

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Renata Martinuzzi de Lima

**UM PROCESSO PARA GERENCIAMENTO DE REQUISITOS DE  
*SYSTEM OF SYSTEMS***

Santa Maria, RS  
2018

**Renata Martinuzzi de Lima**

**UM PROCESSO PARA GERENCIAMENTO DE REQUISITOS DE *SYSTEM OF SYSTEMS***

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestra em Ciência da Computação.**

ORIENTADORA: Prof.<sup>a</sup> Lisandra Manzoni Fontoura

Santa Maria, RS  
2018

de Lima, Renata  
Um Processo para Gerenciamento de Requisitos de  
System of Systems / Renata de Lima.- 2018.  
96 p.; 30 cm

Orientador: Lisandra Manzoni Fontoura  
Dissertação (mestrado) - Universidade Federal de Santa  
Maria, Centro de Tecnologia, Programa de Pós-Graduação em  
Ciência da Computação, RS, 2018

1. System of Systems 2. Sistema de Sistemas 3.  
Engenharia de Requisitos 4. Gerenciamento de Requisitos  
5. Engenharia de System of Systems I. Manzoni Fontoura,  
Lisandra II. Título.

**Renata Martinuzzi de Lima**

**UM PROCESSO PARA GERENCIAMENTO DE REQUISITOS DE *SYSTEM OF SYSTEMS***

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestra em Ciência da Computação.**

**Aprovado em 27 de abril de 2018:**

---

**Lisandra Manzoni Fontoura, Dra. (UFSM)**  
(Presidenta/Orientadora)

---

**Raul Ceretta Nunes, Dr. (UFSM)**

---

**Rejane Frozza, Dra. (UNISC)**

Santa Maria, RS  
2018

## DEDICATÓRIA

*À minha família*

## AGRADECIMENTOS

*Minha eterna e profunda gratidão em primeiro lugar aos meus pais, Jorge e Natalina, e ao meu irmão, Fernando. Por nunca medirem esforços para proporcionar todo suporte à minha vida acadêmica, por serem sempre meus incentivadores, por acreditarem em mim e por estarem sempre junto comigo, me apoiando nos momentos de dificuldade e comemorando nas conquistas. Muito Obrigada.*

*Meu muito obrigada à Professora Lisandra, minha orientadora, por todas as trocas de ideias e conversas produtivas. Obrigada por compartilhar comigo teu conhecimento, e por toda paciência, orientações e conselhos.*

*Aos pesquisadores integrantes do Projeto SIS-ASTROS por compartilharem seu tempo e conhecimento comigo durante o desenvolvimento do estudo de caso deste trabalho. E aos demais professores e colegas da UFSM, muito obrigada.*

*Gratidão à vida e ao universo por me colocar no caminho da pesquisa e da vida acadêmica. Que eu nunca perca a curiosidade e o amor pela pesquisa e produção de conhecimento.*

*When nothing is certain,  
everything is possible.*

*(Margaret Drabble)*

## RESUMO

### UM PROCESSO PARA GERENCIAMENTO DE REQUISITOS DE *SYSTEM OF SYSTEMS*

AUTORA: Renata Martinuzzi de Lima

ORIENTADORA: Lisandra Manzoni Fontoura

*System of systems* (SoS) estão se tornando cada vez mais comuns em nossa sociedade devido à diversidade de *stakeholders*, tecnologias heterogêneas, ubiquidade ou redes de grande escala envolvidas no desenvolvimento de sistemas intensivos em software. Consequentemente, existe um crescente interesse em fechar as lacunas que ainda existem no contexto da Engenharia de *System of Systems* (SoSE), um campo de pesquisa em constante crescimento devido à sua complexidade. A maioria dessas lacunas de pesquisa estão relacionadas ao gerenciamento e à Engenharia de Requisitos de SoS (ERSoS), a falta de métodos ou técnicas bem definidas para ERSoS torna difícil o desenvolvimento e evolução do SoS de modo geral. Portanto, neste trabalho propõe-se um processo para gerenciamento de requisitos de SoS capaz de organizar o trabalho entre as partes envolvidas e gerenciar as mudanças constantes nos requisitos, assim contribuindo para um desenvolvimento evolutivo do SoS mais adaptável e preditivo. Os resultados de uma avaliação de conformidade mostraram que o processo proposto está em conformidade com a maioria das cláusulas das normas e recomendações internacionais relacionadas à Engenharia de Sistemas e Engenharia de Requisitos. Além disso, um estudo de caso foi desenvolvido no contexto de um projeto real com o intuito de aplicar o processo na prática. Os resultados da avaliação qualitativa do estudo mostraram que a abordagem proposta cumpre com o objetivo de solucionar o problema identificado e obteve uma avaliação positiva por parte da equipe do projeto, em termos de correteza, usabilidade e efetividade no contexto aplicado.

**Palavras-chave:** Sistema de Sistemas; Engenharia de Requisitos; Gerenciamento de Requisitos.



## ABSTRACT

### A MANAGEMENT PROCESS FOR SYSTEM OF SYSTEMS REQUIREMENTS

AUTHOR: Renata Martinuzzi de Lima

ADVISOR: Lisandra Manzoni Fontoura

System of Systems (SoS) are becoming more common in our society due to the diversity of stakeholders, heterogeneous technology, ubiquity, or large-scale networks involved in the development of software intensive systems. Consequently, there is an increasing interest in closing the gaps still existing in the System of Systems Engineering (SoSE) context, which is a field in constant growth due to its complexity. Most of these gaps are related to the SoS requirements engineering or management, the lack of well-defined and standardized methods and techniques for Requirements Engineering for SoS (RESoS) makes difficult the development and the evolution of the overall SoS. Hence, in this work we propose a process of requirements management for SoS capable of organizing the work among stakeholders and managing changes in requirements, thus contributing to a more predictable and adaptable evolutionary development of the SoS. The results of a conformity assessment show that our process is in conformity with the most of international standards recommendations related to Systems Engineering and Requirements Engineering. In addition, a study case was developed in the context of a real project in order to apply the process in practice. The results of a qualitative validation identified that the proposed approach addresses the objective of to solve the identified problem and it had a positive evaluation, from the project team, in terms of correctness, usefulness, and effectiveness in the studied context.

**Keywords:** System of Systems; Requirements Engineering; Requirements Management.

## LISTA DE FIGURAS

Figura 4.1 – RMP-SoS - Fluxo principal .....	36
Figura 4.2 – Subprocesso <i>Manage Stakeholders</i> .....	37
Figura 4.3 – Subprocesso - <i>Elicit SoS Requirements</i> .....	41
Figura 4.4 – Subprocesso - <i>Manage Changes</i> .....	46
Figura 4.5 – Subprocesso - <i>Manage SoS Requirements</i> .....	48
Figura 5.1 – Arquitetura do SIS-ASTROS .....	59

## LISTA DE TABELAS

Tabela 5.1 – Resumo da Avaliação de Conformidade .....	57
Tabela 6.1 – Comparação entre os Trabalhos Relacionados .....	71

## LISTA DE ABREVIATURAS E SIGLAS

<i>BPMN</i>		<i>Business Process Modeling Notation</i>
<i>Conops</i>		<i>Concept of Operations</i>
<i>CS</i>		<i>Constituent Systems</i>
<i>DOD</i>	—	<i>USA Department of Defense of the United States</i>
<i>EB</i>		Exército Brasileiro
<i>ER</i>		Engenharia de Requisitos
<i>ES</i>		Engenharia de Sistemas
<i>ERSoS</i>		Engenharia de Requisitos para <i>System of Systems</i>
<i>ESoS</i>		Engenharia de <i>System of Systems</i>
<i>GR</i>		Gerenciamento de Requisitos
<i>ISO</i>		<i>International Organization for Standardization</i>
<i>IEC</i>		<i>International Electrotechnical Commission</i>
<i>IEEE</i>		<i>Institute of Electrical and Electronic Engineers</i>
<i>REOP</i>		Reconhecimento, Escolha e Ocupação de Posição
<i>RMP</i>	—	<i>SoS Requirements Management Process for System of Systems</i>
<i>RSL</i>		Revisão Sistemática de Literatura
<i>SDI</i>	—	<i>USA United States' Strategic Defense Initiative</i>
<i>SIS</i>	—	<i>ASTROS</i> Sistema Integrado de Simulação <i>ASTROS</i>
<i>SC</i>		Sistemas Constituintes
<i>SoS</i>		<i>System of Systems</i>
<i>SPEM</i>		<i>Software and Systems Process Engineering Metamodel</i>
<i>UFSM</i>		Universidade Federal de Santa Maria

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	DEFINIÇÃO DO PROBLEMA	13
1.2	JUSTIFICATIVA E HIPÓTESE	13
1.3	OBJETIVO GERAL	14
<b>1.3.1</b>	<b>Objetivos Específicos</b>	<b>14</b>
1.4	ORGANIZAÇÃO DO TEXTO	14
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>16</b>
2.1	SYSTEM OF SYSTEMS - (SOS)	16
<b>2.1.1</b>	<b>Características e Propriedades</b>	<b>17</b>
<b>2.1.2</b>	<b>Taxonomia de SoS</b>	<b>19</b>
2.2	ENGENHARIA DE SYSTEM OF SYSTEMS (ESOS)	20
<b>2.2.1</b>	<b>Elementos Centrais da ESoS</b>	<b>20</b>
2.2.1.1	<i>Elemento Central 1: Translating Capability Objectives</i>	21
2.2.1.2	<i>Elemento Central 2: Understanding System and Relationships</i>	21
2.2.1.3	<i>Elemento Central 3: Assessing Performance to Capability Objectives</i>	22
2.2.1.4	<i>Elemento Central 4: Developing and Evolving SoS Architecture</i>	22
2.2.1.5	<i>Elemento Central 5: Monitoring and Assessing Changes</i>	23
2.2.1.6	<i>Elemento Central 6: Addressing Requirements and Solution Options</i>	23
2.2.1.7	<i>Elemento Central 7: Orchestrating Upgrades to SoS</i>	24
<b>2.2.2</b>	<b>Artefatos da ESoS</b>	<b>24</b>
2.3	ENGENHARIA DE REQUISITOS (ER)	27
<b>2.3.1</b>	<b>Gerenciamento de Requisitos (GR)</b>	<b>28</b>
<b>3</b>	<b>MÉTODOS</b>	<b>30</b>
3.1	CARACTERIZAÇÃO DA PESQUISA	30
3.2	PROCEDIMENTOS METODOLÓGICOS	30
<b>3.2.1</b>	<b>Revisão Sistemática de Literatura (RSL)</b>	<b>31</b>
<b>3.2.2</b>	<b>Definição da Proposta</b>	<b>31</b>
<b>3.2.3</b>	<b>Construção e Modelagem do Processo</b>	<b>31</b>
<b>3.2.4</b>	<b>Validação da Pesquisa</b>	<b>32</b>
3.2.4.1	<i>Avaliação de Conformidade</i>	32
3.2.4.2	<i>Estudo de Caso e Pesquisa-ação</i>	32
<b>4</b>	<b>REQUIREMENTS MANAGEMENT PROCESS FOR SOS (RMP-SOS)</b>	<b>34</b>
4.1	DESCRIÇÃO GERAL DO PROCESSO	34
<b>4.1.1</b>	<b>Conceitos Importantes</b>	<b>34</b>
4.2	PAPÉIS	35
4.3	SUBPROCESSO 1 - MANAGE STAKEHOLDERS	36
<b>4.3.1</b>	<b>Task: Identify Stakeholders</b>	<b>37</b>
<b>4.3.2</b>	<b>Task: Assign Roles and Responsibilities</b>	<b>38</b>
<b>4.3.3</b>	<b>Task: Establish Collaboration Agreements</b>	<b>38</b>
<b>4.3.4</b>	<b>Task: Prepare Requirements Engineering Work Plan</b>	<b>39</b>
4.4	SUBPROCESSO 2 - ELICIT SOS REQUIREMENTS	40
<b>4.4.1</b>	<b>Task: Identify SoS Capability Objectives (SoS CO)</b>	<b>40</b>
<b>4.4.2</b>	<b>Task: Develop Concept of Operations (SoS Conops)</b>	<b>41</b>
<b>4.4.3</b>	<b>Define SoS Requirements Space (SoS ReqSpace)</b>	<b>42</b>
<b>4.4.4</b>	<b>Task: Get Initial Requirements from Architecture</b>	<b>43</b>

