modelo baseado em heurísticas fundamentadas nos conceitos da Engenharia de Requisitos e de Resiliência para orientar a tarefa de elicitação de requisitos funcionais e não funcionais a partir das informações dos modelos FRAM.

2. Elicitação de Requisitos

Os requisitos de *software* são sentenças que expressam as necessidades dos clientes e que determinam a qualidade do sistema [Leite 2001]. Dentre as várias classificações atribuídas pela comunidade acadêmica, é reconhecido que os requisitos se dividem basicamente em dois tipos principais: funcionais e não funcionais [Cysneiros and Leite 2001].

- Funcionais (RF): são declarações sobre as ações do *software* e descrevem como ele deve reagir com entradas específicas e como deve se comportar em determinadas situações;
- Não funcionais (RNF): são comportamentos ou qualidades específicas que o sistema de informação deve ter.

Sommerville (2003) define também um tipo de requisito num nível superior denominado "requisito de negócio", que corresponde a descrições de necessidades de negócio. São requisitos que abordam objetivos, processos, funções e outras características organizacionais a serem apoiadas pelo futuro *software*. Assim, os requisitos de negócio são intimamente relacionados aos processos de negócio, já que estes representam as atividades realizadas pelos usuários no ambiente organizacional [De Carvalho *et al.* 2021]. Portanto, para definir conceitualmente um *software* é preciso considerar que a sua complexidade é determinada em parte por suas funcionalidades (aquilo que é executado) e em parte pelos requisitos não funcionais associados à performance, confiabilidade, manutenibilidade, portabilidade, entre outros [Cysneiros and Leite 2001]. Essas características desempenham papel crítico no desenvolvimento do produto tecnológico já que erros, devido à não elicitação ou elicitação incorreta dessas necessidades, estão entre os mais caros e difíceis de corrigir [Cysneiros and Leite 2001]. Nos contextos organizacionais (sociotécnicos) é importante realizar a tarefa de elicitação considerando também esses requisitos não funcionais, principalmente àqueles relacionados à minimização dos riscos organizacionais relacionados à resiliência [Jaramillo and Palacios 2014].

A elicitação é a primeira tarefa do processo de requisitos e corresponde à descoberta das necessidades do *software*, identificação das fontes de informação, coleta de fatos e comunicação [Leite 1994]. Ela é considerada a etapa mais importante, decisiva e ao mesmo tempo mais crítica do desenvolvimento de *software* [Christel and Kang 1992], pois afeta todo o processo da Engenharia de Requisitos (ER). O processo da ER é formado por atividades que derivam, validam e atualizam o documento de requisitos do *software* [Méndez Fernández and Wagner 2015].

O modelo heurístico proposto nesse trabalho tem como foco de atuação essa primeira tarefa do processo da ER, adotando a orientação a modelo, no caso o FRAM, como abordagem sociotécnica para compreensão das variabilidades e necessidades de adaptação do contexto complexo demandante de um apoio tecnológico mais resiliente [Clay-Williams *et al.* 2015].

3. Engenharia de Resiliência

A Engenharia de Resiliência é uma perspectiva da chamada Safety-II, na qual há garantia de que "o máximo possível de coisas dão certo" [Hollnagel *et al.* 2015]. Portanto, há relação direta com a capacidade do sistema ter sucesso em condições variadas, pois pressupõe-se que a variabilidade forneça as adaptações necessárias, sendo o motivo para que as coisas aconteçam adequadamente [Hollnagel *et al.* 2015]. As pessoas, nessa perspectiva, são consideradas um recurso necessário para garantir a flexibilidade do sistema e, conseqüentemente, a resiliência [Hollnagel *et al.* 2015].

A Engenharia de Resiliência surge para lidar de forma inovadora com os sistemas sociotécnicos complexos, tratando a resiliência não somente como a capacidade de se adaptar [Woods and Branlat 2011], mas também a capacidade de se adequar às mudanças ou, até mesmo, a estabilidade do meio ambiente para atender aos seus requisitos operacionais atuais e futuros [Sundström and Hollnagel 2006]. É, principalmente, nesse aspecto que o presente trabalho pretende dar suas contribuições, incorporando à tarefa de elicitação de requisitos conceitos que auxiliem na descoberta de características funcionais e não funcionais habilitadoras de comportamentos mais resilientes para os *softwares* em ambientes sociotécnico complexos.

3.1. Sistemas Complexos

Nesse trabalho, o uso do adjetivo “complexo” não deve ser interpretado como sinônimo de “complicado”. “Complexo” refere-se às características próprias de sistemas constituídos por partes com comportamentos simples que se influenciam mutuamente, gerando um comportamento complexo global [Marino and Campos 2015]. Também associado ao termo sistema complexo encontra-se na literatura [Katina *et al.* 2014] a caracterização de sistemas imensos em contextos sujeitos à combinação de fatores conflitantes, além de ambigüidade, incerteza, informações incompletas, incorretas, inexistentes ou sujeitas a mudanças e com insuficiência de recursos. Essa pesquisa não trata desses tipos de sistemas e nem daqueles ditos complicados, cuja variabilidade pode ser prevista, sendo possível avaliar a melhor forma de lidar com seus resultados e definir à priori tecnologia e até automação.

No caso dos sistemas sociotécnicos complexos não é possível saber a melhor maneira de lidar com a variabilidade devido à imprevisibilidade. Portanto, o objetivo do *software* nesses contextos é prover algum suporte tecnológico para amortecer os efeitos da variabilidade e facilitar o trabalho das pessoas, permitindo resolver situações esperadas ou inesperadas [De Carvalho *et al.* 2020]. Dessa forma, um sistema complexo não pode ter seu comportamento antecipadamente determinado (não-determinismo), ser decomposto em partes estáveis (decomponibilidade funcional limitada), ter informação dispersa com funções não posicionadas (natureza distribuída da informação e representação) e, por fim, ter situações imprevisíveis e não lineares (emergência e auto-organização).

Já o conceito sociotécnico está relacionado às interações entre as pessoas, tecnologia e organização, que formam uma estrutura organizada, cujas funções estão interligadas para atingir um objetivo [Marino and Campos 2015]. Portanto, considerar e incorporar uma abordagem sociotécnica à tarefa de elicitação de requisitos para essas tecnologias torna-se uma contribuição relevante.

3.2. Método de Análise da Ressonância Funcional (FRAM)

O Método de Análise de Ressonância Funcional (do inglês *Functional Resonance Analysis Method* - FRAM) foi introduzido inicialmente como estratégia para a investigação de acidentes e avaliação de segurança [Hollnagel 2004], mas tem sido aplicado, por exemplo, em operações marítimas e de emergência, na Saúde (para diretrizes clínicas e definição de indicadores), nos setores aéreo, nuclear, ferroviário e em diversas aplicações da tecnologia da informação para aprimorar o suporte aos sistemas sociotécnicos complexos [Patriarca *et al.* 2020].

O FRAM tem quatro princípios básicos [Hollnagel 2004]:

1. Equivalência de sucesso e falhas: considera falha apenas o lado negativo das adaptações necessárias à complexidade do mundo real e sucesso consequência da capacidade de antecipar a mudança antes que ocorram danos;
2. Ajustes aproximados: trata as condições operacionais subespecificadas e passíveis de mudanças mais ou menos ordenadas, sendo inevitável a realização de ajustes aproximados;
3. Emergência: considera a possibilidade de combinação da variabilidade de múltiplas funções de maneira inesperada, criando resultados desproporcionais e efeitos não lineares, tornando tanto a falha quanto o desempenho esperado (considerado normal) fenômenos emergentes;
4. Ressonância funcional: considera a possibilidade de reforço das variabilidades de várias funções, causando também variabilidade em outra função para além de seus limites normais, não sendo possível entender um sistema pela decomposição dos seus componentes isolados.

O Método de Análise de Ressonância Funcional (FRAM) [Hollnagel 2004; 2012] propõe uma metodologia para identificar e avaliar a variabilidade do desempenho. O FRAM se baseia na modelagem funcional e compartilha dos conceitos de Engenharia de Resiliência sobre a subespecificação dos sistemas sociotécnicos complexos, reconhecendo nela a necessidade de ajustes. Assim, o FRAM é um método para modelar sistemas sociotécnicos complexos com base nos conceitos e preceitos da Engenharia de Resiliência. O foco é a análise das funções que compõem as atividades do sistema e seus acoplamentos, bem como os resultados que emergem dessas relações, independentemente de serem planejados ou não, esperados ou inesperados [Hollnagel and Slater 2018]. A variabilidade das funções do sistema é um diferencial importante desse método e ajuda a identificar como elas podem (ou não) refletir numa questão de segurança (HOLLNAGEL *et al.*, 2006).

Há cinco passos previstos para a aplicação do método FRAM (Hollnagel 2004; Hollnagel and Slater 2018):

1. Definição do objetivo da análise para identificar as principais funções a serem modeladas;
2. Identificação e descrição das principais funções do sistema com descrição das características básicas (chamadas aspectos);
3. Avaliação da variabilidade potencial para caracterizar as variações reais de cada função em uma ou mais instâncias;
4. Identificação da ressonância funcional com base nas dependências e acoplamentos entre as funções e de acordo com a variabilidade potencial ou real;
5. Identificação de contramedidas efetivas a serem introduzidas no sistema, desenvolvendo recomendações para monitorar e gerenciar as variabilidades, criando mecanismos de amortecimento para atenuar os resultados indesejáveis ou aumentar os efeitos positivos esperados.

Na modelagem FRAM, a identificação das funções é orientada pela descrição das atividades normais realizadas pelo sistema sociotécnico. Assim, o modelo contém a descrição verbal das funções, da variabilidade, incluindo seus seis aspectos: Entrada, Saída, Pré-Condição, Controle, Tempo e Recursos, conforme ilustrado na Figura 1.

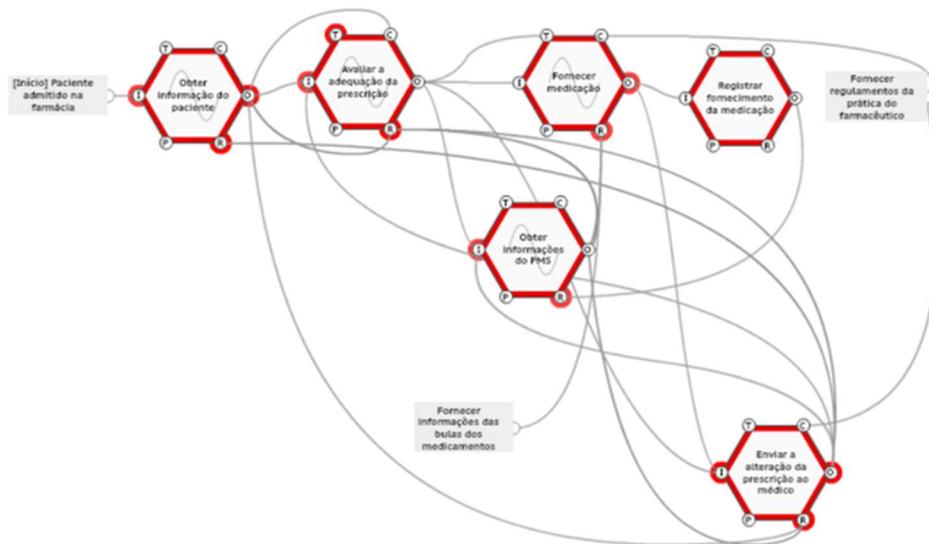


Figura 1. Exemplo de um modelo FRAM

Cada função do FRAM possui seis aspectos, a saber: Entrada, Saída, Pré-Condição, Controle, Tempo e Recursos. Hollnagel (2012) define os seis aspectos da seguinte forma:

- Entrada (I): aquilo que a função processa ou transforma, ou o que inicia a função;
- Saída (O): aquilo que é o resultado da função, mudança de estado ou um produto específico;
- Pré-condição (P): condições que devem existir antes que a função possa ser executada;
- Recursos (R): aquilo de que a função necessita ou deve consumir para que produza o resultado (a saída);

- Tempo (T): restrições temporais que afetam a função (afeta o tempo de início ou término, bem como a duração);
- Controle (C): aspecto que denota como a função é monitorada ou controlada.

O modelo FRAM permite gerar um conjunto de possíveis instanciações para mostrar o efeito das condições reais de trabalho no desempenho do sistema complexo [Macchi 2010]. A Ferramenta *FRAM Model Visualiser* – FMV é empregada na construção do modelo e na elaboração de suas descrições, permitindo ainda a exportação para vários formatos. O método FRAM foi escolhido para modelagem de sistemas complexos nessa pesquisa porque: I) possui ampla aplicação no estudo de sistemas complexos como a Saúde [Clay-Williams *et al.* 2015], foco da avaliação experimental; II) tem sua ênfase na variabilidade, permitindo a identificação dos principais aspectos que afetam a adaptação do sistema [Saurin *et al.* 2017]; III) gera um modelo funcional (e não estrutural), produzindo informações úteis para a descoberta dos requisitos tecnológicos [Frost and Mo 2014]. Todos esses itens são considerados centrais para a construção da solução proposta.

3.3. Variabilidade

O modelo FRAM tem como diferencial a análise do acoplamento entre as funções para entender como a variabilidade e os ajustes podem afetar outras funções e, portanto, o cenário como um todo. Assim, as funções podem se amortecer mutuamente (absorver a variabilidade), criando estabilidade no cenário analisado; ou se reforçar mutuamente (amplificar a variabilidade), tornando instável esse cenário e levando a resultados inesperados e geralmente também indesejados [Hollnagel and Slater 2018]. Assim, a descrição desses acoplamentos pode ser usada para entender os resultados emergentes e promover ações de monitoramento e gerenciamento dos riscos nesses contextos sociotécnicos. Por isso, a caracterização da variabilidade no modelo FRAM é o ponto de partida para entender como as funções podem se interconectar e de que forma isso gera os resultados inesperados. Para tal, o foco da análise deve ser a variação detectada na Saída da função, pois significa que aquela foi capaz de alterar a qualidade do seu produto [Hollnagel and Slater 2018].

Há basicamente três tipos de variações possíveis na Saída:

- Interna ou endógena: é resultado da variação que ocorre na própria função;
- Externa ou exógena: pode ocorrer devido às mudanças do ambiente de trabalho, isto é, às condições sob as quais a função é desempenhada;
- Ressonância funcional: variação ocorrida no aspecto Entrada, Pré-condição, Controle ou Tempo de outras funções.

Uma função pode ainda ter a combinação desses três tipos de variação ao mesmo tempo. O modelo FRAM explica como os ajustes aproximados levam a resultados inesperados e como esses resultados não lineares acontecem, se mostrando informação útil para o UdI e, portanto, para a criação de requisitos capazes de amortecer potenciais riscos na operação dos usuários. Assim, é uma informação importante para a tarefa de elicitação de requisitos. Existem outras abordagens da Engenharia de Software [Yu *et al.* 2008], empenhadas em criar uma base para projetar variabilidade para *software*

personalizável ou adaptável. Contudo, esse não é o mesmo conceito de variabilidade (baseado na Engenharia de Resiliência) que está sendo abordado nesse trabalho.

4. Metodologia da Pesquisa

A Design Science, no contexto dessa pesquisa, é responsável pela concepção e validação da solução proposta, sendo a base epistemológica, e a *Design Science Research* (DSR) é o método que operacionaliza a construção do conhecimento no contexto estudado [Chakrabarti 2010]. A DSR foi escolhida para esse estudo também pelas relevantes contribuições às pesquisas em Sistemas de Informação. A DSR vem se mostrando capaz não só de direcionar a construção do conhecimento, mas também de aprimorar as práticas de desenvolvimento de *software*, gerando novos conhecimentos e soluções tecnológicas para os problemas [Bax 2014].

Para atender às etapas da DSR, a pesquisa foi conduzida da seguinte forma:

1. Pesquisa sistemática da literatura;
2. Enquadramento do problema de pesquisa;
3. Refinamento bibliográfico para delimitação do problema e alternativas de solução;
4. Projeto, desenvolvimento e refinamento do modelo heurístico proposto;
5. Desenvolvimento da ferramenta computacional de apoio – o ReqFRAM;
6. Realização do experimento final de avaliação;
7. Discussão para generalização, explicitação do aprendizado e comunicação dos resultados para conclusões e desenvolvimentos futuros.

Para cada uma dessas etapas da DSR foram escolhidos os métodos de pesquisa mais adequados ao alcance dos objetivos daquele momento. Assim, na etapa de pesquisa sistemática da literatura, utilizou-se a revisão sistemática para garantir as conjecturas teóricas e, aliada à técnica *snowballing*, encontrar os trabalhos correlatos ao tema da pesquisa. Já o método de experimentação foi utilizado na etapa de enquadramento do problema de pesquisa para verificar a possibilidade de descoberta de requisitos e comparação dos resultados obtidos com o método FRAM e a abordagem MacKnight (modelagem BPMN). Após elaboração do modelo heurístico, quinta etapa da pesquisa, a experimentação, aliada à revisão da literatura, também foi aplicada para avaliar o método proposto, permitindo refinamentos. Juntamente com o modelo heurístico, uma ferramenta computacional de apoio (ReqFRAM) foi criada e aplicada nessa avaliação. O método, após refinamento, foi experimentado no caso do sistema de gestão da farmácia canadense para avaliação da possibilidade de ajustes, tal como preconiza as etapas finais da DSR. Essas últimas fases de desenvolvimento e avaliação do artefato permitiram extrair aprendizagens, conclusões e generalizações.

Na sequência, as etapas da DSR citadas anteriormente serão detalhadas. As primeiras foram executadas com base nos conhecimentos adquiridos a partir do processo de pesquisa sistemática da literatura. Ela foi realizada para construir uma base fundamentada de estudos que permitisse identificar e constatar a existência do problema de pesquisa, identificar alternativas de solução e adquirir os conhecimentos necessários para a formulação da solução proposta. Para tal, os seguintes passos foram percorridos:

- Definição da questão da pesquisa: como o modelo FRAM (suas informações, principalmente, a variabilidade) pode ser utilizado na tarefa de elicitação de requisitos para identificar funcionalidades capazes de incorporar ao *software* características que proporcionem um comportamento mais resiliente nos ambientes sociotécnicos complexos;
- Escolha das palavras-chave: FRAM (*Functional Resonance Analysis Method*), *Complex System Modeling*, *Sociotechnical*, *Requirement*, *Elicitation*, *Requirement Engineering*, *Software Engineering*, SAFETY II, *Resilience Engineering*, *Healthcare*;
- Definição das *strings* de busca e suas variações: ((“FRAM” or “*functional resonance analysis method*” or “*complex system modelling*” or “*complex system modeling*” or “*complex system model*”) AND (“*requirement elicitation*” or “*requirement specification*” or “*requirement gathering*”) AND (“*model*” or “*framework*” or “*method*” or “*practice*” or “*approach*” or “*technique*” or “*tool*”) AND (“*process*” or “*heuristics*” or “*guidelines*” or “*rules*” or “*strategy*” or “*way*”) AND (“*Requirement Engineering*” or “*Software Engineering*”) OR (“*Resilience Engineering*” or “*Safety II*”)) AND (“*Health care*” or “*healthcare*” or “*health*”);
- Escolha das bases de busca¹: IEEE, Springer, Google Scholar, LILACS, Pubmed, ACM Digital Library, ISI Web of Science, Science Direct, Scielo, Scopus, SpringerLink, Wiley, WER, Emerald, Medline, Cochrane Library;
- Execução e refinamento das *strings* de busca: inicialmente, 1.164 trabalhos foram identificados
- Armazenamento dos resultados da busca: ferramenta de gestão de referências bibliográficas chamada Zotero²;
- Aplicação dos critérios de inclusão e exclusão:
 - Idioma (somente trabalhos em português e inglês);
 - Área (somente trabalhos da área de saúde ou computação/ tecnologia da informação);
 - Trabalhos publicados na íntegra em bases científicas, incluídos apenas trabalhos sobre a aplicação da modelagem de sistemas complexos para elicitação de requisitos, tendo como foco a Engenharia de Resiliência e a Engenharia de Requisitos;
 - Exclusão de trabalhos de outras áreas temáticas como Psicologia, Educação etc., já que não fazem parte do escopo da pesquisa;

¹ Scopus (www.scopus.com); IEEE Xplore (www.ieeexplore.com.br); ScienceDirect (www.sciencedirect.com); Springer (www.springerlink.com); Web of Science (www.isiknowledge.com); Scielo (www.scielo.org); Google Scholar (www.scholar.google.com); Pubmed (<https://www.nlm.nih.gov/bsd/pubmed.html>); Medline (<https://www.nlm.nih.gov/bsd/medline.html>); Cochrane Library (<https://www.cochranelibrary.com/>); LILACS (<http://lilacs.bvsalud.org/>); Wiley (<https://onlinelibrary.wiley.com/>); WER (<http://wer.inf.puc-rio.br/>); Emerald (<https://www.emerald.com/>); ACM Digital Library (<https://dl.acm.org/>);

² Assistente de pesquisa bibliográfica: Zotero - <https://www.zotero.org/>

- Trabalhos que abordam métodos, heurísticas, diretrizes, *frameworks*, modelos e sistemáticas foram incluídos nos resultados;
- Trabalhos publicados a partir de 1980.
- Análise dos títulos e resumos: foram selecionados 318 trabalhos por atenderem aos critérios de inclusão/exclusão;
- Seleção final: após a leitura dos títulos/resumos, 51 trabalhos estavam efetivamente relacionados ao enfoque da solução.

A abordagem *snowballing* (“bola de neve”) também foi utilizada em caráter complementar à revisão sistemática da literatura. Nessa abordagem, geralmente, o primeiro desafio é identificar um conjunto inicial de artigos para iniciar o procedimento [Wohlin 2014]. Nessa pesquisa, a técnica foi aplicada no conjunto inicial dos 51 trabalhos selecionados na etapa anterior (revisão sistemática tradicional). Assim, novas buscas complementaram o levantamento bibliográfico com refinamentos sucessivos através da técnica de *snowballing* (“bola de neve”). Ao término das interações, foi possível incluir mais 9 trabalhos, totalizando 60 estudos selecionados para análise integral. Portanto, foi com base nesse conjunto final de 60 trabalhos que se formulou a hipótese da pesquisa: o modelo heurístico, considerando o modelo FRAM e suas variabilidades, direciona a tarefa de elicitação de requisitos para identificar requisitos funcionais e não funcionais que incorporam características para um desempenho considerado mais resiliente do *software* em sistemas sociotécnicos complexos. A partir desses estudos, realizou-se uma seleção de trabalhos correlatos que será apresentada mais adiante.

Com a conclusão das três primeiras etapas da DSR, avançou-se para as demais fases que buscam propor a solução (projeto do modelo heurístico). Essa proposição foi conduzida em duas etapas:

1. Verificação da possibilidade de solução através da realização de experimentação, que buscou contribuir com a formulação de um modelo de solução;
2. Refinamento da bibliografia para sustentar conceitualmente as “descobertas” obtidas com a experimentação.