4.4.5	<i>Task: Review SoS CO, SoS Conops and SoS ReqSpace</i> .....	44
4.5	SUBPROCESS 3: MANAGE CHANGES .....	45
4.5.1	<i>Task: Identify CS Requirements Changes</i> .....	45
4.5.2	<i>Analyze Change Request</i> .....	47
4.6	SUBPROCESSO 4 - MANAGE SOS REQUIREMENTS .....	47
4.6.1	<i>Task: Update SoS CO, SoS Conops and SoS ReqSpace</i> .....	48
4.6.2	<i>Task: Document Recommended Requirements for SoS</i> .....	49
4.7	ARTEFATOS .....	50
5	<b>AVALIAÇÃO E VALIDAÇÃO DO PROCESSO</b> .....	<b>53</b>
5.1	AVALIAÇÃO DE CONFORMIDADE .....	53
5.1.1	<b>ISO IEC IEEE 15288 (2011) - Systems and software engineering — System life cycle processes</b> .....	53
5.1.2	<b>ISO IEC IEEE 29148 (2011) - Systems and software engineering - Life cycle processes - Requirements engineering</b> .....	54
5.1.3	<b>Discussão dos Resultados</b> .....	56
5.2	VALIDAÇÃO VIA ESTUDO DE CASO E PESQUISA-AÇÃO .....	58
5.2.1	<b>Identificação do Estudo:</b> .....	58
5.2.1.1	<i>O Projeto SIS-ASTROS</i> .....	58
5.2.2	<b>Etapa 1: Observação, leitura e análise de documentos</b> .....	60
5.2.3	<b>Etapa 2: Identificação do problema e verificação da realidade</b> .....	61
5.2.4	<b>Etapa 3: Aplicação do processo RMP-SoS no contexto do projeto SIS-ASTROS</b> .....	61
5.2.5	<b>Etapa 4: Análise da Aplicação e Apresentação para Equipe</b> .....	62
5.2.6	<b>Etapa 5: Avaliação do Estudo de Caso e Pesquisa-ação</b> .....	63
5.2.6.1	<i>Corretude</i> .....	63
5.2.6.2	<i>Usabilidade</i> .....	64
5.2.6.3	<i>Efetividade</i> .....	65
5.2.6.4	<i>Avaliação geral e sugestões</i> .....	66
5.2.6.5	<i>Discussão e Análise dos Resultados da Avaliação</i> .....	67
5.2.7	<b>Lições Aprendidas</b> .....	68
6	<b>TRABALHOS RELACIONADOS</b> .....	<b>69</b>
6.1	ENGENHARIA DE REQUISITOS PARA SOS .....	69
6.2	UM MÉTODO PARA DEFINIR REQUISITOS DE SOS .....	69
6.3	CAPACIDADES DE SOS PARA ENGENHARIA DE REQUISITOS .....	70
6.4	UM PROCESSO DE ENGENHARIA E GERENCIAMENTO DE REQUISITOS PARA FASE CONCEITUAL DE SISTEMAS COMPLEXOS .....	70
6.5	ANÁLISE DOS TRABALHOS RELACIONADOS .....	70
7	<b>CONCLUSÃO</b> .....	<b>72</b>
7.1	TRABALHOS FUTUROS .....	73
7.2	PUBLICAÇÕES .....	74
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>75</b>
	<b>APÊNDICE A – TABELA DE ATRIBUTOS X TAREFAS DO RMP-SOS</b> .....	<b>77</b>
	<b>APÊNDICE B – TEMPLATES DO RMP-SOS</b> .....	<b>83</b>
	<b>APÊNDICE C – ROTEIRO DE ENTREVISTA E DISCUSSÃO</b> .....	<b>96</b>

# 1 INTRODUÇÃO

Em diferentes domínios de aplicação, é possível encontrar a atuação de sistemas distintos, geralmente já existentes, operando em conjunto para satisfazer um determinado objetivo. Estes sistemas constituintes, juntos estabelecem o que é denominado de “Sistema de Sistemas” ou pelo termo em inglês “*System of Systems*” (SoS), termo que vem sendo usado para descrever estes sistemas complexos, compostos de sistemas constituintes independentes, os quais trabalhando em conjunto conseguem atingir um objetivo específico através da sua sinergia (NIELSEN et al., 2015).

A Engenharia de *System of Systems* (ESoS) é um assunto de crescente interesse dentro da comunidade de Engenharia de Sistemas (ES). No entanto, de acordo com o (SEBOK, 2016), as diferenças estruturais e a grande diversidade de *stakeholders* em um projeto de desenvolvimento de um SoS levam a complexidades e desafios ainda pouco explorados pela Engenharia de Sistemas tradicional. Os sete Elementos Centrais da ESoS, definidos pelo (DOD-USA, 2008), descrevem como processos da ES são aplicados ao contexto de SoS.

Três destes elementos relacionam-se diretamente com a Engenharia de Requisitos de SoS (ERSoS), são eles: “*Translating Capability Objectives*”, “*Developing and Evolving an SoS Architecture*”, e “*Addressing Requirements and Solution Options*” (DOD-USA, 2008). Por meio da descrição desses processos é possível ter-se uma ideia de como as atividades devem ser executadas, porém não há um padrão definido, nem um método consistente para gerenciamento, elicitação, análise, especificação, verificação ou validação dos requisitos de um SoS. Ou seja, o paradigma da ERSoS ainda não atingiu um nível de maturidade ou sofisticação experienciado pela ER tradicional. Portanto, as abordagens bem sucedidas no âmbito da ERSoS ainda são escassas e muitos problemas e desafios de pesquisa podem ser encontrados (LIMA; VARGAS; FONTOURA, 2017).

Uma questão recorrente citada em diversos estudos da área na última década, trata-se da necessidade de um gerenciamento de requisitos que facilite o trabalho colaborativo entre o grande número de *stakeholders* (NIELSEN et al., 2015) e que apoie a evolução do SoS de forma adaptável e flexível (DAHMANN; BALDWIN, 2008).

Um processo de gerenciamento de requisitos de SoS sistematizado seria capaz de clarificar as relações de gerenciamento entre o SoS e os sistemas constituintes facilitando a colaboração entre os diversos envolvidos que participam do ciclo de desenvolvimento de um SoS. Assim, opções de gerenciamento mais consistentes facilitarão a ERSoS, garantindo um processo de desenvolvimento mais previsível e adaptável (LIMA; FONTOURA, 2017).

## 1.1 DEFINIÇÃO DO PROBLEMA

(DAHMANN; ROEDLER, 2016) cita o tópico de capacidades e requisitos como um ponto fraco da Engenharia de SoS, pois tradicionalmente, ciclos de desenvolvimento de sistemas começam com um conjunto claro e completo de requisitos e fornecem uma abordagem disciplinada para o desenvolvimento de um sistema que atenda a estes requisitos.

Entretanto, um SoS compreende múltiplos sistemas independentes com seus próprios requisitos que podem ou não alinhar-se aos objetivos de capacidades do SoS (WALDEN et al., 2015). Nesse caso, (DAHMANN; ROEDLER, 2016) apontam a necessidade de clarificar relações de gerenciamento entre o SoS e os sistemas constituintes, garantindo as propriedades-chaves do SoS, ou seja, a independência operacional e gerencial de seus sistemas constituintes (NIELSEN et al., 2015). Além disso, (NIELSEN et al., 2015) afirmam que devido ao grande número de *stakeholders* envolvidos em um projeto de SoS, incluindo proprietários e operadores dos sistemas constituintes, há uma clara necessidade de se empregar métodos e ferramentas que suportem um trabalho colaborativo para elicitación de requisitos entre estes envolvidos.

Outro ponto importante a ser considerado é que o desenvolvimento de um SoS é um processo longo, contínuo, em constante evolução e suscetível a diversas mudanças nos requisitos. Como declara (LEWIS et al., 2009), em um SoS existem muitas oportunidades de mudança, por isso é possível afirmar que “mudança” é o estado normal de um SoS. Nesse sentido, há também a falta de mecanismos que possam antecipar as mudanças nos requisitos e avaliar seus impactos de forma conjunta entre gerentes e engenheiros dos sistemas constituintes (DAHMANN; BALDWIN, 2008).

(DAHMANN; ROEDLER, 2016) também declaram que a falta de uma padronização na Engenharia de *System of Systems* vem se tornando cada vez mais evidente, e assim há a oportunidade de definição de padrões ou adaptação de normas já existentes que possam ser aplicadas no âmbito da ESoS.

## 1.2 JUSTIFICATIVA E HIPÓTESE

A Engenharia de Requisitos é crucial para o sucesso de qualquer projeto de desenvolvimento de um sistema. A falta de um modelo sistemático de gerenciamento de tarefas para elicitar, analisar e especificar requisitos de um SoS é uma realidade bastante citada em produções científicas da Engenharia de SoS. Soluções em gerenciamento de requisitos podem ser encontradas no âmbito da Engenharia para sistemas complexos, no entanto essas não são efetivamente aplicáveis a SoS pois não contemplam as especificidades relevantes da interação entre o SoS e os sistemas constituintes, como a distribuição geográfica, o desenvolvimento evolucionário, e o comportamento emergente (LIMA; VARGAS; FONTOURA, 2017).

Um processo de gerenciamento de requisitos de SoS bem definido seria capaz de clarifi-



car as relações de gerenciamento entre o SoS e os sistemas constituintes facilitando a colaboração entre os diversos envolvidos que participam do ciclo de desenvolvimento de um SoS. Assim, opções de gerenciamento mais consistentes facilitariam a ERSoS, garantindo um processo de desenvolvimento mais coerente com a realidade e adaptável às necessidades de mudanças.

### 1.3 OBJETIVO GERAL

Desenvolver um processo para gerenciamento de requisitos de *System of Systems* que seja capaz de organizar o trabalho na interação entre os *stakeholders*, e então fornecer mecanismos flexíveis para adaptação de mudanças nos requisitos durante o processo evolutivo de um SoS. E ainda, com o intuito de clarificar relações de gerenciamento tornando-as mais consistentes e permitindo que ciclo de desenvolvimento evolutivo do SoS seja mais previsível.

#### 1.3.1 Objetivos Específicos

1. Integrar o gerenciamento de trabalho entre engenheiros e gerentes do SoS e dos Sistemas Constituintes, clarificando os papéis e responsabilidades na tomada de decisões.
2. Prover técnicas e meios de adaptação de mudanças nos requisitos, e avaliar seus impactos no desenvolvimento do SoS.
3. Alinhar as tarefas do processo de acordo com as normas internacionais no âmbito da Engenharia de Sistemas ou Engenharia de SoS, e da Engenharia de Requisitos.
4. Validar a aplicabilidade do processo proposto em um projeto real.

### 1.4 ORGANIZAÇÃO DO TEXTO

O texto desta dissertação está organizado da seguinte forma:

O Capítulo 2 traz a Revisão Bibliográfica, contextualizando os conceitos e definições mais importantes relacionados a *System of Systems* (SoS), Engenharia de SoS e Engenharia de Requisitos.

O Capítulo 3 descreve os métodos de trabalho, caracterizando a pesquisa desenvolvida e descrevendo todos os procedimentos metodológicos usados na concepção deste trabalho.

O Capítulo 4 apresenta a abordagem proposta, por meio da descrição do processo para gerenciamento de requisitos de SoS e das suas principais características e recursos.

O Capítulo 5 descreve as duas etapas de validação da proposta, por meio da Avaliação de Conformidade e do Estudo de Caso e Pesquisa-ação, bem como uma discussão dos resultados.

O Capítulo 6 apresenta uma revisão dos trabalhos de outros autores que estão relacionados a esta dissertação, e discute os pontos de comparação entre eles.

Finalmente o Capítulo 7 apresenta as conclusões obtidas ao final do trabalho, bem como ideias para trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

Este capítulo descreve os conceitos fundamentais relacionados à pesquisa. Relata-se de forma objetiva, os conceitos de *System of Systems*, bem como suas características, e taxonomia. Conceitos e Elementos Centrais da Engenharia de SoS, bem como seus processos e artefatos. Assim como aspectos importantes da Engenharia e Gerenciamento de Requisitos.

### 2.1 SYSTEM OF SYSTEMS - (SOS)

A primeira ideia de reunir sistemas independentes para desempenhar uma tarefa específica foi proposta pelo SDI-USA (*United States' Strategic Defense Initiative*) na década de 1980, assim diversas definições para essa ideia surgiram com o passar dos anos (NIELSEN et al., 2015). Uma fonte clássica da literatura na área de *System of Systems* (SoS), publicada na década de 1990, aborda a questão de que em um sentido formal, não há uma descrição ou definição considerada correta para SoS. Assim, (MAIER, 1996) tenta estabelecer se há uma distinção taxonomica útil entre sistemas complexos de larga escala que são geralmente chamados de *System of Systems*.