Uma experimentação foi aplicada para atender aos itens acima, permitindo avaliar a aplicabilidade da questão da pesquisa. Os experimentos verificaram a possibilidade de identificação de requisitos para um determinado processo. Para tal, foi escolhido o contexto da Saúde, um sistema sociotécnico complexo. O objetivo principal era avaliar a possibilidade de identificar requisitos de *software* a partir da aplicação de um método orientado ao modelo BPMN (experimentação 1) em comparação ao modelo FRAM (experimentação 2) [De Carvalho *et al.* 2021]. O experimento com o modelo FRAM se mostrou satisfatório para a tarefa de elicitação e foi bem avaliado pelos participantes quando aplicado a um contexto complexo. Esses resultados [De Carvalho *et al.* 2021] foram importantes para justificar a continuidade da pesquisa na busca de um modelo para sistematizar a tarefa de elicitação de requisitos a partir da modelagem FRAM, considerando todas as informações do UdI juntamente com os conceitos da Engenharia de Resiliência no âmbito dos sistemas complexos.

Na sequência, foi elaborado o modelo de solução, chamado modelo heurístico, que se justificasse pela necessidade de sistematizar as diretrizes fundamentais da solução prática concebida para o contexto abordado. Assim, a ideia do modelo heurístico era

formalizar e organizar adequadamente as orientações práticas para extração de requisitos a partir do modelo FRAM.

Nas etapas posteriores, buscou-se definir os requisitos básicos necessários à implementação de um apoio tecnológico para o modelo heurístico proposto. Por fim, realizou-se a avaliação desse modelo com o objetivo de identificar contribuições e possibilidades de melhoria. Os resultados e reflexões obtidos nesta última etapa serviram de base para a divulgação da pesquisa, contribuindo para a ampliação do conhecimento. As atividades realizadas e os resultados alcançados nessas etapas serão detalhados nas próximas seções.

5. Trabalhos Relacionados

Nas primeiras etapas da DSR empregou-se a revisão sistemática que, aliada à técnica *snowballing*, permitiu a seleção de 60 trabalhos correlatos ao tema da pesquisa. Esses estudos foram analisados e apenas 7 se destacaram por apresentar um enfoque da solução mais próximo à abordagem pretendida para essa pesquisa. Nos 7 trabalhos selecionados, observa-se que a Engenharia de Resiliência não se mostra explicitamente sistematizada, apesar de todos eles abordarem conceitos afetos à área e reconhecerem a relação cada vez mais estreita desta com o *design* de dispositivos tecnológicos. Isso se deve principalmente ao fato de a Engenharia de Resiliência não ter como foco o desenvolvimento de sistemas, apesar de poder contribuir com informações sobre o UDI (se considerarmos o “trabalho como feito”, por exemplo). Essa é uma importante constatação, pois justifica como o método proposto pode agregar uma perspectiva inovadora ao quadro conceitual analisado.

Em geral, as abordagens clássicas de especificação de sistemas de informação não se concentram em aspectos relevantes das relações sociotécnicas e podem produzir resultados indesejados quando se trata de sistemas complexos, pois adotam formas simplificadas de representar a realidade e não consideram interações funcionais entre equipamentos, procedimentos e recursos humanos. Em consequência, os dispositivos tecnológicos tornam-se insuficientes na capacidade de responder às situações de maneira resiliente [Hollnagel *et al.* 2006]. Os trabalhos correlatos à pesquisa abordam sob algum aspecto a aplicação de abordagens sociotécnicas no *design* de *software*, a saber:

- Abordagem KARE (*Knowledge Acquisition and Sharing for Requirements Engineering*) busca apoiar o processo da Engenharia de Requisitos, mas os conceitos da Engenharia de Resiliência não são empregados. Além disso, o conceito de sistema complexo não fica claro no trabalho, confundindo-se muitas vezes com as características de complexidade do projeto, dos requisitos e dos próprios envolvidos [Ratchev *et al.* 2003];
- Metodologia *State Analysis* para capturar requisitos de sistema na forma de modelos explícitos. Nessa proposta não há aplicação clara dos conceitos da Engenharia de Resiliência e nem abordagem relacionada à influência dos fatores sociotécnicos [Ingham *et al.* 2006];
- Metodologia UHRAF (*Unified HCI Requirements Analysis Framework*) para tratamento dos requisitos aplicados à Interface Humano-Computador, baseada em cenários e SUM (*Safety and Usability Model*) para inspeção do modelo gerado pelo *framework*. Apesar de lidar com sistemas sociotécnicos complexos, o

trabalho não tem como objetivo organizar e aplicar na prática os conceitos da Engenharia de Resiliência, tendo como foco requisitos específicos para a Interface Humano-Computador [Tung *et al.* 2009];

- O modelo que adota uma perspectiva holística para a elicitação de requisitos, associando o termo complexo à natureza da interrelação entre pessoas, sistemas e ambiente. Consta-se que o trabalho não utiliza modelagem e nem os conceitos preconizados pela Engenharia de Resiliência sobre os sistemas complexos, focando apenas em situações complexas [Katina *et al.* 2014];
- Abordagem sociotécnica para sistemas de informação com o relato do caso do sistema de regulação médica. Não há sistematização da aplicação dos conceitos relacionados à Engenharia de Resiliência em associação ao tema da elicitação de requisitos. A modelagem também não é uma técnica empregada no enfoque da solução [C Vargens 2004];
- A abordagem da criticidade do processo da Engenharia de Requisitos para sistemas sociotécnicos, propondo abordagem holística que engloba não apenas os aspectos funcionais e não funcionais, mas também os requisitos humanos. Nessa proposta não estão sistematizados os conceitos advindos da Engenharia de Resiliência para sistemas complexos e não são aplicados métodos de modelagem [Gregoriades *et al.* 2015];
- O método *Knowledge Acquisition in automated Specification* (KAOS) aplicado à obtenção dos objetivos para a identificação dos requisitos na fase da elicitação. O conceito associado ao termo sistema complexo adotado nesse trabalho não é o mesmo dessa pesquisa, proveniente da Engenharia de Resiliência. Além disso, os autores não sistematizam a aplicação dos conceitos da Engenharia de Resiliência, apesar de terem adotado a modelagem como um recurso para tratar os objetivos [Woldeamlak *et al.* 2016].

Em todos esses trabalhos os conceitos de variabilidade e sistema complexo não são aplicados sob a perspectiva da Engenharia de Resiliência. Além disso, constata-se que há uma lacuna importante a ser explorada no que se refere à sistematização dos conceitos da Engenharia de Resiliência para dar contribuições à ER, principalmente no contexto dos sistemas sociotécnicos complexos. Esses trabalhos trazem soluções do ponto de vista do *software*, mas não utilizam a modelagem de sistemas complexos e nem extraem da variabilidade informações sobre o UdI para “injetar” características sociotécnicas na tarefa de elicitação de requisitos, contribuindo com uma performance mais resiliente dos sistemas complexos.

6. O Modelo Heurístico Proposto

O modelo proposto como solução é baseado em heurísticas, que são empregadas como diretrizes para obtenção de uma solução satisfatória para a questão da descoberta de requisitos a partir dos modelos FRAM. Não há pretensão de tornar o conjunto de heurísticas proposto um guia completo e único para a tarefa de elicitação, mas sim contribuir para a sistematização dessa tarefa tendo o modelo FRAM como ponto de partida, fornecendo um direcionamento mais claro aos profissionais atuantes na interseção desses dois campos, Engenharia de Resiliência e de Requisitos.

Portanto, a solução proposta é um modelo baseado em heurísticas (diretrizes) orientadas ao modelo FRAM para guiar a tarefa de elicitação de requisitos em sistemas sociotécnicos complexos, a fim de prever comportamentos e funcionalidades com características mais resilientes. Para tal, pretende-se unir conceitos, técnicas e ferramentas da Engenharia de Resiliência e de Requisitos, permitindo a construção de uma proposta interdisciplinar. Os conceitos e abordagens utilizados nas heurísticas foram reunidos e organizados numa lógica específica, como mostra na Figura 2.



Figura 2. Conceitos e abordagens do modelo proposto

Como apresentado acima, o modelo proposto tem suas diretrizes baseadas em conceitos agrupados em três grandes eixos:

- Identificação das ações/operações do *software*: grupo de diretrizes que auxiliam na descoberta das operações do sistema de informação quando aplicadas nas Funções do modelo FRAM. Essas heurísticas têm como base conceitual, principalmente, o terceiro princípio³ da Teoria da Atividade⁴ e sua aplicação na tarefa de elicitação de requisitos. Assim, as atividades realizadas pelos indivíduos e representadas como funções no modelo FRAM podem ser desdobradas em ações e operações para o futuro *software*;
- Extração das informações do UdI a partir do modelo FRAM: grupo de heurísticas responsáveis pela captação dos símbolos do UdI a partir das informações contidas no modelo FRAM, ou seja, nome da função, descrição e seus seis aspectos. Com

³ Princípio da estrutura hierárquica da atividade: responsável pela diferenciação dos procedimentos humanos em vários níveis (atividade, ação e operação) [Nardi 1996]. Um resultado é obtido pela transformação de um objeto mediado por uma ferramenta pela ação de um sujeito, mas através de um processo com várias fases ou etapas. Uma atividade é produto de um processo evolutivo de ações e operações. Assim, uma atividade é formada por ações, que por sua vez, também podem ser decompostas em operações [Nardi 1996].

⁴ A Teoria da Atividade, se entendida de forma mais ampla, é o quadro filosófico e interdisciplinar que estuda o processo de desenvolvimento das diversas práticas humanas nos níveis social e individual [Nardi 1996].

as informações em linguagem natural, o Léxico Ampliado da Linguagem⁵ é utilizado nas heurísticas para detectar os sujeitos, verbos, objetos e estados a partir das informações consideradas nativas do modelo FRAM;

- Interpretação das variabilidades: grupo de heurísticas que trabalham especificamente as características referentes a não linearidade e emergência (do ato de emergir/ surgir), próprias de sistemas complexos, captadas pela variabilidade do modelo FRAM. Essas heurísticas adotam preceitos da *Resilience Analysis Grid*⁶ (RAG) e da abordagem MacKnight [Knight *et al.* 2005] para desdobrar a variabilidade (endógena, exógena e de ressonância funcional) em necessidades tecnológicas não funcionais, capazes de gerar uma performance considerada mais resiliente para os sistemas complexos.

Os conceitos acima são utilizados no modelo heurístico segundo uma lógica organizada em quatro etapas, conforme esquematizado na Figura 3. Para aplicação do método é necessário observar algumas premissas responsáveis pela checagem do modelo FRAM, ou seja, pela verificação das informações e elementos mínimos necessários à aplicação das diretrizes nas demais etapas. Essas premissas envolvem a existência de determinados aspectos da função FRAM, completude das informações e padronização da escrita. Algumas regras de sintaxe também são avaliadas, segundo os conceitos enunciados pelos autores de referência no método FRAM [Hollnagel 2004; Hollnagel 2012; Hollnagel *et al.* 2014]. Cabe esclarecer que, apesar de existirem outras avaliações possíveis, as diretrizes contemplam apenas as verificações consideradas relevantes para o método proposto.

Observadas as 13 premissas, executa-se a primeira etapa que é responsável pela pré-avaliação do modelo FRAM, “entrada” do modelo proposto. Por isso, essa etapa é denominada Pré-Processamento. Ela possui as principais regras e boas práticas de modelagem FRAM, segundo as referências conceituais sobre o tema [Hollnagel 2012; Hollnagel 2004; Hollnagel *et al.* 2014]. As heurísticas dessa primeira etapa são doze e estão enunciadas da seguinte forma:

⁵ [Leite 1992] propõe a representação dos requisitos em listas utilizando a linguagem natural através do Léxico Ampliado da Linguagem (LAL). O objetivo é realizar o registro da linguagem utilizada pelos atores do UdI, chamada de símbolos, representados por palavras ou frases [Felicissimo *et al.* 2004]. Cada símbolo do LAL é definido por uma noção e um impacto, que descrevem, respectivamente: I) seu significado e suas relações com outros símbolos; II) efeitos do seu uso e da sua ocorrência no UdI ou efeitos das ocorrências de outros símbolos sobre ele. Assim, chega-se aos quatro elementos do LAL: sujeito, verbo, objeto e estado [Cysneiros and Leite 2001].

⁶ Considera-se um sistema com funcionamento resiliente quando é capaz de sustentar suas operações sob condições esperadas e inesperadas, ajustando seu funcionamento antes, durante ou após eventos (mudanças, perturbações e oportunidades) [Hollnagel *et al.* 2015]. A RAG é uma matriz de perguntas associadas a cada uma das quatro habilidades <RMAA> (Resposta, Monitoração, Aprendizado e Antecipação) para análise do potencial de um sistema para ter um desempenho resiliente, isto é, sua capacidade de <RMAA> [Hollnagel 2011]. Com a RAG é possível detectar as lacunas ou vulnerabilidades do sistema, determinando o que pode ser necessário desenvolver para habilitá-lo em sua capacidade <RMAA>. A RAG é uma base a partir da qual conjuntos mais específicos de perguntas podem ser desenvolvidos e reformulados, sendo mais relevantes para o domínio do sistema [Hollnagel 2015].

- HPP01: Toda Função⁷ deve representar uma ação que produz um resultado, descrevendo algo que uma pessoa faz para executar uma ação específica individual ou coletiva; uma ação isolada de um sistema tecnológico (um sistema automatizado); ou uma ação de um sistema tecnológico em conjunto com uma ou mais pessoas (uma função sociotécnica);
- HPP02: Toda Função deve ser descrita de forma clara e objetiva por um verbo ou locução verbal na forma infinitiva, em vez de estados;
- HPP03: Toda Função deve ter pelo menos uma Entrada;
- HPP04: Todos os Aspectos devem ser descritos com um nome (substantivo) ou uma frase (expressão) substantiva, indicando sempre um estado ou resultado de algo e nunca uma atividade;
- HPP05: Toda Entrada deve indicar algo que ativa, inicia, é usado ou consumido ou transformado por uma Função. Ex: material, energia, informação;
- HPP06: Toda Saída descreve o resultado de uma Função ou o resultado do processamento de uma Entrada. Ex: material, energia, informação;
- HPP07: Toda Pré-condição, quando existir, deve representar um estado que deve ser verdadeiro ou uma condição que deve ser verificada antes que uma Função seja realizada;

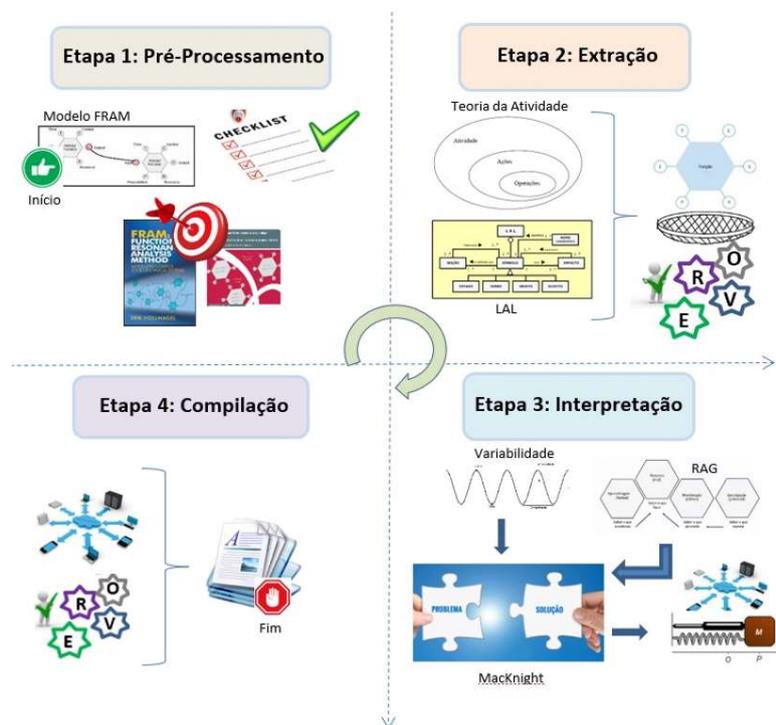


Figura 3. Etapas do método proposto

⁷ Sempre que esse trabalho se referir aos Aspectos do modelo FRAM (Entrada, Saída, Pré-condição, Controle, Recurso e Tempo) ou à Função será usada letra inicial maiúscula para nomear esses elementos. Assim, será possível diferenciá-los de outras possíveis conotações atribuídas aos mesmos termos.

- HPP08: Todo Recurso, quando existir, deve representar algo necessário ou consumido enquanto a Função é executada. Ex: material, energia, informação, competência, software, ferramentas, mão de obra etc.;
- HPP09: Todo Controle, quando existir, deve representar aquilo que supervisiona ou regula a função para que ela produza a Saída desejada. Ex: plano, cronograma, procedimento, conjunto de guidelines/ diretrizes, instruções, algoritmo ou programa de computador, controle social, expectativas pessoais etc.;
- HPP10: O Aspecto Tempo, quando existir, deve representar as várias formas da dimensão tempo impactar a realização da Função (restrição ou recurso);
- HPP11: Toda descrição de uma Função deve ser detalhada em texto livre e conter a informação sobre quem executa a Função, considerando o papel do indivíduo na organização e as suas atividades;
- HPP12: Toda Saída deve ser um Aspecto não-saída de outra Função (Entrada, Pré-condição, Recurso, Controle, Tempo).

Todas essas regras traduzem uma espécie de *checklist* para atribuição de um valor correspondente ao grau de adequação (atende, atende parcialmente ou não atende). Assim, pode-se calcular ao final um percentual geral de adequação da função FRAM nessa primeira etapa. Considerando o grau atende com valor igual a 1, atende parcialmente sendo 0,5 e não atende correspondente a 0, é possível calcular o percentual de adequação com a seguinte fórmula:

$$\%A = \frac{\sum_{1}^{12} HPP \times 100}{12}, \text{ onde \%A é o percentual de adequação da Função FRAM avaliada.}$$

O resultado desse cálculo é um valor percentual entre 0 e 100% que mostra quão adequada está a Função FRAM perante as heurísticas propostas. O somatório de “HPP” corresponde à soma dos valores da adequação (0 ou 0,5 ou 1) atribuídos a cada uma das diretrizes avaliadas individualmente. Ao final, recomenda-se um grau de adequação total para a Função FRAM igual ou maior a 70%. Esse critério foi escolhido por se mostrar suficiente dentro dos parâmetros mínimos necessários sem prejuízo da aplicação das demais heurísticas propostas no modelo. Essa margem de adequação computa a possibilidade de se ter, num primeiro momento, atributos da Função FRAM não preenchidos já que nem sempre a totalidade desses atributos são detectados ou relevantes para a modelagem FRAM.

Cabe ainda esclarecer que, pelo modelo proposto, não há obrigatoriedade de utilizar a totalidade de um modelo FRAM na tarefa de elicitação de requisitos. Assim, fica a critério do profissional de TI definir as instâncias do modelo FRAM que comporão o domínio da aplicação. Por isso, a unidade de “entrada” do método é a Função do modelo FRAM, permitindo que o engenheiro ou analista de requisitos / sistemas defina o conjunto de Funções consideradas como escopo para aplicação do método.

A segunda etapa, chamada Extração, é aplicada idealmente no conjunto de funções do modelo FRAM que atingiram o percentual de adequação mínimo. As 12 heurísticas dessa etapa têm como base principal os conceitos da Teoria da Atividade e as diretrizes de identificação dos símbolos do LAL (sujeito, verbo, objeto e estado) com suas noções e impactos. Elas incluem um processo cognitivo de aplicação que é mostrado na Figura 4.

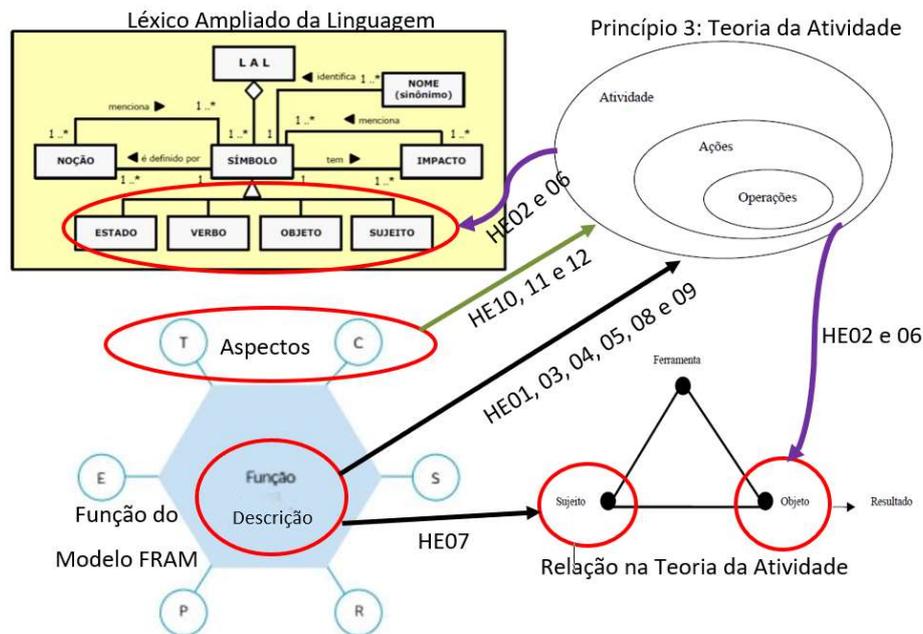


Figura 4. Processo de aplicação das heurísticas de extração

O esquema acima representa como as heurísticas foram concebidas a partir das conjecturas teóricas definidas para a pesquisa, baseadas principalmente no Léxico Ampliado da Linguagem e na Teoria da Atividade. Assim, é possível perceber que há dois conjuntos de heurísticas: um que atua na Função e em sua descrição; outro que atua nos seus Aspectos. Em ambos os conjuntos há diretrizes (HE01, 02, 05, 06, 07 e 10) que tratam da extração de ações/operações do futuro *software*, derivando também seus objetos/entidades (com detalhamento dos dados) e atores (sujeitos/ indivíduos que interagem com essas ações/operações). Já outras heurísticas (HE03, 04, 08, 09, 11 e 12) têm foco nas ações/operações de *software* detectadas, tratando especificamente as mudanças de estados e restrições a que estão sujeitas. Essas últimas podem gerar novas ações/operações (usando o 3º princípio do desdobramento pela Teoria da Atividade) do *software*, devendo-se identificar também seus objetos/entidades e dados manipulados. No caso da mudança de estado, a condição observada para que a alteração do estado aconteça é uma informação importante e deve ser elicitada, bem como os estados inicial e final. O mesmo procedimento é recomendado para o caso das restrições que muitas vezes são identificadas no próprio modelo FRAM, devendo-se enunciar-las sempre em associação às ações/operações de *software* que as geraram. Essas heurísticas têm como base as diretrizes que definem os símbolos do LAL. Cabe destacar que todas as heurísticas estão interrelacionadas e usam os resultados umas das outras para produzir novas informações. Portanto, todos os resultados iniciais produzidos pelas heurísticas são baseados nas informações sobre o UdI e obtidas a partir das Funções FRAM captadas pelo modelador no trabalho de campo. Caso essas informações nativas do modelo não estejam completas

o suficiente para a aplicação das heurísticas, o profissional de TI deve voltar a campo utilizando técnicas tradicionais de levantamento de requisitos para obter as informações complementares sempre direcionado pelo modelo FRAM e pelas instruções das heurísticas.