Por isso, atualmente uma diversidade de definições e conceitos para *System of Systems* pode ser encontrada na literatura, vejamos algumas delas. Para (MAIER, 1996), o uso do termo SoS implica em um grupo taxonômico, ou seja, implica que existem diferentes classes de sistemas, e essas classes são úteis em termos de Engenharia apenas se elas representarem diferentes demandas em *design*, desenvolvimento e operação. Sendo assim, o termo SoS é comumente usado para se referir a coleções de componentes suficientemente complexos a ponto de serem considerados sistemas, e reunidos em um sistema de maior escala.

(MAIER, 1996) ainda afirma que *System of Systems* é uma reunião de componentes que individualmente devem ser reconhecidos como sistemas e possuir duas características adicionais: (1) *Independência Operacional de Componentes* - O sistema constituinte deve ser capaz de operar de forma independente e útil; e (2) *Independência Gerencial de Componentes* - Os sistemas constituintes são adquiridos separadamente e integrados para fazer parte do SoS, mas devem manter uma contínua existência operacional independente do SoS.

Outra definição importante encontrada na literatura é dada pelo (DOD-USA, 2008), em seu *Systems Engineering Guide for Systems of Systems*, que considera SoS como um conjunto ou um arranjo de sistemas que resulta quando os sistemas independentes são integrados em um sistema maior que proporciona capacidades únicas. O guia ainda ressalta que um SoS é um sistema, mas que nem todos os sistemas são um SoS.

(KEATING et al., 2003) descreve SoS como “um meta-sistema, composto de múltiplos subsistemas embarcados, inter-relacionados, complexos e autônomos que podem ser diferentes

em tecnologia, contexto, operação, geografia, entre outros. Estes subsistemas complexos devem funcionar como um meta-sistema integrado para produzir resultados desejáveis em performance para atingir uma missão de alto nível”.

(ABBOTT, 2006) argumenta que *System of Systems* não podem ser vistos como uma versão maior de uma hierarquia de sistemas já conhecida. SoS são quantitativamente e estruturalmente diferentes de sistemas tradicionais. Para isso, o termo “*System of Systems*” deve ser reservado para esse novo paradigma. O autor conclui que a melhor maneira de pensar em SoS é como “uma coleção de sistemas participantes que compartilham regras e interfaces que evoluem lentamente o suficiente para que investir neles valha a pena”.

(BOARDMAN; SAUSER, 2006) argumentam que um SoS é um sistema, mas o fragmento “*of*” no termo em inglês “*System of Systems*” é importante, porque forma uma postura contrária a simples junção de sistemas, e refere-se aos sistemas que subordinam suas partes e relações para atender uma finalidade específica.

### 2.1.1 Características e Propriedades

(MAIER, 1996) define cinco principais características de SoS que são referenciadas pelo acrônimo “OMGEE” (iniciais dos termos em inglês: *Operational Independence, Managerial Independence, Geographic Distribution, Evolutionary Development, e Emergent Behaviour*):

- *Independência Operacional*: Qualquer sistema que faz parte de um SoS é independente e capaz de operar de maneira funcional mesmo que o SoS seja desmembrado.
- *Independência Gerencial*: Além de colaborar com os outros sistemas, os sistemas constituintes são gerenciados individualmente para que eles possam operar individualmente de maneira eficaz.
- *Distribuição Geográfica*: As partes que colaboram em um SoS estão distribuídas geograficamente, nesse caso salienta-se que os sistemas envolvidos só podem se comunicar entre si, mas não podem trocar quantidades consideráveis de massa ou energia.
- *Desenvolvimento Evolucionário*: A existência e o desenvolvimento de um SoS são evolucionários, em relação aos objetivos e funcionalidades que podem estar sob constante mudança. Um SoS nunca parece estar completamente formado.
- *Comportamento Emergente*: Através da colaboração entre os sistemas constituintes é possível atingir uma “sinergia” que faz com que o comportamento do sistema satisfaça um determinado objetivo que não pode ser atingido por qualquer um dos sistemas individualmente.

(ABBOTT, 2006) usa uma maneira diferente para caracterizar SoS. Ele declara que SoS são diferentes de sistemas tradicionais, pois eles são “Abertos no Topo” (*Open at the top*),

”Abertos no Fundo” (*Open at the bottom*) e ”Continuamente Evolutivos” (*Continually Evolving*).

- *Open at the Top*: Significa que em SoS não há nível superior de aplicação pré-definido. Ao invés disso, um SoS permite uma contínua introdução de novas aplicações.
- *Open at the Bottom*: Não há nível “mais baixo” fixado em SoS. O nível mais baixo de um SoS pode mudar a qualquer momento.
- *Continually Evolving*: Significa dizer que um SoS nunca está terminado. Ele pode evoluir continuamente assim como o ambiente em que ele está inserido muda. *System of Systems* podem evoluir de três maneiras (tecnologicamente, em usabilidade, e em padrões de interface).

(BOARDMAN; SAUSER, 2006) definem algumas características, bastante citadas em trabalhos posteriores, que diferenciam *System of Systems* de sistemas tradicionais, são elas:

- *Autonomia (Autonomy)*: É a habilidade de fazer escolhas independentes, o direito de buscar razões para ser e se comportar de tal forma. No que diz respeito aos sistemas constituintes e ao SoS, a autonomia exige que ambos sejam operacionalmente e gerencialmente independentes.
- *Pertencimento (Belonging)*: Sistemas legados podem precisar se submeter a mudanças doutrinárias a fim de ser útil dentro de um SoS. Na relação entre sistemas constituintes e o SoS, essa característica implica que os sistemas compartilhem um contexto de missão de forma intencional e compreendam seu papel no propósito do SoS.
- *Conectividade (Connectivity)*: É basicamente a habilidade de um sistema de se conectar a outros sistemas. Considerando que há muitos possíveis arranjos de sistemas, deve haver muitos meios de se atingir a proposta do SoS.
- *Diversidade (Diversity)*: Refere-se à heterogeneidade entre os sistemas constituintes, pois um SoS só é capaz de atingir seu propósito mais elevado apoiando-se na diversidade de seus sistemas constituintes. Ou seja, os sistemas constituintes têm uma identidade única e coletivamente compreendem um conjunto heterogêneo.
- *Emergente (Emergence)*: Significa que o comportamento dos sistemas constituintes que interagem com o SoS determina o comportamento coletivo. O termo implica que o SoS evolui, possui inteligência coletiva, exhibe sinergia entre os sistemas internos, funciona num ambiente dinâmico e se adapta a diferentes condições.

### 2.1.2 Taxonomia de SoS

(MAIER, 1996) declara que a necessidade de classificar *System of Systems* em distintas categorias é válida para guiar a seleção de princípios de arquitetura em sua construção. Esses tipos, definidos a seguir, são baseados no grau de controle gerencial de cada SoS, pois isso define quão adaptável e cooperativo cada sistema constituinte será em relação aos requisitos, interfaces e formatos de dados e tecnologias. E isso influencia nos desafios enfrentados na construção do *System of System* (NIELSEN et al., 2015).

O Departamento de Defesa dos Estados Unidos, (DOD-USA, 2008), baseado nos estudos de (MAIER, 1996) e (DAHMAN; BALDWIN, 2008), classifica os *System of Systems* em quatro tipos distintos de acordo com as relações estabelecidas entre o SoS e os sistemas constituintes.

- *SoS Virtual (Virtual)*: SoS virtuais não têm uma autoridade administrativa central nem uma proposta comum. Isso torna o comportamento do SoS e os objetivos a serem atingidos emergentes, o que pode ser uma característica desejável, mas nesse caso o SoS deve apoiar-se em mecanismos “invisíveis” para se manter. (MAIER, 1996) diz que o sistema é controlado pelas forças que cooperam entre si e que cumprem as normas fundamentais desejáveis.
- *SoS Colaborativo (Collaborative)*: Nesse tipo, os sistemas constituintes interagem voluntariamente para cumprir os propósitos centrais do SoS, ou seja, os sistemas constituintes não são “obrigados” a seguir um controlador central, eles podem fornecer ou negar um serviço, proporcionando assim meios para respeitar e manter os padrões de funcionamento do SoS. O SoS Colaborativo pode emergir para um SoS Reconhecido devido à importância das missões suportadas pelo SoS ou pela complexidade das interações entre os constituintes. SoS podem ser expostos a uma repentina reconfiguração, mesmo que tenham uma considerável direção central.
- *SoS Dirigido (Directed)* São aqueles que o SoS é integrado e construído para atingir propósitos específicos. Ele é centralmente gerenciado durante operações de longo prazo para continuar cumprindo esses propósitos assim como novos que os sistemas queiram abordar. Os sistemas componentes mantêm a sua habilidade de operar independentemente, mas seu modo de operação normal é subordinado ao propósito central do SoS.
- *SoS Reconhecido (Acknowledged)*: De acordo com (DAHMAN; BALDWIN, 2008), essa categoria reconhece objetivos, um gerente designado para tal e recursos para o SoS. No entanto, os sistemas constituintes mantêm sua propriedade independente, seus próprios objetivos, desenvolvimento, financiamento e sustentação. As mudanças no sistema são baseadas na colaboração entre o SoS e os sistemas constituintes.

## 2.2 ENGENHARIA DE SYSTEM OF SYSTEMS (ESOS)

O (SEBOK, 2016) define a Engenharia de SoS (ESoS) não como uma nova disciplina, mas como uma nova oportunidade para a comunidade de Engenharia de Sistemas (ES) de definir os sistemas complexos do século 21. Enquanto a ES tradicional é um campo já estabelecido, a ESoS representa um desafio aos engenheiros de sistemas a nível global. De forma geral, a ESoS requer considerações além daquelas geralmente associadas à engenharia, incluindo fenômenos sócio-técnicos e às vezes socioeconômicos.

De acordo com o *Systems Engineering Guide for System of Systems* do (DOD-USA, 2008), a Engenharia de SoS presta uma considerável atenção e investe um grande tempo e esforço entendendo mudanças que estão fora de seu escopo de controle, mas que podem potencialmente impactar o SoS.

Seu escopo gira em torno do controle de requisitos e abordagens de implementação, acompanha o desenvolvimento, integração e teste, e ainda avalia o impacto das mudanças nas necessidades de capacidades do usuário final. Assim, o papel central da Engenharia de SoS é estabelecer e manter um *framework* técnico persistente para guiar a evolução do SoS, através do desenvolvimento de uma arquitetura evolutiva. A arquitetura de um SoS promove uma visão integrada do conjunto de sistemas constituintes do SoS (DOD-USA, 2008).

A Engenharia de SoS então foca a atenção em um conjunto de elementos centrais conforme eles evoluem um conjunto de sistemas novos ou existentes a fim de atender os objetivos de capacidade do usuário. De acordo com (DAHMANN et al., 2011), os engenheiros de SoS são os “jogadores-chave” para cumprir as tarefas de traduzir os objetivos de capacidades em requisitos de SoS, avaliar até que ponto esses objetivos estão sendo atendidos e monitorar e avaliar o impacto de mudanças externas no SoS. Os objetivos principais da Engenharia de SoS são: entender os sistemas que contribuem com o SoS e suas relações, desenvolver uma arquitetura para o SoS que age como um *framework* persistente a fim de atender aos requisitos do SoS e opções de solução. E, finalmente, o Engenheiro de SoS orquestra as melhorias enquanto monitora e integra as mudanças feitas nos sistemas a fim de melhorar o desempenho do SoS.

### 2.2.1 Elementos Centrais da ESoS

O Departamento de Defesa dos Estados Unidos (DOD-USA, 2008) descreve a Engenharia de Sistemas em ambientes SoS por meio de um conjunto composto por sete elementos centrais. Estes elementos fornecem um contexto para a aplicação de processos da Engenharia de Sistemas tradicionais no contexto de *System of Systems*.

Estes sete elementos relacionam-se entre si e são conduzidos de forma contínua, sem uma estrutura de tempo ou sequenciamento definida. E podem ser conduzidos por membros de equipes de engenharia do SoS únicas ou múltiplas, dependendo do tamanho e escopo do

SoS. Além de serem implementados pelos Engenheiros de SoS, há uma notável participação dos Engenheiros dos Sistemas Constituintes conforme necessidade e características do SoS.