Na terceira etapa, denominada Interpretação, o foco é a variabilidade, um dos diferenciais do método proposto. As heurísticas dessa etapa aplicam os conceitos da abordagem MacKnight [Knight *et al.* 2005] para identificar as dificuldades (entraves, problemas, imprevistos) descritas na variabilidade de Saída da função FRAM com o objetivo de definir ações de contorno/ mecanismos de amortecimento, isto é, necessidades do futuro *software*. O conjunto de diretrizes e a dinâmica do processo de aplicação dessas heurísticas é mostrado na Figura 5.

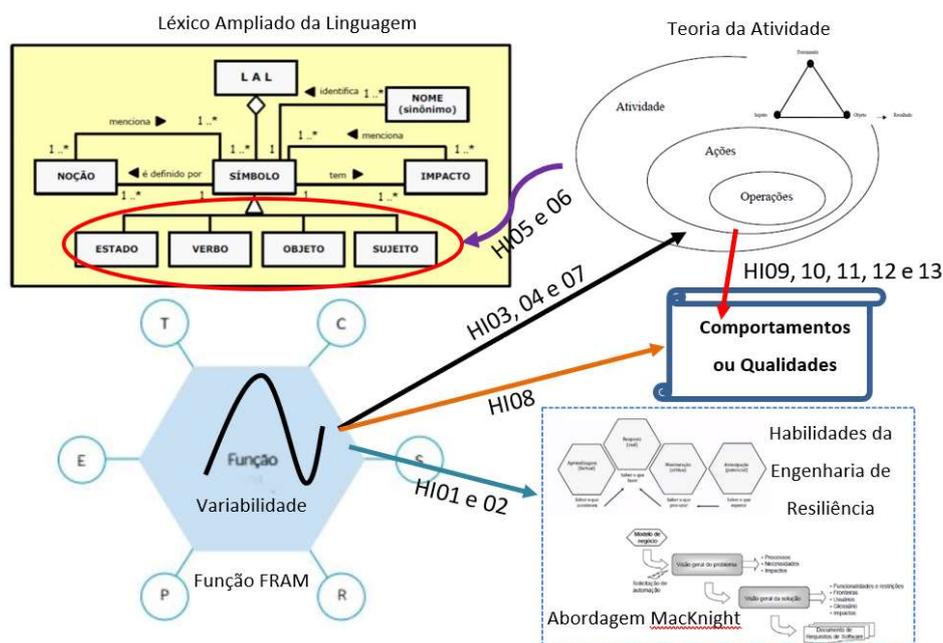


Figura 5. Processo de aplicação das heurísticas de Interpretação

As heurísticas iniciais usam a técnica de derivação das dificuldades a partir da descrição da variabilidade. Considerando a perspectiva da Engenharia da Resiliência, adicionou-se também as oportunidades, pois as variabilidades também podem gerar resultados positivos. Para tratar essas dificuldades e oportunidades, seguindo a abordagem MacKnight, são definidas as necessidades, que são os mecanismos de amortecimento das dificuldades ou de impulsionamento/ incentivo das oportunidades sob o ponto de vista tecnológico. Assim, ações/ operações do *software* se desdobram (pelo 3º princípio da Teoria da Atividade) para apoiar ou automatizar a execução das necessidades. As quatro habilidades da Engenharia da Resiliência são propostas nas heurísticas para tornar mais robusta a escolha dos mecanismos tecnológicos de amortecimento ou incentivo que se traduzem pelas necessidades. Assim, as quatro habilidades avaliadas na RAG são aplicadas às oportunidades e vulnerabilidades para conduzir a escolha das necessidades, priorizando um desempenho mais resiliente.

As heurísticas intermediárias trabalham as ações/operações identificadas para o *software*, definindo seus objetos/entidades e dados manipulados, mudanças de estados e restrições, seguindo a mesma lógica das heurísticas da etapa de Extração. Para tal, foi aplicado o mesmo conjunto de técnicas e conceitos (Teoria da Atividades e LAL,

principalmente). Já as heurísticas finais dessa etapa de Interpretação trabalham a descoberta dos atores envolvidos nas ações/operações do *software*, bem como os comportamentos e qualidades associados a essas ações, ou seja, os requisitos não funcionais. Nesse ponto, espera-se que o profissional de TI explore as características não funcionais que possam estar vinculadas aos aspectos mais gerais da tecnologia (requisitos não funcionais independentes) ou restritos à forma de operar do *software* (requisitos não funcionais dependentes de requisitos funcionais). Nesse momento, torna-se importante observar que, ao definir características não funcionais, deve-se avaliar se há necessidade de criar outras ações/operações para o *software*, devendo-se recomençar a aplicação das heurísticas para esse novo grupamento.

Por fim, a quarta etapa é a Compilação que remete às diretrizes responsáveis pela consolidação sistematizada das informações produzidas nas etapas anteriores. A dinâmica de aplicação das diretrizes é mostrada na Figura 6.

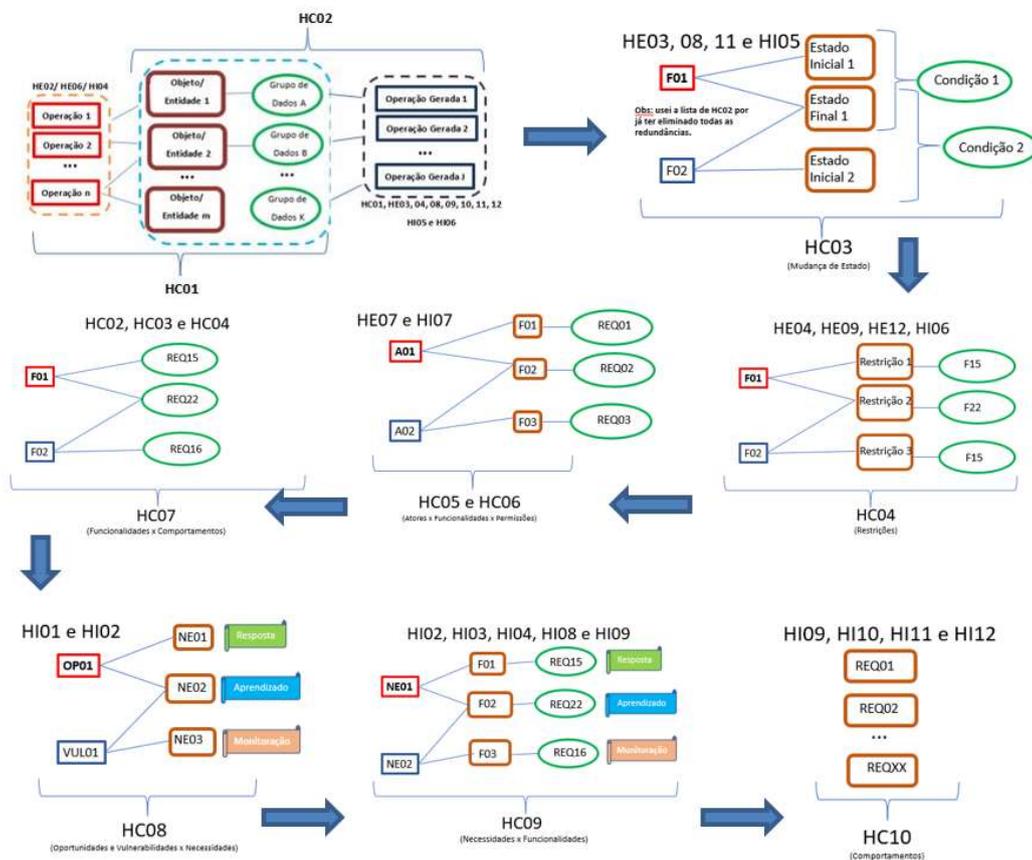


Figura 6. Processo de aplicação das heurísticas de Compilação

As heurísticas dessa quarta etapa orientam a manipulação dos resultados obtidos nas fases anteriores, visando à elaboração de uma lista inicial e consistente de requisitos funcionais e não funcionais. Para elaborar essa “saída” consolidada, buscou-se as referências e boas práticas para escrita de requisitos e elaboração documentos de especificação. Nessa primeira tarefa da ER, a elicitação de requisitos, o que se espera é a identificação inicial de uma lista de desejos, ou seja, requisitos para o futuro *software*. As demais etapas do processo da ER auxiliarão no refinamento dessa lista inicial, podendo gerar novas iterações para a tarefa de elicitação. Assim, nessa última etapa do modelo heurístico, as “saídas” geradas por cada uma das heurísticas propostas foram agrupadas e

organizadas num único documento, denominado “Relatório Geral de Requisitos” (apresentado na próxima seção) para diferenciar de quaisquer outros documentos referenciados na literatura em outras tarefas do processo da ER.

As heurísticas dessa última etapa se relacionam seguindo uma lógica de processamento que utiliza as informações obtidas nas etapas anteriores, conforme mostrado na Figura 6. A ordem de execução proposta visa concatenar as informações produzidas ao longo das etapas de forma organizada e lógica com o objetivo de compor as funcionalidades que são sempre formadas por uma “triáde”: operações, seus objetos/entidades e grupos de dados. Em algumas diretrizes, novas operações são geradas a partir de mudanças de estados e restrições. Assim, nas heurísticas HC01 e 02, o objetivo é criar uma única lista com essas triádes <operações ou operações geradas + objetos/entidades + grupos de dados> sem redundâncias para atribuir uma identificação única. Com isso, tem-se uma lista de requisitos funcionais que será enriquecida de mais detalhes ao longo da aplicação das demais heurísticas.

As heurísticas HC03 e HC04 tratam, respectivamente, da atribuição das mudanças de estados (estados iniciais, finais e suas condições) e restrições a esses requisitos funcionais. Assim, são montadas as relações entre as funcionalidades e suas mudanças de estados, bem como as funcionalidades e as restrições. Dessa forma, os requisitos funcionais são “enriquecidos” com mais informações sobre o UdI, permitindo a geração de requisitos mais robustos nas demais etapas da ER.

Já as diretrizes HC05, 06, 07 e 10 cuidam dos requisitos não funcionais, associando-os às funcionalidades e aos atores, quando pertinentes. Portanto, esses requisitos não funcionais associam-se diretamente às funcionalidades, que, por sua vez, foram geradas a partir da elicitação das informações sobre as Funções e Aspectos do modelo FRAM.

Por fim, as heurísticas HC08 e 09 tratam dos requisitos obtidos a partir da interpretação das informações sobre a variabilidade, uma questão central para o método proposto. Essas diretrizes propõem a relação entre oportunidades e/ou vulnerabilidades com as necessidades e dessas com as funcionalidades e outros aspectos não funcionais. Essas relações já garantem uma iniciativa de rastreabilidade, que é uma característica importante para o processo da Engenharia de Requisitos.

7. A Ferramenta ReqFRAM

A proposição do modelo heurístico gerou a necessidade de desenvolvimento de uma ferramenta computacional para apoiar a aplicação do método. Assim, surgiu o ReqFRAM, um *software* desenvolvido com as tecnologias Python 3.8 para linguagem de programação, desenvolvido no Framework Django 3.0 e banco de dados PostgreSQL 4.2. A ferramenta incorpora em sua lógica de execução as quatro etapas previstas no método (Pré-processamento, Extração, Interpretação e Compilação). O ReqFRAM tem, nessa primeira versão, o tratamento de cinco regras de negócio, oito grupos de requisitos não funcionais (segurança da informação, usabilidade, acessibilidade, manutenibilidade, portabilidade, disponibilidade, confiabilidade e desempenho) e treze grupos de funções (Controle de Acesso, Lista de Análises, Realização do Pré-processamento, Atribuição de Índice de Adequação, CRUDs, Consulta ao Catálogo de RNFs, Eliminação de Redundâncias, Atribuição de IDs e Habilidades, Geração de Relatórios e Exibição de Dicas).

O ReqFRAM tem um fluxo de processamento que inicia com a autenticação usando login e senha para visualização dos modelos FRAM (em versão PDF gerada pela FMV) a serem processados. Após selecionar o conjunto de modelos para análise, o ReqFRAM inicia a primeira etapa, o Pré-processamento, para aplicação automatizada das heurísticas com aplicação de mineração textual e realização de cálculo dos percentuais de adequação. Em cada heurística, a ferramenta realiza a mineração de texto nos atributos nome, descrição e Aspectos da Função FRAM para detectar automaticamente substantivos, verbos ou locuções verbais no infinitivo, verbos no particípio e adjetivos com o objetivo de aplicar as verificações propostas nas diretrizes. Com o resultado da mineração textual e aplicação das regras, a própria ferramenta atribui um valor de adequação (atende, atende parcialmente ou não atende) a cada heurística.

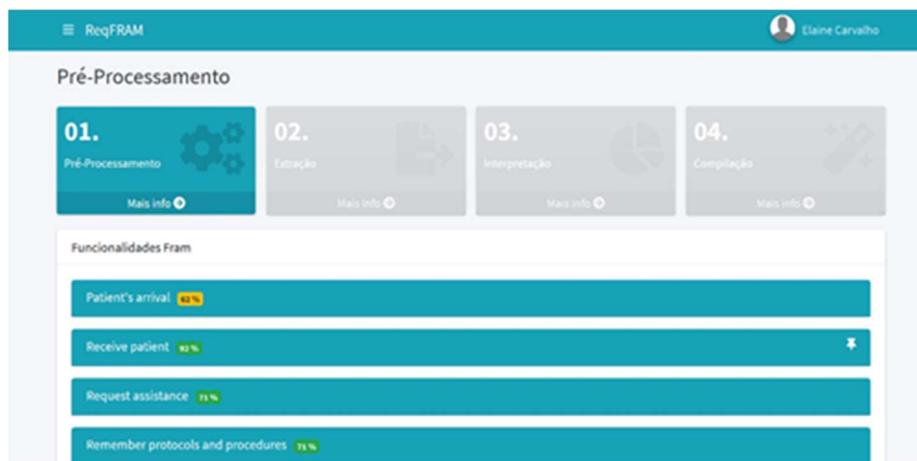


Figura 7. Etapas da aplicação do método

Ao final, são apresentadas todas as heurísticas já avaliadas para que o usuário tenha a possibilidade de alterá-las se achar pertinente. Na tela principal, a lista de Funções FRAM aparece atualizada com o índice de adequação calculado e a cor correspondente (verde, amarelo ou vermelho). Ainda ao lado de cada Função avaliada, aparece um ícone “alfinete” ativo, mostrando que ela está selecionada para avançar para a próxima etapa, a Extração. Nessa fase, a ferramenta destaca os atributos da Função considerados relevantes para aplicação das heurísticas e mostra as dicas de preenchimento dos campos.

Com a conclusão dessa etapa, as Funções FRAM com variabilidade podem avançar para a etapa de Interpretação. Nesse ponto, a ferramenta fornece um campo para preenchimento da descrição completa da variabilidade. Posteriormente, o *software* destaca em cada heurística os atributos da Função e, diretriz após diretriz, os campos são exibidos para preenchimento com possibilidade de consulta às dicas e ao catálogo de requisitos não funcionais. Por fim, na Compilação, o ReqFRAM concatena os campos preenchidos nas etapas anteriores seguindo a lógica definida, elimina as redundâncias e atribui os códigos sequenciais únicos para identificação dos requisitos e demais “saídas” a serem geradas. Todas as informações são automaticamente consolidadas num arquivo Microsoft Excel, conforme mostrado na Figura 8.

Relatório Completo de Requisitos Caso Farmácia			
Autor		Elaine Alves	
Data		23/09/2020	
A) Oportunidades e Vulnerabilidades com suas Necessidades classificadas segundo as habilidades da Engenharia de Resiliência:			
Habilidade: Resposta			
Vulnerabilidade		Necessidade	
ID	Descrição	ID	Descrição
VUL1	Falta de confiança nos mecanismos de privacidade da informação	NE1	Garantir sigil das informações do paciente
VUL2	Falta de acesso às informações do histórico de medicamentos do paciente	NE2	Permitir acesso ágil e fácil aos dados do histórico médico do paciente
B) Lista de Funcionalidades e Requisitos Não Funcionais do Software para as Necessidades de acordo com as habilidades da Engenharia de Resiliência:			
Habilidade: Resposta			
Necessidade		Funcionalidade	
ID	Descrição	ID	Descrição
NE2	Permitir acesso ágil e fácil aos dados do histórico médico do paciente	F12	O sistema deve recuperar do sistema de prontuário eletrônico, através de consulta por CPF do paciente ou CRM do médico ou palavra-chave, todos os dados do histórico médico, medicamentos usados, exames realizados, indicações das últimas prescrições e prescrição mais atual do paciente
Necessidade		Requisito Não Funcional	
ID	Descrição	ID	Descrição
NE1	Garantir sigil das informações do paciente	REQ-7	O sistema deve ter seus procedimentos de segurança e controle de acesso validados e certificados na ISO/IEC 27001
NE2	Permitir acesso ágil e fácil aos dados do histórico médico do paciente	REQ-5	O sistema deve se integrar ao sistema de prontuário eletrônico da província para obter todos os dados relacionados ao histórico médico, últimas prescrições e medicamentos usados pelo paciente
C) Lista de Mudanças de Estado e Funcionalidades:			
Funcionalidades com Mudança de Estado		Funcionalidade Gerada pela Mudança de Estado	
ID	Descrição	ID	Descrição
F8	O sistema deve transformar o atendimento de um paciente de "iniciado" para "em andamento" sempre que o cadastro do atendimento for criado através do CPF do paciente pelo farmacêutico após sua autenticação no sistema	F6	O sistema deve verificar possibilidade de fornecimento, informando CEP da residência do paciente e CEP da província responsável pela prescrição
Funcionalidade Gerada para Atender à Restrição		Restrição	
ID	Descrição	ID	Descrição
F7	O sistema deve cancelar o atendimento do paciente informando CEP residencial, CEP da província responsável pela prescrição e motivo do cancelamento	R1	A farmácia só pode fornecer medicação a pacientes residentes na mesma província da realização da consulta médica.
F13	O sistema deve transformar o atendimento iniciado em cancelado sempre que o CEP residencial do paciente for diferente do CEP da província responsável pela prescrição		
E) Lista de Funcionalidades Mapeadas para os Atores com suas Permissões/ Controles de Acesso:			
ID	Ator	ID	Funcionalidade Mapeada
A1	Farmacêutico	F1	O sistema deve manter (consultar, incluir, alterar e excluir) todas as informações (dados pessoais, exames médicos, histórico médico, medicamentos anteriores, prescrição médica) do paciente
ID	Ator	ID	Controles e Permissões
A1	Farmacêutico	REQ-8	O sistema deve permitir que o farmacêutico tenha acesso completo às operações do cadastro do atendimento do paciente
...
F) Lista de Características Não Funcionais Atribuídas às Funcionalidades do Software:			
ID	Funcionalidade do Software	ID	Característica Não funcional Atribuída
F5	O sistema deve armazenar cópia digital da prescrição médica com associação ao registro do atendimento do paciente.	REQ-3	O sistema deve ser compatível com scanner para digitalização, armazenamento e associação automática das imagens ao CPF do paciente.
...
G) Lista de Funcionalidades Desejadas para o Software:		H) Lista de Requisitos Não Funcionais Desejados para o Software:	
Funcionalidades		Requisitos Não Funcionais	
ID	Descrição	ID	Descrição
F1	O sistema deve manter (consultar, incluir, alterar e excluir) todas as informações (dados pessoais, exames médicos, histórico médico, medicamentos anteriores, prescrição médica) do paciente	REQ-1	O sistema deve ser capaz de controlar os acessos dos usuários mediante perfis de acesso (farmacêutico deve ter perfil próprio com permissão de consulta, edição e inclusão de dados no cadastro dos pacientes).
F2	O sistema deve permitir a autenticação do usuário por login e senha	REQ-2	O sistema deve permitir integração com o sistema de gestão dos laboratórios da província para recuperar todos os dados disponíveis sobre os exames laboratoriais realizados pelo paciente mediante consulta do seu CPF.
...

Figura 8. Parte do relatório final de requisitos gerado pelo ReqFRAM

Esse relatório consolida a lista de requisitos sob diversas possibilidades de análises devido aos “cruzamentos de informações” (requisitos x mudanças de estados, requisitos x permissões/controles de acesso, restrições x requisitos etc.), realizando ainda a segregação entre dois principais tipos de requisitos: funcionais e não funcionais. Além disso, as vulnerabilidades e necessidades são indicadas juntamente com seus respectivos requisitos, segundo as habilidades da RAG (Resposta, Monitoração, Aprendizado e Antecipação). Esse é um diferencial importante dessa ferramenta, já que contribui para a análise da resiliência sob a perspectiva dos requisitos de *software*.

8. Avaliação Experimental

A experimentação foi o método de pesquisa aplicado nessa etapa da DSR para avaliar o modelo heurístico. A experimentação foi utilizada por permitir que determinado fenômeno seja observado em um ambiente controlado e com um cenário de pesquisa definido [De Carvalho *et al.* 2021]. Apesar das limitações identificadas (heterogeneidade dos participantes, limitações metodológicas etc.), o principal objetivo foi a realização da tarefa de elicitação de requisitos aplicada a um contexto sociotécnico complexo, nesse caso, a Saúde (caso da Farmácia canadense traduzido de [Li *et al.* 2019]), utilizando uma abordagem tradicional [Knight *et al.* 2005] em comparação ao modelo heurístico proposto já usando a ferramenta ReqFRAM.

As experimentações foram realizadas por dois grupos distintos formados cada um por 3 (três) participantes. Todos os seis participantes tinham o perfil de analista de requisitos e experiência mínima de 5 (cinco) anos na área de TI. Os experimentos aconteceram em condições idênticas de infraestrutura com um intervalo temporal de quinze dias entre uma experimentação e outra. Ambos os experimentos tiveram a mesma duração (2,5 horas) e usaram a descrição tabular do mesmo cenário da farmácia canadense, que descreve apenas a dinâmica de fornecimento dos medicamentos da rede pública à população canadense, sendo parte de um contexto maior [Li *et al.* 2019]. O cenário usado nos experimentos era formado por seis Funções FRAM com grande necessidade de integração da informação, pois o contexto complexo retratado evidencia a necessidade de comunicação entre os farmacêuticos das províncias e desses com os médicos responsáveis pela prescrição dos medicamentos.