#### 2.2.1.1 *Elemento Central 1: Translating Capability Objectives*

A primeira tarefa do Engenheiro de SoS é desenvolver um entendimento básico das expectativas para o SoS e dos requisitos principais para atender a essas expectativas. Os objetivos de um SoS são geralmente declarados em termos de amplas capacidades. Assim, este elemento busca transformar os objetivos de capacidade em requisitos, ou seja, o engenheiro e o gerente de SoS tem a tarefa de clarificar e operacionalizar estes objetivos e expectativas dos *stakeholders* em um conjunto de requisitos pronto para a incorporação a uma futura *baseline* técnica do SoS.

Esse elemento fornece ao engenheiro de SoS um entendimento amplo das prioridades e suas relações, o que será útil no desenvolvimento e gerenciamento dos requisitos. Porém, como o SoS tende a evoluir com o passar do tempo, o engenheiro de SoS precisa entender e continuar rastreando as dinâmicas das mudanças conforme elas influenciam os objetivos e expectativas dos *stakeholders* (DOD-USA, 2008).

#### 2.2.1.2 *Elemento Central 2: Understanding System and Relationships*

De acordo com o (DOD-USA, 2008), um dos aspectos mais importantes da Engenharia de SoS é desenvolver um entendimento sobre os sistemas envolvidos e suas relações e interdependências. A importância das interfaces entre sistemas constituintes de um SoS é que elas permitem acesso ao seu comportamento, e isso envolve entender o conjunto de sistemas que impactam as capacidades do SoS e o modo que eles interagem e contribuem no atendimento dos objetivos de capacidades.

O comportamento e o desempenho do SoS são criados a partir da combinação das interações (processos e fluxo de dados) dentro e em torno do SoS. Por isso essas interfaces são críticas para o sucesso do SoS. Os limites e as interfaces podem ser dinâmicos, pois os sistemas interagem com um ou mais sistemas em tempos diferentes para atingir as capacidades do SoS, e em alguns casos, fornecendo serviços a outros sistemas (DOD-USA, 2008).

Esse elemento envolve diferentes dimensões: primeiro, pensamos em definir as funcionalidades dos sistemas e como eles irão compartilhar dados durante as operações. Por segundo, as relações entre sistemas constituintes, como por exemplo: as relações organizacionais entre sistemas; os *Stakeholders*, incluindo os usuários do SoS e dos SC, e seus contextos organizacionais como base para seus papéis na ESoS; os requisitos; e as relações entre os processos e planos de desenvolvimento dos SC e do SoS.



### 2.2.1.3 *Elemento Central 3: Assessing Performance to Capability Objectives*

Teste e Avaliação de SoS são conduzidos em dois níveis: “DT and E” (*Developmental Test and Evaluation*) e “OT and E” (*Operational Test and Evaluation*). Esse elemento central suporta o “OT and E”, e foca no desenvolvimento de métricas e na coleta de dados de uma variedade de configurações ao longo do tempo para monitorar o desempenho do SoS no que diz respeito aos objetivos do usuário.

Nesse elemento, o engenheiro do SoS trabalha com a equipe de teste e avaliação a fim de estabelecer medidas e métodos de desempenho e avaliação. Nesse nível, o desempenho é medido em termos de objetivos de capacidade com foco na utilidade dos recursos do SoS para o usuário. Portanto, essas métricas tem o objetivo de mensurar o comportamento e o desempenho do SoS em operações reais. Além disso, as métricas orientadas ao usuário devem ser capazes de ser aplicadas através da implementação ou de ambientes operacionais.

Devido à complexidade, quando as mudanças feitas para suportar os objetivos do SoS são introduzidas em um ambiente operacional, interações inesperadas são comuns. Portanto, o teste e a avaliação de SoS precisam identificar e avaliar a capacidade adicional fornecida por essas interações inesperadas, isso inclui a necessidade de considerar possíveis interações prejudiciais entre os sistemas e como essas interações podem ser identificadas e gerenciadas.

### 2.2.1.4 *Elemento Central 4: Developing and Evolving SoS Architecture*

O aspecto chave da ESoS é estabelecer um *framework* técnico persistente para endereçar a evolução do SoS, a fim de atingir as expectativas dos usuários, incluindo possíveis mudanças nas funcionalidades, desempenho ou interface dos sistemas (DOD-USA, 2008). Essa arquitetura não fornece detalhes dos sistemas individuais, pelo contrário, ela define o modo como os sistemas trabalham juntos para atender as necessidades dos usuários e aborda a implementação de sistemas individuais apenas quando uma funcionalidade é fundamental para questões do SoS.

A arquitetura de um SoS, é de alguma forma, limitada pela estrutura e conteúdo dos sistemas constituintes, particularmente à medida que mudanças nos sistemas são acessíveis e viáveis, já que os sistemas precisam continuar a operar em outras configurações em paralelo com a sua participação no SoS. O processo de design do SoS gera como saída uma arquitetura física que define os componentes físicos do SoS (os sistemas constituintes).

Idealmente, a arquitetura de um SoS irá persistir durante múltiplos incrementos do desenvolvimento do SoS, permitindo mudanças em algumas áreas, e promovendo estabilidade em outras. Ao passar do tempo o SoS vai enfrentar mudanças de diferentes dimensões: nos objetivos de capacidade, na experiência do usuário real, mudanças na tecnologia ou no Conops (Conceito de Operações). Além das mudanças imprevistas nos sistemas, que podem afetar a viabilidade da arquitetura e podem pedir por mudanças. Conseqüentemente, o engenheiro de

SoS deve regularmente avaliar a arquitetura para garantir que esta suporta a evolução do SoS.

#### 2.2.1.5 *Elemento Central 5: Monitoring and Assessing Changes*

A proposta desse elemento é antecipar as mudanças que estão fora do controle do SoS, mas que podem afetar suas funcionalidades ou desempenho de suas capacidades. Isso inclui as mudanças nas tecnologias usadas para suportar o SoS ou mudanças nas missões individuais de cada sistema constituinte, bem como demandas externas do SoS. Além disso, a equipe de engenharia de SoS precisa estar ciente do desenvolvimento e das atividades de modernização programadas para o SoS e para os SC.

É fundamental que o engenheiro de SoS se envolva com os engenheiros dos sistemas constituintes para entender a natureza de suas mudanças e avaliar os impactos potenciais no SoS. Para atender a essa necessidade, alguns SoS podem estabelecer meios de configuração precoce onde os engenheiros dos sistemas constituintes são convidados a compartilhar todas as mudanças antecipadas com o engenheiro de SoS no início dos processos de planejamento.

#### 2.2.1.6 *Elemento Central 6: Addressing Requirements and Solution Options*

Em um processo de desenvolvimento de SoS, os engenheiros dos SC trabalham em conjunto com o gerente e engenheiro do SoS para revisar, priorizar e recomendar quais requisitos serão implementados a cada iteração, assim os requisitos selecionados para cada iteração são movidos para a *baseline* funcional. Essa análise inclui o controle das mudanças nos requisitos de SoS de alto nível para manter a estabilidade e coerência.

A equipe de SoS é responsável por liderar o desenvolvimento, avaliação e seleção de abordagens técnicas para atender aos requisitos. Assim, o produto dessas atividades é um plano técnico para a evolução do SoS geralmente através de mudanças incrementais por parte dos sistemas constituintes e às vezes com componentes adicionais especificamente para os SoS.

As opções para atender aos requisitos podem incluir: adicionar novos sistemas, adicionar sistemas já existentes mas novos ao SoS, atualizar ou estender funcionalidades de sistemas existentes, ou convencer os SC a se adaptarem às mudanças em função do SoS. A evolução dos requisitos é muitas vezes conduzida por uma variedade de fontes: mudanças no ambiente do SoS, comportamentos emergentes; mudanças nos SC; problemas de atualização do SoS; perspectivas e necessidades dos usuários; e oportunidades tecnológicas.

### 2.2.1.7 Elemento Central 7: Orchestrating Upgrades to SoS

Este elemento opera como um complemento do Elemento Central 6. Aqui, o engenheiro de SoS facilita, monitora e coordena as mudanças implementadas nos sistemas para efetuar melhorias de desempenho e adicionar capacidades ao SoS. Tanto em nível de SoS quanto nos sistemas constituintes, os engenheiros precisam trabalhar com os gerentes do programa para determinar a melhor fase de algumas das iterações para se coordenar para atender aos ritmos de atualização agendados.

Este elemento é desencadeado quando as várias organizações envolvidas na implementação aceitam um plano técnico para abordar os requisitos do SoS. Mais importante ainda, o plano inclui os gerentes e engenheiros dos SC que estarão implementando as mudanças que concordaram durante o desenvolvimento do plano. Este plano é então executado sob este elemento.

### 2.2.2 Artefatos da ESoS

(DAHMANN et al., 2011) afirma que os Artefatos de SoS podem ser vistos como “objetos de limite” que, em alguns casos, fazem a ponte entre os elementos centrais da ESoS e em outros casos, entre as diferentes organizações responsáveis pelos sistemas constituintes do SoS.

Alguns desses artefatos definidos por (DAHMANN et al., 2011) estão diretamente ligados à Engenharia de Requisitos, outros são gerados ao longo do ciclo de vida evolutivo do SoS e indiretamente contribuem para o processo de gerenciamento dos requisitos. Uma breve definição de cada um é dada a seguir.

1. ***SoS Capability Objectives***: É uma declaração geral sobre os objetivos de alto nível para o SoS. Eles descrevem as capacidades necessárias pelo usuário, idealmente baseados em algum material autoritário (políticas, diretivas, entre outros). São usados pelo gerente de SoS, pelos *stakeholders* e pelas equipes de engenharia como base para obter os requisitos e métricas do SoS. Os Objetivos de Capacidade fornecem a base para traduzir as necessidades operacionais em requisitos de alto nível, para avaliar o desempenho e para desenvolver a arquitetura e as opções de solução para o SoS.
2. ***SoS Conops (Concept of Operations)***: Descreve como as funcionalidades dos sistemas no SoS serão empregadas numa configuração operacional. É desenvolvido pelos engenheiros do SoS, com participação ativa dos usuários que têm a função de descrever a forma que planejam usar os sistemas para atingir seus objetivos, e influenciados pelos diversos ambientes e condições previamente definidas. O SoS Conops é desenvolvido em paralelo com o *SoS Capability Objectives*, e os engenheiros do SoS o usam para definir o *SoS Requirements Space*, para identificar aspectos dos sistemas que podem impactar o design do SoS, e para selecionar métricas de desempenho e ambientes de teste.

3. ***Systems Information***: É coletado e organizado pela equipe de engenharia do SoS e usado como base para negociações conforme o SoS evolui. Descreve informações dos SC que impactam os objetivos de capacidade do SoS e inclui aspectos técnicos e programáticos dos SC que são relevantes ao SoS. O conteúdo é produzido pelos diversos *stakeholders*, incluindo os usuários operacionais e organizações envolvidas com o SoS e os SC. As informações desse documento ajudam a equipe de ESoS a entender os componentes do SoS, incluindo suas perspectivas técnicas, organizacionais e físicas, que são base para o desenvolvimento e evolução da arquitetura, para o monitoramento e avaliação das mudanças de ambos, o SoS e dos SC, assim como para desenvolver soluções para as capacidades do SoS.
4. ***SoS Requirements Space***: Este artefato limita as necessidades de primeira ordem dos usuários, incluindo tarefas e missões opcionais, e define as funções requeridas para fornecer as capacidades considerando a variabilidade do ambiente do usuário que impacta na forma como as funções serão executadas. Inclui as capacidades que precisam ser atendidas e relatório de problemas. É desenvolvido pela equipe de engenharia do SoS, com fortes conexões entre as comunidades operacionais do SoS e dos SC e com as equipes de engenharia dos SC. O *SoS Requirements Space* é usado para: determinar as informações necessárias para entender os sistemas e suas relações, comparar o desempenho com os objetivos de capacidade, desenvolver uma arquitetura de SoS consistente, identificar as áreas que serão atendidas em determinado incremento, identificar as opções de solução, elaborar um plano de incrementos do SoS, e ainda para desenvolver um plano de teste e avaliação de mudanças.
5. ***SoS Performance Measures and Methods***: Fornece a base para avaliar o desempenho geral do SoS e planejar a melhoria contínua do mesmo. Essas medidas de desempenho são rastreáveis em relação aos objetivos de capacidade do SoS. Elas são criadas pelas equipes de engenharia do SoS e dos SC, em conjunto com a comunidade de teste e avaliação a fim de avaliar o status e progresso dos objetivos do SoS. Além disso, são usadas para estruturar eventos e gerar os dados necessários.
6. ***SoS Performance Data***: Esse artefato, juntamente com dados sobre imprevistos observados durante a análise de performance, são coletados de diferentes ambientes pelas equipes de engenharia do SoS e dos SC para avaliar o progresso no sentido de alcançar os objetivos de capacidade. Esses dados são usados para avaliar o impacto das mudanças e identificar áreas que precisam de maior atenção. Além disso, fornecem *feedback* sobre a variabilidade na implementação da arquitetura, fatores que impactam as capacidades, e necessidades adicionais em termos de objetivos de capacidade baseadas na experiência dos usuários operacionais. Assim, funcionam como base para atender aos requisitos e organizar as atualizações do SoS.