As hipóteses definidas para os experimentos estavam associadas à capacidade de gerar (experimento com o modelo heurístico e a ferramenta ReqFRAM) ou não (experimento com a abordagem MacKnight) uma lista de requisitos funcionais e não funcionais mais robusta (com características consideradas resilientes [Jaramillo and Palacios 2014]), ou seja, requisitos que incorporam aspectos associados às quatro habilidades (RMAA) da Engenharia de Resiliência). As variáveis controladas nesses experimentos são as Funções FRAM, que compõem o modelo usado como cenário de aplicação, e os métodos aplicados para realizar a tarefa de elicitação (MacKnight e o modelo heurístico proposto).

O documento usado como “entrada” por cada participante nos experimentos tinha um formato descritivo tabular, gerado pela própria ferramenta FMV, contendo as Funções FRAM (nome, descrição e seus aspectos) correspondentes ao caso da Farmácia canadense. No caso do experimento usando o modelo heurístico e a ferramenta ReqFRAM, uma informação adicional foi utilizada: a descrição da variabilidade de cada Função (quando existente), conforme disponibilizado pelo modelador nas suas notas de campo. Ao final dos experimentos, cada participante entregou sua lista de requisitos. No caso do segundo experimento, a lista foi gerada automaticamente pela ferramenta ReqFRAM (relatório final de requisitos apresentado na seção anterior). Essa lista de requisitos gerada é considerada a variável mensurada ou dependente para os experimentos. Para a comparação dos resultados obtidos em ambos os experimentos, as listas de requisitos geradas foram avaliadas e as informações inconsistentes e/ou redundantes foram descartadas para fins de análise comparativa.

O cenário da Farmácia canadense usado nos experimentos descreve um processo que inicia quando um paciente chega à farmácia, após consulta com o médico, para obter a medicação necessária. A primeira função executada pelo farmacêutico é a obtenção de informações do paciente para identificar sua necessidade. Logo em seguida, o farmacêutico complementa essas informações iniciais com outras obtidas a partir do PMS (PMS – *Pharmacy Management System*) que é o Sistema de Gestão da Farmácia. Com todas essas informações reunidas, o farmacêutico tem condições de avaliar a adequação da prescrição realizada para o paciente, analisando inclusive as possíveis interações medicamentosas. Após essa avaliação, acontece o fornecimento da medicação ao paciente com devido registro desse ato no Sistema de Gestão da Farmácia. Se, na tarefa de avaliar a prescrição, o farmacêutico detectar a necessidade de alterá-la significativamente, todas as mudanças efetuadas devem ser enviadas ao médico. Nesse cenário, no momento de obter informações do paciente, o farmacêutico precisa lidar com

barreiras linguísticas, desconfiança do paciente quanto à privacidade das informações, equívocos na prescrição, falta de indicação e ausência de outras informações sobre a prescrição e o paciente. Alguns mecanismos de amortecimento são adotados pelos atores para contornar os efeitos dessas perturbações endógenas e exógenas, mas há carência de apoio tecnológico. Portanto, garantir mecanismos de mitigação dessas perturbações torna-se importante para reduzir atrasos no atendimento, reduzir desconfianças ou evitar o fornecimento de medicação equivocada. Ter requisitos de *software* que potencializem esses mecanismos se torna uma estratégia importante para capacitar esse sistema complexo (Farmácia) a ter uma performance considerada mais resiliente.

Ao final dos experimentos, após realizar a comparação qualiquantitativa dos resultados, foi possível concluir que, no caso do sistema complexo representado pelo método FRAM, o modelo heurístico proporcionou resultados mais satisfatórios para a tarefa de eliciação de requisitos em relação à abordagem MacKnight. Isso se justifica pela detecção de funcionalidades e características não funcionais em maior volume e tratamento mais direcionado das variabilidades, foco da pesquisa. No experimento com a abordagem MacKnight, ainda sem considerar a variabilidade, muitos requisitos associados às mudanças de estado, restrições e até mesmo requisitos não funcionais não foram detectados (elicitados), ainda que a informação de “entrada” fosse a mesma, o modelo FRAM. Contudo, no experimento com o modelo heurístico, esses mesmos requisitos apareceram de forma sistemática nas listas geradas por todos os participantes. Um exemplo desse caso são os requisitos “RF: O sistema deve verificar possibilidade de fornecimento, informando CEP da residência do paciente e CEP da província responsável pela prescrição” e “RF: O sistema deve cancelar o atendimento do paciente informando CEP residencial, CEP da província responsável pela prescrição e motivo do cancelamento”. Eles não tinham relação com a variabilidade e foram elicitados a partir de heurísticas do modelo proposto, mas não foram detectados com a aplicação do método tradicional (MacKnight) por nenhum dos participantes. Os quadros teóricos utilizados na construção do modelo heurístico proposto explicam essa diferença. Eles amplificam o potencial de eliciação de requisitos ao explorarem de forma complementar e sistemática as informações do UdI existentes no modelo FRAM.

Assim, chega-se ao final dos experimentos com a confirmação das hipóteses experimentais. No primeiro experimento confirmou-se a possibilidade de elicitar requisitos com aplicação do método tradicional orientado ao modelo FRAM, mas não foi possível detectar requisitos específicos para amortecimento das perturbações causadas pelas variabilidades das Funções FRAM. Por consequência, também não foi possível estabelecer uma relação desses requisitos com as habilidades da Engenharia de Resiliência. O segundo experimento, no entanto, conseguiu comprovar que é possível elicitar requisitos a partir das heurísticas propostas, com geração de requisitos específicos para satisfazer às necessidades de apoio tecnológico que visam ao amortecimento ou mitigação das variabilidades. A partir das heurísticas do modelo proposto também foi possível identificar com qual habilidade da Engenharia de Resiliência determinado requisito contribui ao amortecer os efeitos de certa variabilidade. Essa é uma correlação importante, já que favorece a análise do sistema complexo, capacitando-o a ter uma performance mais resiliente.

Portanto, o modelo heurístico proposto proporciona a descoberta de requisitos com maior potencial de reforço às características positivas do sistema complexo. Isso ocorre pela aplicação do quadro teórico referente à análise da RAG, presente nas

heurísticas da etapa de Interpretação. Ela possibilitou a identificação de oportunidades e, conseqüentemente, necessidades capazes de impulsionar habilidades (resposta, monitoração, aprendizado, antecipação) do sistema complexo mediante o apoio de requisitos tecnológicos a serem desenvolvidos para o futuro *software*. No primeiro experimento, com o método tradicional, essa perspectiva analítica não aconteceu por limitações do próprio método, que não foi criado para tal. Com o modelo heurístico, no segundo experimento, a ampliação da perspectiva de análise do UdI para a elicitación ocorreu devido às contribuições teóricas da Engenharia de Resiliência.

Para exemplificar isso, pode-se comentar o resultado obtido na Função FRAM 4 – Fornecer medicação. As variabilidades das funções à montante (antes) da Função FRAM 4 (Fornecer medicação) impactam no sistema complexo avaliado e criam necessidades de adaptação tanto para atenuar as perturbações provocadas quanto para potencializar a performance do sistema. Assim, na Função 4 de fornecimento da medicação há tanto a perturbação exógena quanto a endógena. A exógena acontece quando o medicamento prescrito não atende às condições clínicas do paciente, provenientes da avaliação realizada na Função 3 – Avaliar adequação da prescrição. Já a perturbação endógena ocorre quando a medicação prescrita é considerada adequada (também resultante da avaliação da Função 3), mas não está disponível na farmácia. Com isso, o mecanismo de amortecimento utilizado pelo farmacêutico é a busca de uma medicação alternativa. No entanto, ao procurar por uma substituição, o farmacêutico precisa lidar constantemente com *trade-off* entre o tempo e a precisão, já que nem sempre é possível optar pelo rigor em vez da eficiência, uma vez que o paciente deve sair da farmácia com um medicamento sem esperar demais, pois há outros pacientes também aguardando atendimento.

Com base nesse contexto, os participantes do segundo experimento encontraram oportunidades e vulnerabilidades que geraram necessidades e, em última instância, requisitos funcionais e não funcionais aptos a suportar tecnologicamente esse sistema complexo, habilitando-o a ter uma performance mais resiliente de acordo com as quatro habilidades da Engenharia de Resiliência. Assim, a oportunidade “OP01: Disponibilização das bulas digitais dos medicamentos pelos fornecedores” e a vulnerabilidade “VUL03: Falta de informação sobre disponibilidade de medicamentos em estoque” foram associadas, respectivamente, às necessidades “NE03: Aplicar *text mining* nas bulas para filtrar os medicamentos-candidatos” (habilidade: Aprendizado) e “NE04: Acompanhar informações do sistema de estoque sobre medicamentos disponíveis” (habilidade: Monitoração). A primeira necessidade aponta para a habilidade de aprendizado, já que sugere o uso do recurso de mineração textual por palavras-chave correspondentes ao quadro clínico do paciente (fumante, grávida, diabético etc.) nas bulas dos medicamentos. O objetivo é a eliminação dos remédios não aplicáveis ao quadro clínico avaliado, criando uma base de conhecimento para os perfis clínicos pesquisados a partir das minerações realizadas. A outra necessidade está relacionada à habilidade de monitoração, já que implementa mecanismos para observação constante do nível de estoque dos medicamentos a partir da integração com outro sistema (Sistema de Gestão de Estoque), visando à detecção dos remédios indisponíveis. Percebe-se ainda que essa capacidade de monitoração induz à criação de outras necessidades e requisitos relacionados à habilidade de antecipação, pois, a partir da informação sobre a indisponibilidade dos medicamentos, o futuro *software* (pelos requisitos elicitados) pode indicar antecipadamente os remédios substitutos a partir da disponibilidade em estoque

versus perfil clínico aprendido. Com isso, esse sistema sociotécnico complexo representado pelo modelo FRAM poderá alcançar uma capacidade de desempenho mais resiliente.

9. Conclusão

O presente trabalho, como uma versão estendida do artigo SBSI 2021 [Alves Carvalho *et al.* 2021], apresentou com mais detalhes os passos da DSR para a construção de uma abordagem para a tarefa de elicitação de requisitos aplicada a sistemas sociotécnicos complexos. O modelo heurístico, projetado à luz das conjecturas teóricas formuladas a partir da revisão sistemática da literatura (etapas da DSR), respondeu à questão da pesquisa “como elicitar requisitos de *software* para sistemas complexos a partir de modelos FRAM?”. Ou seja, as heurísticas sistematizadas em quatro etapas e fundamentadas nos conceitos da Engenharia de Resiliência e de Requisitos, tendo como elemento principal o modelo FRAM e suas variabilidades, possibilitam a elicitação de requisitos de *software* para sistemas complexos. Esse modelo heurístico, base de criação da ferramenta computacional ReqFRAM, foi objeto de experimentação. Após a realização dos experimentos, concluiu-se que o modelo proposto apresentou resultados considerados satisfatórios ao gerar uma lista de requisitos funcionais e não funcionais com características que atendem, em algum grau, as habilidades (Resposta, Monitoração, Aprendizado e Antecipação) da RAG. Portanto, mostrou-se que a tarefa de elicitação para sistemas complexos consegue ser realizada a partir do modelo FRAM com a aplicação das heurísticas propostas. Esses resultados também foram importantes para validar as conjecturas teóricas da pesquisa, conforme preconizado pela DSR.