7. ***SoS SE Planning Elements***: Fornece a estrutura e um processo para a Engenharia de SoS, como um Plano de Engenharia de Sistemas faz em um programa de aquisição. Elementos-chave incluem: ritmo de atualizações do SoS; estruturas organizacionais e processos de decisão; e revisões técnicas. Esses elementos são desenvolvidos e evoluídos pela equipe de engenharia do SoS e pelas equipes de engenharia dos SC chave. Os elementos fornecem regras de engajamento de ES básicas para o SoS e são utilizadas pelos *stakeholders* para entender o processo do SoS como um todo.
8. ***SoS Risks and Mitigations***: São artefatos abordados ao longo do processo de desenvolvimento do SoS. A equipe de engenharia do SoS trabalha em conjunto com as equipes dos SC a fim de encontrar riscos potenciais associados com as capacidades do SoS. O status dos riscos e sua mitigação são atualizados periodicamente e rastreados pelas equipes de engenharia do SoS, dos SC e pelos *stakeholders* do SoS para compreender possíveis riscos, problemas e obstáculos para alcançar as capacidades desejadas e orientar a seleção de soluções alternativas.
9. ***SoS Master Plan***: É um plano integrado que fornece uma visão superior sobre várias atualizações incrementais para implementação da estratégia de evolução do SoS. É um plano desenvolvido e evoluído pela equipe de engenharia do SoS com a colaboração das equipes de engenharia dos SC. É usado para entender o estado atual e os planos seguintes de desenvolvimento do SoS, pois este fornece suporte a decisões e ajuste das prioridades ao longo do tempo.
10. ***Agreements***: Busca formalizar papéis e responsabilidades dos *stakeholders* em um nível amplo, bem como formalizar compromissos específicos em uma etapa do desenvolvimento. Acordos são fundamentais para o sucesso da engenharia do SoS, considerando que o SoS atravessa fronteiras organizacionais entre os seus sistemas constituintes.
11. ***SoS Architecture***: É um *framework* técnico e persistente que suporta a evolução do SoS para atender às necessidades dos usuários e atender a possíveis mudanças em termos de funcionalidades, desempenho ou interfaces do sistema. Uma arquitetura define a forma como os sistemas constituintes funcionam juntos. Inclui os sistemas, as funções essenciais do SoS suportadas pelos SC, seus relacionamentos e dependências, além de funcionalidades de ponta-a-ponta, fluxo de dados e protocolos de comunicação.
12. ***SoS Technical Baselines***: São linhas de base desenvolvidas para cada incremento de desenvolvimento do SoS. Incluem uma linha de requisitos, uma para base alocada, uma para base de produto do SoS e linhas de base detalhadas dos sistemas, mantidas por eles próprios. São usadas para entender o estado atual do SoS, monitorar as melhorias para o próximo incremento, e para planejar mudanças para incrementos futuros.

13. **Technical Plans:** É desenvolvido um plano para cada incremento do desenvolvimento do SoS e inclui planos de implementação, integração e teste. Seguem os princípios de planejamento técnico para sistemas, prestando atenção na definição, revisões e riscos criados por eventos críticos ao longo do processo evolutivo. Os planos são usados para orientar atividades e documentar acordos sobre mudanças a serem feitas em determinado incremento do SoS, para acompanhar o progresso das implementações de mudanças e os problemas com a implementação.
14. **Integrated Master Schedules (IMS):** Também é criado um para cada incremento do SoS. Incluem os pontos-chaves dos planos técnicos que precisam ser abordados no desenvolvimento do SoS. Concentra-se nas principais atividades da ESoS, nos pontos de integração e *links* para o desenvolvimento detalhado dos cronogramas de atualização dos sistemas.

### 2.3 ENGENHARIA DE REQUISITOS (ER)

A descrição das funções e das restrições de um sistema de software são os *requisitos* para um sistema. (SOMMERVILLE, 2011) afirma que é importante conhecer os dois diferentes níveis de descrição de requisitos, pois a maioria dos problemas surgem quando não existe uma clara separação entre esses níveis. Os *Requisitos de Usuário* são aqueles descritos apenas por uma declaração abstrata em alto nível (geralmente em linguagem natural), de um serviço que o sistema deve oferecer ou uma restrição a um sistema; e os *Requisitos de Sistema*, nos quais as definições das funções do sistema, dos serviços e restrições são mais detalhadas e formais.

O conjunto de tarefas e técnicas que levam a um entendimento dos requisitos é chamado de *Engenharia de Requisitos* (ER) (PRESSMAN, 2011). Complementando esse conceito, (SOMMERVILLE et al., 2003) diz que a ER é o processo de descobrir, documentar, analisar e verificar as funções e restrições de um sistema.

A Engenharia de Requisitos fornece os mecanismos apropriados para entender aquilo que o cliente deseja que seja desenvolvido, analisando suas necessidades, avaliando a sua viabilidade, negociando uma solução razoável, especificando a solução sem ambiguidades, validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema (PRESSMAN, 2011).

Estes mecanismos são descritos de diferentes formas por diferentes autores. Por exemplo, (SOMMERVILLE, 2011) define que a ER é formada por processos que podem incluir quatro atividades de alto nível: Estudo de Viabilidade; Elicitação e Análise; Especificação; e Validação. Já (PRESSMAN, 2011), diz que a ER abrange sete tarefas distintas: Concepção; Levantamento; Elaboração; Negociação; Especificação; Validação; e Gestão. Entretanto, ambas abordagens possuem os mesmos objetivos, de avaliar se a concepção do sistema é realmente útil para o cliente, levantando os requisitos necessários e os convertendo em algum formato padrão,

entendível pelos *stakeholders*, e ainda verificar se estes realmente definem o sistema que o cliente espera, adaptando-se às mudanças que possam surgir durante o ciclo de desenvolvimento do sistema.

O produto final da ER é um documento de requisitos de um sistema, em inglês chamado de *SRS-System Requirements Specification*, que é uma declaração oficial de “o que” os desenvolvedores do sistema devem implementar. Esse documento de requisitos tem um conjunto de usuários diversificado, ou seja, diversos *stakeholders* participam de sua produção e revisão. Por isso, o documento precisa ser um compromisso com a comunicação dos requisitos para o cliente, ter a definição dos requisitos em detalhes para os desenvolvedores e testadores do sistema, e deve incluir informações relevantes para a evolução do sistema (SOMMERVILLE, 2011).

(PRESSMAN, 2011) ainda afirma que os requisitos de sistemas mudam, e que o desejo de mudar os requisitos persiste ao longo da vida de um sistema. Para isso existe o *Gerenciamento de Requisitos*, uma atividade adicional da ER, que ajuda a equipe a identificar, controlar e acompanhar as necessidades e suas mudanças a qualquer momento enquanto o projeto de desenvolvimento do sistema evolui.

### **2.3.1 Gerenciamento de Requisitos (GR)**

São as atividades que garantem que os requisitos sejam identificados, documentados, mantidos, comunicados e rastreados ao longo do ciclo de vida de um sistema (IEEE, 2011). Além disso, (SOMMERVILLE et al., 2003) diz que GR é o processo de compreender e controlar as mudanças nos requisitos de sistemas, e é realizado em conjunto com outros processos da ER.

O planejamento do GR é um estágio essencial ao processo de gerenciamento de requisitos, pois é onde toma-se decisão sobre aspectos como: identificação de requisitos, processo de gerenciamento de mudanças, políticas de rastreabilidade e suporte de ferramentas de apoio. Aqui é importante o destaque para a necessidade de se pensar em políticas de rastreabilidade de requisitos. (SAYÃO; LEITE, 2006), dizem que a rastreabilidade vem sendo identificada como um fator de qualidade importante no desenvolvimento de sistemas, e esta pode ser definida como uma técnica para prover relacionamentos entre requisitos, arquitetura e implementação final do sistema. Estes relacionamentos permitem a visualização de que o projeto atende aos requisitos e ajuda na detecção prévia de requisitos não atendidos.

Além disso, uma das etapas mais importantes da Gerência de Requisitos, trata-se do conjunto de atividades que avalia o impacto e o custo das mudanças, e deve ser aplicado a todas as mudanças propostas para os requisitos. Assim, todas elas serão tratadas de modo consistente, e as mudanças no documento de requisitos poderão ser feitas de maneira controlada. Nesse sentido, (SOMMERVILLE et al., 2003) define três estágios principais para um processo de Gerenciamento de Mudanças: análise do problema e especificação da mudança; análise e custo da mudança e implementação de mudança. Além disso, para efetivamente analisar o impacto

das mudanças, é necessário que a fonte de cada requisito seja conhecida e que o fundamento lógico de qualquer mudança seja documentado.



### 3 MÉTODOS

O presente capítulo tem por objetivo apresentar a caracterização do trabalho desenvolvido nessa pesquisa, bem como a descrição dos procedimentos metodológicos, ou seja, qual foi a sequência de passos necessários para atingir os objetivos desta dissertação.

#### 3.1 CARACTERIZAÇÃO DA PESQUISA

De acordo com (YIN, 2010) e (GIL, 2010), uma pesquisa pode ser caracterizada de acordo com a sua natureza, estratégia, tipo e área de conhecimento e método de pesquisa.

Sendo assim, a pesquisa desenvolvida pode ser classificada como sendo de natureza *Aplicada*, pois objetiva a produção de conhecimentos para aplicações práticas dentro de uma área específica. A estratégia usada é *Qualitativa*, ou seja, os resultados são interpretativos e consistem em uma análise e interpretação dos dados que não se podem traduzir em números. Quanto ao tipo de pesquisa pode ser considerada *Exploratória*, cujo o objetivo é proporcionar uma maior familiaridade com o problema, a fim de torná-lo mais explícito para construir hipóteses para sua solução (YIN, 2010) (GIL, 2010).

A área de conhecimento é definida como "*Ciências Tecnológicas*" de acordo com a classificação apresentada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). A pesquisa tem caráter *Experimental*, pois de acordo com (GIL, 2010), por meio dessa pesquisa é possível identificar uma busca bibliográfica, um estudo de caso e pesquisa-ação.

#### 3.2 PROCEDIMENTOS METODOLÓGICOS

No primeiro momento do desenvolvimento deste trabalho, a pesquisa concentrou-se em obter os conceitos básicos para entendimento do tema principal - *System of Systems* - além de conhecer o estado da arte a fim de elencar tópicos de interesse com base na leitura de artigos científicos publicados em anais de eventos e *journals*. Após observação de que o tópico "Engenharia de Requisitos" dentro do contexto de SoS apresentava diversas lacunas de pesquisa, uma Revisão Sistemática de Literatura, usando a abordagem *Snowballing* foi desenvolvida com o objetivo de reunir estudos atuais sobre o tópico, e principalmente encontrar desafios de pesquisa a serem explorados (LIMA; VARGAS; FONTOURA, 2017).

A seguir é apresentado um resumo da Revisão Sistemática de Literatura desenvolvida, bem como um breve detalhamento sobre as etapas seguintes do desenvolvimento deste trabalho: Definição da Proposta; Construção e Modelagem do Processo e Validação.

### 3.2.1 Revisão Sistemática de Literatura (RSL)

A RSL foi desenvolvida usando a abordagem *Snowballing*, buscando por responder as quatro questões de pesquisa: (Q1): *Quais estudos são relacionados à tradução dos objetivos de capacidades em requisitos de SoS de alto nível?*; (Q2): *Existe algum meio comumente usado para especificar requisitos de SoS?*; (Q3): *Quais estudos são relacionados ao gerenciamento de requisitos no processo evolutivo do SoS?* e (Q4): *Quais são os desafios de pesquisa encontrados no âmbito da Engenharia de Requisitos para SoS?*

A revisão identificou dezenove estudos publicados entre os anos de 2005 e 2016, que respondiam as questões de pesquisa. Como o principal objetivo era elencar problemas de pesquisa, o resultado da Q4 identificou que entre os desafios de pesquisa mais citados estava a necessidade de um gerenciamento de requisitos e das suas mudanças ao longo do processo evolutivo do SoS. Além disso, muitos desafios técnicos relacionados às atividades de validação e verificação de requisitos que considerem as características intrínsecas de um SoS, também foram identificados (LIMA; VARGAS; FONTOURA, 2017).

O artigo (LIMA; VARGAS; FONTOURA, 2017), produzido durante esta etapa do trabalho foi submetido e aceito em um evento de nível internacional, qualificado como *Qualis B1*, a *29th International Conference on Software Engineering and Knowledge Engineering - (SEKE 2017)*. O artigo foi apresentado no dia 05 de Julho de 2017 na cidade de *Pittsburgh (PA), EUA* e publicado nos anais do evento.

### 3.2.2 Definição da Proposta

Após a identificação do tópico de pesquisa e objeto de estudo, o problema de pesquisa foi definido, e então traçada uma proposta de solução. Um artigo de proposta foi submetido e aceito no *X Workshop de Teses e Dissertações em Sistemas de Informação*, evento satélite do XIII Simpósio Brasileiro de Sistemas de Informação - (SBSI 2017) e apresentado no dia 08 de Junho de 2017, na cidade de Lavras, MG e publicado nos anais do evento (LIMA; FONTOURA, 2017). Após apresentação do artigo e seminário de andamento em Julho de 2017, a proposta de solução ao problema foi refinada e iniciou-se a construção e modelagem do processo.

### 3.2.3 Construção e Modelagem do Processo

O processo RMP-SoS foi desenvolvido com base em uma cuidadosa análise de literatura científica relacionada a requisitos de SoS. Assim, pode-se afirmar que este processo para gerenciamento de requisitos de SoS é uma compilação de ideias já apresentadas em estudos publicados anteriormente, combinados com métodos e técnicas oriundos da Engenharia de Sistemas tradicional e da Gerência de Projetos.

Os atributos essenciais para modelagem do processo foram coletados através dessa análise de literatura científica, diretrizes e recomendações no contexto de SoS, e estes foram relacionados com as características intrínsecas dos SoS (já mencionadas no capítulo 2) e então subprocessos, tarefas e artefatos foram definidos e relacionados a cada conjunto de atributos para criação desse processo. Esses atributos e suas relações podem ser visualizados na tabela do Apêndice A. O Processo para Gerenciamento de Requisitos de SoS, nomeado de **RMP-SoS** (*Requirements Management Process for SoS*), foi modelado usando o Metamodelo SPEM 2.0, através da ferramenta *Eclipse Process Framework Composer*. E seu detalhamento será apresentado no capítulo 4

### 3.2.4 Validação da Pesquisa

Para validar o trabalho desenvolvido nesta dissertação, duas etapas foram seguidas: Na primeira delas, foi executada uma Avaliação de Conformidade com as normas internacionais existentes no contexto da Engenharia de Sistemas e da Engenharia de Requisitos. Na segunda etapa, a fim de verificar a aplicabilidade do processo, foi desenvolvido um estudo de caso combinado com uma pesquisa-ação no contexto do Projeto SIS-ASTROS (detalhes do projeto são descritos na seção 5.2.1.1 do capítulo 5). Os resultados de ambas etapas são descritos e analisados no Capítulo 5.

#### 3.2.4.1 Avaliação de Conformidade

De acordo com a norma ABNT NBR ISO/IEC 17000 de 2005, uma Avaliação de Conformidade pode ser a “demonstração de que requisitos especificados relativos a um produto, processo, sistema, pessoa ou organismo são atendidos”. Sendo assim, podemos afirmar que uma avaliação feita para verificar se um objeto atende a determinadas condições pré-estabelecidos, encaixa-se nesse conceito.

Neste trabalho, foi desenvolvida uma avaliação de conformidade considerando os padrões internacionais aceitos no âmbito da Engenharia de Sistemas (ISO/IEC/IEEE 15288) e Engenharia de Requisitos (ISO/IEC/IEEE 29148), com o objetivo de identificar e explicar como o processo proposto atende as cláusulas especificadas nestas normas.

#### 3.2.4.2 Estudo de Caso e Pesquisa-ação

Para aplicação e validação do processo RMP-SoS usou-se um método misto como abordagem. Métodos mistos de pesquisa surgiram como uma estratégia mais complexa de pesquisa

pelo reconhecimento de que todos os métodos têm limitações, assim os pontos fracos de um determinado método podem ser compensados pelos pontos de fortes de outro (EASTERBROOK et al., 2008).

Considerando que esta é uma pesquisa de postura “teórica-crítica”, de acordo com a classificação apresentada por (EASTERBROOK et al., 2008), os autores indicam que as melhores abordagens de pesquisa sejam o *estudo de caso* e a *pesquisa-ação*.

Um estudo de caso pode ser definido como um “inquérito empírico que investiga um fenômeno contemporâneo em seu contexto na vida real, especialmente quando as fronteiras entre fenômeno e contexto não são claramente evidentes” (YIN, 2010). Um estudo de caso *exploratório* é usado para investigações iniciais de algum fenômeno para derivar novas hipóteses e construir teorias. Já um estudo de caso *confirmatório* é usado para testar uma teoria existente ou proposta. Como ponto fraco dessa abordagem, o autor identifica que a coleta de dados e a análise dos mesmos pode ser aberta a diversas interpretações e têm um viés de pesquisa (EASTERBROOK et al., 2008).

A pesquisa-ação é comumente usada em campos como educação, psicologia, ou ciências sociais, e nos últimos anos vem também sendo adotada no campo da computação. Nesta abordagem, os pesquisadores tentam resolver um problema do mundo real, enquanto simultaneamente estudam a experiência da resolução do problema e com a proposta explícita de ajudar a melhorar a situação (EASTERBROOK et al., 2008). Uma condição prévia para executar uma pesquisa-ação é ter um “dono de problema” disposto a colaborar com a pesquisa, tanto para identificar o problema, quanto para envolver-se na sua resolução.

Pesquisa-ação é um processo cíclico que permite que a “ação” (mudança ou melhoria) e “pesquisa” (entendimento e conhecimento) sejam atingidos ao mesmo tempo (KHAN; TZORTZOPOULOS, 2016). Para avaliar a qualidade de uma pesquisa-ação, deve-se considerar a *autenticidade do problema*, ou seja, se ele é realmente importante e se precisa de solução, e a *autenticidade do conhecimento* gerado aos participantes, com ênfase em identificar lições de aprendizado úteis. Entretanto, de acordo com (EASTERBROOK et al., 2008), a maior dificuldade da pesquisa-ação é sua relativa imaturidade como um método empírico.

Sendo assim, para validar a abordagem proposta nesta dissertação, propõe-se um estudo de caso confirmatório com o objetivo de provar que a teoria deste trabalho é verdadeira através da observação da unidade de análise, neste caso a Engenharia de Requisitos do Projeto SIS-ASTROS, e como complemento, uma pesquisa-ação, na qual a aplicação do processo RMP-SoS é executada, a fim de gerar conhecimento por meio de lições aprendidas.

A coleta e avaliação dos dados foi feita através de uma entrevista seguindo como roteiro o *framework* proposto por (KHAN; TZORTZOPOULOS, 2016) após algumas adaptações para o domínio desta pesquisa, no qual os participantes puderam opinar sobre os benefícios e pontos a melhorar da prática. A descrição detalhada dessa etapa é apresentada no Capítulo 5.

## 4 REQUIREMENTS MANAGEMENT PROCESS FOR SOS (RMP-SOS)

Este capítulo descreve o processo proposto de forma detalhada, apresentando suas tarefas, papéis, artefatos e diagramas. O RMP-SoS (*Requirements Management Process for SoS*) é um processo para gerenciamento de requisitos de SoS que tem como objetivo principal organizar as tarefas executadas pelos envolvidos com a tradução de objetivos de capacidades em requisitos de alto-nível e no gerenciamento de mudanças dos mesmos, permitindo assim que o ciclo de vida evolutivo do SoS seja mais consistente e previsível.

### 4.1 DESCRIÇÃO GERAL DO PROCESSO

O Processo para Gerenciamento de Requisitos de SoS, nomeado de **RMP-SoS** (*Requirements Management Process for SoS*), foi modelado usando o Metamodelo SPEM 2.0, por meio da ferramenta *Eclipse Process Framework Composer* e pode ser visualizado no endereço <rmpos.com >. Seu fluxo de trabalho principal é representado pelo diagrama BPMN na Figura 4.1.

O RMP-SoS consiste em quatro subprocessos organizados de acordo com o foco de suas tarefas, que são: gerenciar as partes interessadas (*Manage Stakeholders*), elicitar requisitos de SoS (*Elicit SoS Requirements*), gerenciar mudanças (*Manage Changes*) e gerenciar requisitos de SoS (*Manage SoS Requirements*).

Os atributos necessários para construção do processo foram levantados por meio de uma análise de literatura científica, bem como normas e recomendações no âmbito da Engenharia de Sistema de Sistemas, já mencionados no capítulo 2, e as relações entre estes atributos com os subprocessos e tarefas desenvolvidos encontram-se especificados na Tabela do apêndice A.

Para um melhor entendimento das tarefas do RMP-SoS, alguns conceitos importantes são introduzidos inicialmente. Os papéis fundamentais para o desenvolvimento das tarefas, a descrição de cada subprocesso bem como suas tarefas, passos e os artefatos gerados são definidos a seguir.

#### 4.1.1 Conceitos Importantes

Para um melhor entendimento das tarefas e produtos do processo, três conceitos que definem os "tipos de necessidade dos *stakeholders*", definidos por (HOLT et al., 2012), são considerados fundamentais.

- Meta (*Goal*): define e descreve um **resultado desejado** para o qual os esforços de um sistema ou projeto são direcionados.

- Capacidade (*Capability*): descreve a capacidade ou habilidade de uma organização de pessoas ou sistemas de fazer algo para **cumprir as metas declaradas**. Uma capacidade é geralmente demonstrada pela execução de um conjunto de processos definidos.
- Requisito (*Requirement*): define uma **propriedade** de um sistema que é necessária ou desejada por um *stakeholder* para fornecer as capacidades declaradas. Para um SoS, os requisitos são geralmente descritos pelas capacidades necessárias que um SoS deve entregar.

## 4.2 PAPÉIS

Papéis são utilizados para identificar os principais atores (*workers*) envolvidos no processo de gerenciamento de requisitos, que irão desempenhar as tarefas e ser responsáveis pelos artefatos gerados.

- Gerente de Requisitos de SoS (*SoS Requirements Manager*): Responsável pelo gerenciamento das partes envolvidas nas atividades de requisitos, pelos acordos de colaboração entre essas partes e pelo gerenciamento das atividades desse processo. O Gerente de Requisitos de SoS pode ser a mesma pessoa que desempenha o papel de Gerente do Projeto, caso este esteja envolvido também no gerenciamento de requisitos.
- Engenheiro de Requisitos de SoS (*SoS Requirements Engineer*): Responsável direto pela execução das tarefas de elicitação e gerenciamento de Requisitos do SoS.
- Gerente de Requisitos do SC (*CS Requirements Manager*): Responsável pelo gerenciamento de requisitos de cada Sistema Constituinte em particular.
- Engenheiro de Requisitos do SC (*CS Requirements Engineer*): Responsável pela engenharia de requisitos de cada Sistema Constituinte em particular.
- Usuário Operacional do SoS (*SoS Operational User*): Usuário operacional do SoS, aquele que fará uso do sistema em tempo de operação integrada.
- Comitê de Controle de Mudanças (*Change Control Board*): É formado pelo *SoS Requirements Manager*, *SoS Requirements Engineer*, *CS Requirements Engineer*, comitê responsável por tomar decisões sobre implementar ou não mudanças propostas e avaliar seu custo e impacto..
- Cliente (*Customer*): Cliente do projeto, não desempenha tarefas no processo de forma direta.

- *Outros Stakeholders (Other Stakeholders)*: Outras pessoas envolvidas a nível de projeto que não desenvolvem ativamente tarefas do processo.