A solução proposta nesse trabalho, apesar das contribuições, deve passar por avaliações adicionais futuras, visando ampliar o entendimento do seu potencial e gerar novas oportunidades de melhoria. Outros estudos alinhados à questão da variabilidade também devem ser conduzidos para aprimorar a abordagem proposta. A variabilidade é um tema central para a pesquisa e ainda há muito a ser explorado no campo da Engenharia de Resiliência, pois há indefinições e interpretações conflitantes na literatura sobre alguns aspectos que impactam a forma de descrever, representar e interpretar as manifestações da variabilidade no trabalho de campo. Como esse elemento é crucial para a aplicação do método, torna-se fundamental avaliar as boas práticas e recomendações que forem surgindo para padronizar a formalização da variabilidade. Outras pesquisas também devem ser realizadas futuramente para avaliar a adequação dos requisitos gerados pelo método proposto às características normativas exigidas para os sistemas complexos, bem como estudos para incorporar a tarefa de análise de requisitos nesse método e estratégias de aprimoramento da variabilidade como fonte de requisitos não funcionais.

Agradecimentos

Agradecemos aos órgãos de fomento à pesquisa, parceiros da UFRJ, CAPES, CNPQ, CEE/FIOCRUZ, PPGI/UFRJ e todo corpo docente pela realização dessa pesquisa.

Referências

Alves Carvalho, E., Orlando Gomes, J., Jatobá, A., Ferreira Silva, M., & Rodrigues Carvalho, P. V. (2021). Software Requirements Elicitation for Complex Systems with the Functional Resonance Analysis Method (FRAM). In *XVII Brazilian Symposium on Information Systems*, pp. 1-8.

- Bax, M. P. (2014). Design Science: filosofia da pesquisa em ciência da informação e tecnologia. In: *XV Encontro Nacional de Pesquisa em Ciência da Informação*. Belo Horizonte, 3883-3903.
- Baxter, G. and Sommerville, I. (2011). “Socio-technical systems: From design methods to systems engineering”. *Interacting with computers*, 23 (1), 4-17.
- Bitencourt, Aryslene. S., Paiva Débora Maria, B. and Cagnin Maria Istela. (2016). Elicitação de Requisitos a partir de Modelos de Processos de Negócio em BPMN: Uma Revisão Sistemática. In: *SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI)*, 12, Florianópolis. Anais do XII Simpósio Brasileiro de Sistemas de Informação. Porto Alegre: Sociedade Brasileira de Computação, 200 – 207.
- Chakrabarti, A. (2010). “A course for teaching design research methodology”. *AI EDAM*, 24 (3), 317-334. <http://dx.doi.org/10.1017/S0890060410000223>. [GS Search]
- Christel, M. G. and Kang, K. C. (1992). “Issues in requirements elicitation”. Carnegie-Mellon Univ Pittsburgh Software Engineering Inst.
- Clay-Williams, R., Hounsgaard, J. and Hollnagel, E. (2015). Where the rubber meets the road: using FRAM to align work-as-imagined with work-as-done when implementing clinical guidelines. *Implementation Science*, 10 (1), 1-8. doi: [10.1186/s13012-015-0317-y](https://doi.org/10.1186/s13012-015-0317-y) [GS Search]
- Cysneiros, L. M. and Leite, J. C. S. P. (2001). “Requisitos não funcionais: da elicitação ao modelo conceitual”. PhD Tese, PUC-RJ.
- C. Vargens, J. M. (2004). “Uma abordagem sociotécnica para design e desenvolvimento de sistemas de informação em saúde no âmbito do SUS”. PhD Tese, Rio de Janeiro.
- De Carvalho, E. A., Gomes, J. O., Jatobá, A. et al. (2021). Employing resilience engineering in eliciting software requirements for complex systems: experiments with the functional resonance analysis method (FRAM). *Cognition Technology and Work*, 1-19. doi: <https://doi.org/10.1007/s10111-019-00620-0>. [GS Search]
- Do Prado Leite, J. C. S., Santoro, F. M., Cappelli, C., Batista, T. V. and Santos, F. J. N. (2016). “Ownership relevance in aspect-oriented business process models”. *Business Process Management Journal*.
- Felicissimo, C. H. et al. 2004. “C&L: Um Ambiente para Edição e Visualização de Cenários e Léxicos”. Sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software Brasília, Brasil, 43-48.
- Frost, B. and Mo, J. P. (2014). System hazard analysis of a complex socio-technical system: the functional resonance analysis method in hazard identification. In: *Proc. of Australian System Safety Conference*, Melbourne Australia, 28 – 30.
- Gregoriades, A., Hadjicosti, J., Florides, C. and Pamapaka, M. (2015). “Human Requirements Validation for Complex Systems Design”. *Procedia Manufacturing*, 3, 3033–3040.
- Hollnagel, E. (2004). “Barriers and accident prevention”. Aldershot: Ashgate.
- Hollnagel, E. (2011). “Epilogue: RAG – The resilience analysis grid”. In E. Hollnagel et al. (eds), *Resilience Engineering in Practice: A guidebook*. Farnham, UK: Ashgate, 275-296.
- Hollnagel, E. (2012). *FRAM: The Functional Resonance Analysis Method*, Ashgate.
- Hollnagel, E. (2015). “Introduction to the Resilience Analysis Grid (RAG)”. In: *RAG- Resilience Analysis Grid*. [S.I]: Farnham, UK: Ashgate.

- Hollnagel, E., Hounsgaard, J. and Colligan, I. (2014). “FRAM-the Functional Resonance Analysis Method: a handbook for the practical use of the method”. Centre for Quality, Region of Southern Denmark.
- Hollnagel, E. and Slater, D. (2018). The Functional Resonance Analysis Method and Manual. Version 2. doi: 10.13140/RG.2.2.10569.24162.
- Hollnagel, E., Wears, R. L. and Braithwaite, J. (2015). “From Safety-I to Safety-II: a white paper”. The resilient health care net: published simultaneously by the University of Southern Denmark, University of Florida, USA, and Macquarie University, Australia.
- Hollnagel, E., Woods, D. and Leveson, N. (2006). Resilience Engineering: concepts and precepts. Ashgate, Farnham.
- Ingham, M. D., Rasmussen, R. D., Bennett, M. B. and Moncada, A. C. (2006). “Generating requirements for complex embedded systems using State Analysis”, *Acta Astronautica*, 58 (12), 648–661.
- Jaramillo, H. D. and Palacios, A. J. (2014). Requirements elicitation of resilience for systems information based on the model CERT-RMM. In: *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2014. 1-6.
- Katina, P. F., Keating, C. B. and Ra’ed, M. J. (2014). “System requirements engineering in complex situations”. *Requirements Engineering*, 19 (1), 45–62.
- Knight, Debora Mac., Araujo, Renata Mendes de and Borges, M. R. S. (2005). A Systematic Approach for Identifying System Requirements from the Organization's Business Model. In: *Simpósio Brasileiro de Sistemas de Informação, Florianópolis, Brasil. II Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS: Sociedade Brasileira de Sistemas de Informação, 1, 1-10.
- Leite, J. C. S. P. (1992). Enhancing the Semantics of Requirements Statements. In *Proceedings of the XII International Conference of the Sociedad Chilena de Ciencia de la Computacion*. Santiago, 281-297.
- Leite, J. C. S. P. (1994). “Engenharia de Requisitos”. Notas de Aula, PUC-RJ.
- Leite, J. C. S. P. A. (2001). “Gerenciando a Qualidade de Software com Base em Requisitos”. In: ROCHA, A. R. C., MALDONADO, J. C., WEBER, K. C. (org.). *Qualidade de Software Teoria e Prática*. 1 ed. São Paulo: Prentice-Hall. 1, 238-246.
- Leite, J. C. S. P. (2007). Aula 12. Engenharia de Requisitos. Disponível em: <<http://engenhariaderequisitos.blogspot.com/>>. Acesso em: 02 de agosto de 2020.
- Levenson, N. (2004). “A new accident model for engineering safer systems”. *Safety Science*, 42(4), 237-270.
- Li, R. C., Pereira, R. A. M., Jatobá, A., Vidal, M. C. R., de Carvalho, P. V. R., Grindrod, K. and Burns, C. (2019). “Information Technology Systems at the sharp end of medication therapy management”. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63 (1), 698–702. doi: [10.1177/1071181319631502](https://doi.org/10.1177/1071181319631502). [[GS Search](#)]
- Macchi, L. (2010). A Resilience Engineering approach for the evaluation of performance variability: development and application of the Functional Resonance Analysis Method for air traffic management safety assessment (Doctoral dissertation).
- Marino, T. B. and Campos, M. L. M. (2015). “Tratamento de informações geradas a partir de fontes de colaboração heterogêneas para apoio à resposta em emergências”. Tese de Doutorado. Programa de Pós-Graduação em Informática. Universidade Federal do Rio de Janeiro.

- Mayer, M., Odenthal, B., Faber, M., Winkelholz, C. and Schlick C. (2014). “Cognitive engineering of automated assembly processes”. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 24 (3), 348–368.
- McAllister, C. A. (2006). *Requirements determination of information systems: User and developer perceptions of factors contributing to misunderstandings*. Capella University.
- Méndez Fernández, D. and Wagner, S. (2015). “Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany”. *Information and Software Technology*, 57, 616–643. doi: [10.1016/j.infsof.2014.05.008](https://doi.org/10.1016/j.infsof.2014.05.008). [[GS Search](#)]
- Nardi, B. A. (1996). “Context and Consciousness - Activity Theory and Human-Computer Interaction”. MIT Press.
- Patriarca, R. et al. (2020). “Framing the FRAM: A literature review on the functional resonance analysis method”. *Safety Science*, 129, 104-827.
- Rasmussen, J. (1997). “Risk management in a dynamic society: a modelling problem”. *Safety science*, 27 (2-3), 183-213.
- Ratchev, S., Urwin, E., Muller, D., Pawar, K. S. and Moulek, I. (2003). “Knowledge based requirement engineering for one-of-a-kind complex systems”. *Knowledge-Based Systems*, 16 (1), 1–5.
- Saurin, T. A., Rosso, C. B. and Colligan, L. (2017). Towards a resilient and lean health care. In *Resilient health care*, 3 (17), 3-17.
- Sommerville, I. (2003) *Engenharia de Software*. [S.I]: Reading (MA): Addison-Wesley.
- Sundström, G. and Hollnagel, E. (2006). “Learning how to create resilience in business systems”. *Resilience Engineering. Concepts and Precepts*. Aldershot, UK: Ashgate.
- Tung, Y. W. and Chan, K. C. C. (2009). “A Unified Human–Computer Interaction Requirements Analysis Framework for Complex Socio-technical Systems”. *International Journal of Human–Computer Interaction*, 26 (1), 1–21.
- 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* (pp. 1-10).
- Woldeamlak, S., Diabat, A. S. and Vetinovic, D. (2016). “Goal-Oriented Requirements Engineering for Research-Intensive Complex Systems: A Case Study”, *Systems Engineering*. Disponível em: <https://onlinelibrary-wiley.ez29.capes.proxy.ufrj.br/doi/abs/10.1002/sys.21355>. Acessado em: 12 maio de 2020.
- Woods, D. and Branlat, M. (2011). “Basic patterns in how adaptive systems fail”. In: Hollnagel, E., Pariès, J., Woods, D.D., Wreathall, J. (Eds.), *Resilience Engineering in Practice: A Guidebook*. Ashgate Publishing, Ltd, 127–144.
- Yu, Y., Lapouchnian, A., Liaskos, S., Mylopoulos, J. and Leite, J. C. (2008). From goals to high-variability software design. In: *International Symposium on Methodologies for Intelligent Systems*, Springer, Berlin, Heidelberg, 1-16.