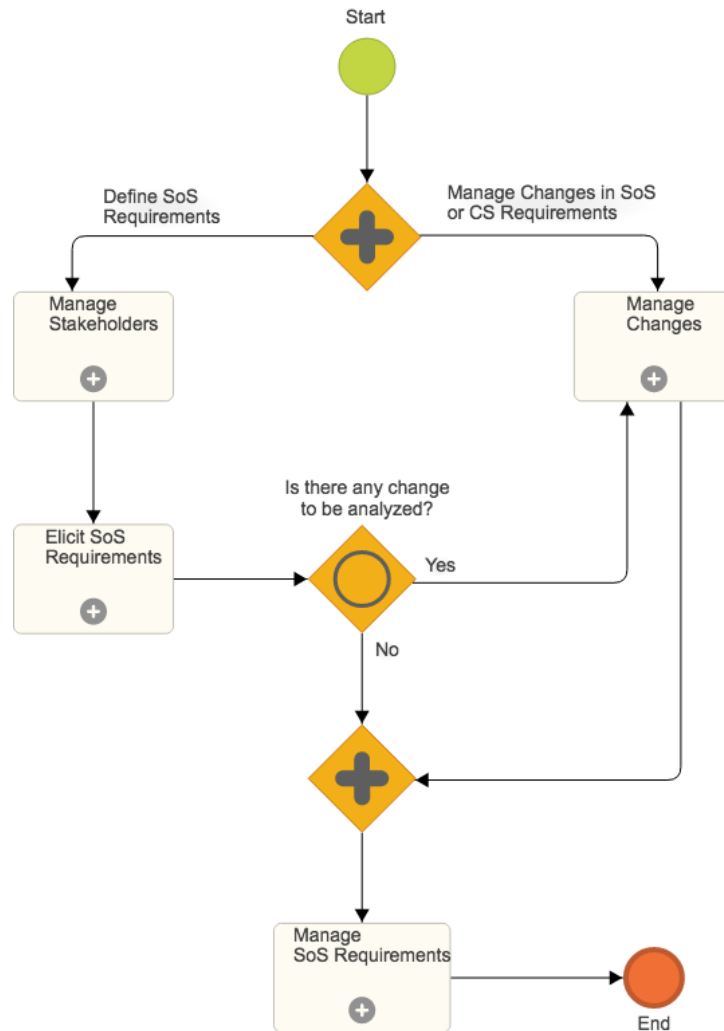


Figura 4.1 – RMP-SoS - Fluxo principal

#### 4.3 SUBPROCESSO 1 - MANAGE STAKEHOLDERS

Este primeiro subprocesso tem o objetivo de identificar as pessoas envolvidas na Engenharia de Requisitos do SoS, bem como suas habilidades, conhecimentos e atribuir os papéis e responsabilidades de cada um. Além disso, busca estabelecer acordos entre esses *stakeholders* e traçar um plano consistente para execução das atividades da ER do SoS. O fluxo das tarefas propostas nesse subprocesso é representado pelo diagrama BPMN na Figura 4.2.

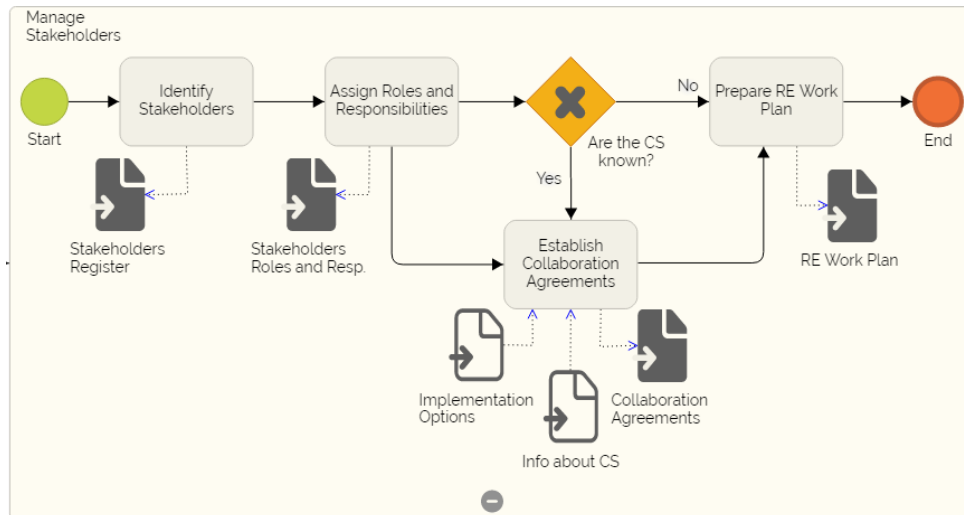


Figura 4.2 – Subprocesso *Manage Stakeholders*

#### 4.3.1 Task: *Identify Stakeholders*

Essa tarefa tem como objetivo identificar as principais partes envolvidas nas atividades de requisitos ligadas ao SoS e aos Sistemas Constituintes. É importante ressaltar que cada *stakeholder* tem habilidades e conhecimentos que serão indispensáveis na definição dos requisitos. Nesse caso, além de identificá-los é necessário que suas habilidades sejam descritas para a organização de um ambiente de trabalho em que ele possa contribuir de forma apropriada (PMI, 2008).

O responsável por essa tarefa é o Gerente de Requisitos de SoS, que conta com a ajuda do Engenheiro de Requisitos de SoS. Os passos dessa tarefa são descritos a seguir:

1. **Listar todas as partes interessadas:** Fazer um levantamento inicial de todas as possíveis partes interessadas no processo da engenharia de requisitos do SoS.
2. **Entender os interesses de cada stakeholder:** Após listar e identificar os *stakeholders*, é necessário entender quais são os interesses de cada um dentro do projeto e dentro do processo da Engenharia de Requisitos. É importante identificar interesses conflitantes nesse momento.
3. **Descrever habilidades e conhecimento:** Identificar as habilidades e o nível de conhecimento de cada *Stakeholder* a fim de atribuir as atividades adequadas no processo.
4. **Estabelecer os níveis de influência dos stakeholders:** O nível de influência corresponde à capacidade que a parte interessada tem de alavancar os resultados do projeto ou de paralisá-lo.
5. **Classificar os stakeholders por ordem de importância:** Identificar os *stakeholders-chaves* e estabelecer qual seu nível de importância no projeto.



O artefato de saída é chamado de *Stakeholders Register* e seu template pode ser visualizado no apêndice B.

#### 4.3.2 *Task: Assign Roles and Responsibilities*

Essa tarefa tem como objetivo principal distribuir os papéis e responsabilidades necessários para a execução das atividades de Engenharia e Gerenciamento de Requisitos de SoS. Nota-se que papéis e responsabilidades podem variar ao longo do processo evolutivo do SoS. Entretanto, a maioria dos papéis serão atribuídos aos *stakeholders* diretamente envolvidos no trabalho de definição de requisitos (PMI, 2008).

Um número de papéis é definido por padrão por diferentes guias e manuais, entretanto estes podem ser adaptados de acordo com o contexto de cada projeto, desde que sigam determinadas regras (PRINCE2, 2017). O responsável por essa tarefa é o Gerente de Requisitos de SoS. E os passos dessa tarefa são descritos a seguir:

1. **Identificar as habilidades e responsabilidades necessárias para cada tarefa:** Analisar as tarefas da Engenharia de Requisitos de SoS e identificar quais habilidades e quais condições de acesso são necessárias para conclusão eficaz das mesmas.
2. **Identificar os *stakeholders* com as habilidades necessárias:** Consultar o "*Stakeholder Register*" a fim de identificar quem da equipe é um potencial responsável para determinada tarefa.
3. **Distribuir os papéis e responsabilidades adequados:** Distribuir os papéis e responsabilidades adequados a cada *Stakeholder* de acordo com seus conhecimentos e habilidades, seguindo o padrão escolhido pelo gerente do projeto.

O artefato de saída é chamado de *Stakeholders Roles and Responsibilities* e seu template pode ser visualizado no apêndice B.

#### 4.3.3 *Task: Establish Collaboration Agreements*

Com o objetivo de obter um engajamento entre os todos os *stakeholders* que participam das atividades de requisitos de SoS e dos SC, essa tarefa visa gerar como produto de trabalho documentos que expressem esses acordos e os níveis de colaboração das partes nas atividades da Engenharia de Requisitos de SoS.

É importante ressaltar que em determinadas tarefas da ER, é fundamental que o engenheiro do SoS se envolva com os engenheiros dos SC para entender a natureza de suas mudanças e avaliar os impactos potenciais nos SoS. Para atender a essa necessidade, é necessário

que acordos sobre compartilhamento de dados e informações sejam estabelecidos (DOD-USA, 2008).

Os responsáveis por essa tarefa são o Gerente de Requisitos de SoS e o Engenheiro de Requisitos de SoS. Os passos dessa tarefa são descritos a seguir:

1. **Identificar as tarefas que precisam de colaboração:** Identificar as tarefas que precisam de colaboração das partes envolvidas nos Sistemas Constituintes e SoS.
2. **Definir as informações a serem compartilhadas:** Definir quais dados e informações precisam ser compartilhadas entre SC e SoS, ou seja, quais os dados de requisitos dos SC o Engenheiro de Requisitos de SoS precisa ter acesso.
3. **Estabelecer condições de acesso:** Definir as condições e regras de acesso (“o que” e “como”) aos dados e informações necessários para execução de determinada tarefa.
4. **Estabelecer acordos:** Assinar e registrar acordos conforme a necessidade.

Os artefatos de saída são documentos formais ou informais que definem e caracterizam estes acordos, chamados de *Collaboration Agreements*.

#### 4.3.4 *Task: Prepare Requirements Engineering Work Plan*

A fim de autorizar o início das atividades da Engenharia de Requisitos é preparado um plano de trabalho. Nesse plano é indispensável que sejam descritos: o tipo de SoS e seu ambiente, um breve resumo do escopo do projeto, a proposta de trabalho nessa fase do projeto, as restrições, as políticas de rastreabilidade, bem como a definição de um glossário (ARNAUT; FERRARI; SOUZA, 2016).

O responsável por esta tarefa é o Engenheiro de Requisitos de SoS, e os passos da mesma são descritos a seguir.

1. **Identificar o tipo de SoS e descrever o ambiente:** Definir se o SoS em questão é *Virtual*, *Acknowledge*, *Directed* ou *Collaborative*, de acordo com seu controle operacional, e descrever a principal característica que determina essa escolha. Ainda, descrever o ambiente operacional do SoS (entidades, influência, evolução).
2. **Resumir escopo do projeto:** Obter o plano de desenvolvimento do projeto e incluir um resumo com todas as informações relevantes para as atividades de requisitos.
3. **Descrever a proposta de trabalho:** Descrever brevemente como as atividades da ER serão realizadas, e em que ordem elas devem ser executadas, bem como revisadas e atualizadas. Além disso, realizar o planejamento em termos de tempo e custo para cada atividade.

4. **Incluir restrições:** Identificar e descrever quaisquer restrições que condicionem a execução das atividades da ER do SoS.
5. **Incluir política de rastreabilidade de requisitos:** Definir políticas que estabelecem os relacionamentos entre os objetivos de capacidades desejados e os *stakeholders*, bem como entre os objetivos de capacidades e os requisitos. Além disso, definir como esses registros serão mantidos.
6. **Autorizar o início das atividades de requisitos:** Com o *RE Work Plan* pronto, a execução das atividades de requisitos pode ser iniciada.
7. **Incluir Glossário:** Definir um glossário com os termos e abreviaturas usadas nos documentos produzidos neste processo e seus respectivos significados.

O artefato de saída é chamado de *RE Work Plan* e seu template pode ser visualizado no apêndice B. Note que o plano de trabalho pode ser revisado e atualizado conforme as atividades são executadas e precisem de ajustes.

Ao final deste subprocesso, o *RE Work Plan* atual é enviado ao próximo subprocesso a fim de servir como base para a primeira tarefa *Identify SoS Capability Objectives*.

#### 4.4 SUBPROCESSO 2 - ELICIT SOS REQUIREMENTS

O segundo subprocesso tem a proposta de organizar o fluxo das tarefas ligadas à eliciação de requisitos de SoS, assim identificando os objetivos de capacidade, desenvolvendo o conceito de operações, definindo o espaço de requisitos e os requisitos de interoperabilidade definidos pela arquitetura do SoS, através das tarefas descritas a seguir. A forma como estas tarefas e seus artefatos estão organizadas é representada pelo diagrama BPMN na Figura 4.3.

##### 4.4.1 *Task: Identify SoS Capability Objectives (SoS CO)*

Busca identificar em alto nível os objetivos de capacidades para o SoS, descrevendo na linguagem do usuário "o que" precisa ser feito pelo SoS, ou seja, as missões primárias e secundárias, as restrições operacionais, e definir a rastreabilidade entre os *stakeholders*, seus objetivos e os objetivos de capacidades para o SoS (ARNAUT; FERRARI; SOUZA, 2016).

O responsável por esta tarefa é o Engenheiro de Requisitos de SoS, e os passos da mesma são descritos a seguir.

1. **Analisar as metas gerais do projeto:** Analisar e entender o objetivo geral do projeto do SoS, identificar a proposta específica e as metas que motivam o desenvolvimento e a execução do SoS.

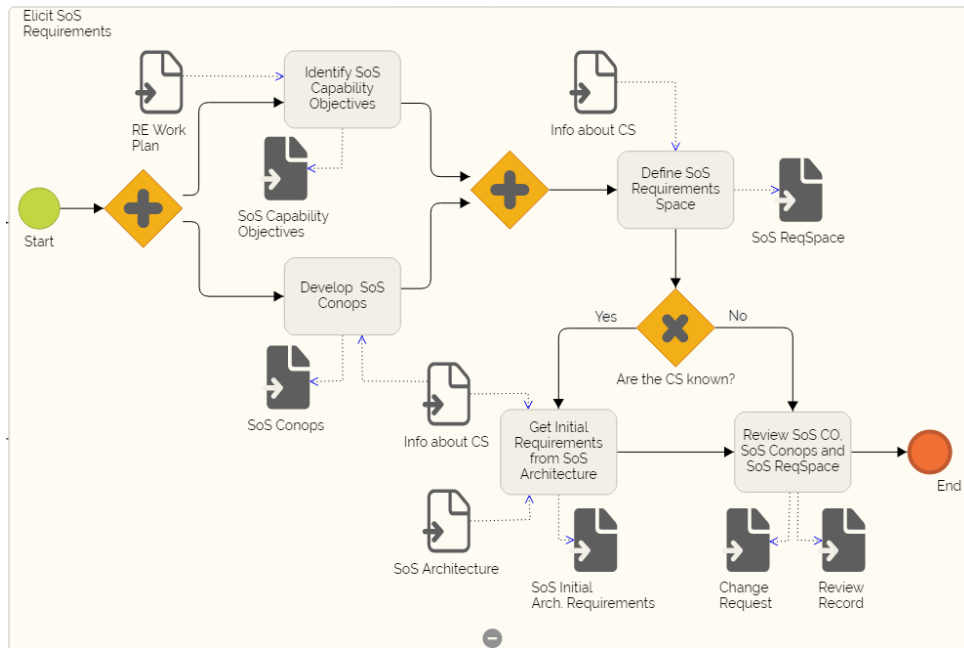


Figura 4.3 – Subprocesso - *Elicit SoS Requirements*

2. **Identificar restrições:** Identificar as restrições de desenvolvimento disponíveis e conhecidas.
3. **Descobrir os objetivos dos stakeholders:** Nessa interação com os Stakeholders, descobrir quais são suas necessidades e expectativas que o SoS precisa atender.
4. **Estabelecer rastreabilidade dos objetivos dos stakeholders:** Estabelecer a rastreabilidade dos objetivos de capacidade desejados pelos stakeholders conforme políticas de rastreabilidade previamente definidas.
5. **Registrar informações:** Registrar as informações identificadas em uma declaração de alto nível.
6. **Refinar as informações e documentar no SoS CO:** Refinar as informações registradas e documentar os objetivos de capacidade no documento *SoS Capability Objectives*.

O artefato de saída é nomeado por *SoS Capability Objectives* (disponível no apêndice B), e é importante ressaltar que este documento pode vir a ser atualizado no decorrer do processo, por meio de tarefas de revisão e atualização.

#### 4.4.2 *Task: Develop Concept of Operations (SoS Conops)*

Esta atividade é desenvolvida em paralelo com a *Identify Capability Objectives* e tem o objetivo de descrever a forma como o sistema funciona do ponto de vista do usuário operacional. O SoS Conops é desenvolvido pelo Engenheiro de SoS com ativa participação dos usuários

operacionais do SoS, que irão descrever o modo como planejam operar e usar o SoS para atingir seus objetivos. Portanto, deve incluir a descrição do usuário e resumir as necessidades, metas e características da comunidade de usuários do sistema (SEBOK, 2016).

O responsável é o Engenheiro de Requisitos do SoS, com ativa participação dos Usuários Operacionais do SoS. Os passos desta tarefa são descritos a seguir.

1. **Descrever o escopo do documento:** Descrever brevemente a visão geral do SoS, a proposta para implementação do SoS, os recursos necessários e os *stakeholders* envolvidos.
2. **Obter visão geral do SoS:** Descrever em alto nível os requisitos de missão, as inter-relações dos sistemas constituintes-chaves, e a interdependência entre eles.
3. **Descrever o ambiente operacional pela perspectiva do usuário:** Resumir o papel de cada usuário no ambiente operacional do SoS, esclarecer a ordem das operações, resumir procedimentos, e descrever diagramas de fluxo associados a tomada de decisão.
4. **Descrever o ambiente operacional:** Descrever a infraestrutura associada aos SC (essa tarefa só pode ser realizada em incrementos finais, onde já se conhece os SC do SoS).
5. **Descrever Cenários Operacionais:** Descrever o sistema “em ação”, usando um ou mais cenários de representação para refletir as perspectivas dos *stakeholders*.

O artefato de saída é o documento *SoS Conops*, cujo template está disponível no apêndice B. E é importante ressaltar que este documento também pode vir a ser atualizado no decorrer do processo, por meio de tarefas de revisão e atualização.

#### 4.4.3 *Define SoS Requirements Space (SoS ReqSpace)*

Tem o objetivo de limitar as necessidades primárias dos usuários do SoS e definir as funções para fornecer as capacidades, considerando a variabilidade do ambiente de usuário que impacta o modo como estas funções serão executadas.

Essa tarefa é desenvolvida pela equipe de Engenharia de SoS, com relações entre as equipes de engenharia dos SC. Portanto, o responsável é o Engenheiro de Requisitos de SoS (DAHMAN et al., 2011). Os passos desta tarefa são descritos a seguir.

1. **Listar as necessidades de primeira ordem dos usuários:** Limitar as necessidades dos usuários, incluindo as tarefas operacionais e missões.
2. **Definir as funcionalidades para atender as capacidades:** Definir as funcionalidades que são necessárias para atender as capacidades desejadas do SoS.

3. **Identificar variabilidades no ambiente de usuário:** Identificar as possíveis variabilidades no ambiente de usuário que impactam a forma como as funcionalidades serão executadas.
4. **Incluir objetivos de desempenho:** Descrever objetivos de desempenho para o SoS.
5. **Demonstrar a rastreabilidade entre as funcionalidades e capacidades:** Manter a rastreabilidade mútua entre as capacidades desejadas pelos *stakeholders* e as funcionalidades para atendê-las.
6. **Identificar Problemas:** Identificar eventuais problemas para atender as capacidades e descrevê-los.
7. **Refinar as informações e documentar no SoS ReqSpace:** Refinar as informações registradas e documentar as funcionalidades no documento *SoS Requirements Space*.

O artefato de saída é o documento *SoS Requirements Space*, cujo template está disponível no apêndice B. É importante ressaltar que este documento também pode vir a ser atualizado no decorrer do processo, por meio de tarefas de revisão e atualização.

#### 4.4.4 *Task: Get Initial Requirements from Architecture*

Essa tarefa tem o objetivo de obter os requisitos iniciais necessários para fornecer uma estrutura que permita mudanças nas funcionalidades do SoS. Uma arquitetura de SoS é ela mesma uma geradora de requisitos. Quando os engenheiros de SoS desenvolvem uma arquitetura para o SoS, eles precisam sobrepor os sistemas constituintes de uma forma estruturada para que estes trabalhem em conjunto e compartilhem informações. Em alguns casos, essa estrutura sobreposta poderá ser diferente do design original dos sistemas constituintes, assim algumas mudanças nos sistemas podem ser necessárias para suportar a arquitetura do SoS (DOD-USA, 2008).

É importante ressaltar que a arquitetura pode adicionar requisitos que imediatamente podem não atender aos requisitos funcionais do SoS, mas que podem fornecer necessidades estruturais que irão permitir uma ampliação de funcionalidades no futuro. O responsável pela tarefa é o Engenheiro de Requisitos do SoS, com colaboração dos Engenheiros de Requisitos dos SC. Os passos desta tarefa são os seguintes.

1. **Acessar a arquitetura do SoS:** Após as primeiras iterações no desenvolvimento do SoS, já é possível obter acesso ao projeto da arquitetura.
2. **Acessar as informações sobre os SC:** Com as informações obtidas sobre os sistemas constituintes, o engenheiro de SoS obtém os requisitos necessários para prover a interoperabilidade e a associação consistente dos SC.

3. **Analisar quais os atributos de integração são necessários:** Nem sempre todos os SC serão capazes de se auto relacionar sem a adição de funcionalidades novas, o engenheiro de SoS deve identificar quais atributos e funcionalidades são necessários e especificá-los em forma de requisitos.
4. **Listar os requisitos obtidos a partir da Arquitetura:** Documentar os requisitos identificados no artefato *SoS Initial Architecture Requirements* que servirão como fonte para revisão e atualização dos demais documentos.

Esta tarefa gera como artefato o documento *SoS Initial Architecture Requirements*, cujo template pode ser encontrado no apêndice B, que irá servir como entrada para a tarefa de revisão dos *SoS Capability Objectives*, *SoS Conops* e *SoS Requirements Space*.

#### 4.4.5 **Task: Review SoS CO, SoS Conops and SoS ReqSpace**

Tem o objetivo de revisar e validar os Objetivos de Capacidade, o Conceito de Operações e os requisitos de alto nível. Reuniões entre os *stakeholders*-chaves do projeto são desenvolvidas a fim de revisar e validar os documentos de requisitos.

Essa tarefa pode ser executada diversas vezes, com a responsabilidade do Engenheiro de Requisitos de SoS, e as verificações incluem (SOMMERVILLE, 2011):

1. **Verificação de Validade:** Qualquer conjunto de requisitos é um compromisso firmado com os *stakeholders*, para isso é importante que haja uma análise e uma reflexão aprofundada sobre todos os requisitos indicados como necessários, a fim de validar se sua necessidade é real ou se pode ser atendida de outra forma.
2. **Verificação de Consistência:** Buscar identificar restrições contraditórias ou descrições diferentes de um mesmo objetivo de capacidade ou função desejada.
3. **Verificação de Completude:** Verificar se os documentos incluem requisitos que definem todas os objetivos de capacidade pretendidos pelos usuários do SoS.
4. **Verificação de Realismo:** Verificar os requisitos para assegurar que estes podem ser realmente implementados considerando o orçamento e o cronograma de desenvolvimento.
5. **Revisar os requisitos de interoperabilidade:** Obter os *SoS Initial Architecture Requirements* e identificar as mudanças necessárias para promover a integração consistente dos SC.
6. **Gerar Review Record ou Change Request:** Se as validações apontarem que somente correções pequenas de validade, consistência, completude ou realismo são necessárias,

um registro de revisão é gerado e enviado para atualização dos documentos. Caso algum problema seja identificado, ou uma mudança de maior porte seja necessária, uma requisição de mudança precisa ser enviada para análise mais específica.

Essa tarefa pode gerar um artefato chamado *Change Request* que é geralmente um problema identificado ou uma proposta específica de mudança, que será enviada para o subprocesso *Manage Changes* para ser analisada e avaliada pelo *Change Control Board*. Ou pode gerar um *Review Record* que pode trazer alguma sugestão de correção aos documentos revisados para a tarefa de atualização.

Sendo assim, ao final deste subprocesso, se a tarefa de revisão gerar uma *Review Record* que será enviada ao subprocesso *Manage SoS Requirements* a fim de corrigir pequenos erros ou inconsistências encontrados durante a revisão. Porém, caso a tarefa identifique mudanças significativas nos documentos de SoS e gere uma *Change Request*, esta será enviada ao subprocesso *Manage Changes* para ser analisada de acordo com determinados critérios. Os templates de ambos artefatos estão disponíveis no apêndice B.

#### 4.5 SUBPROCESS 3: MANAGE CHANGES

Com o objetivo de gerenciar as possíveis mudanças nos requisitos que podem surgir ao longo do desenvolvimento evolutivo do SoS, este subprocesso pode iniciar analisando se há alguma mudança nos requisitos dos SC que possam impactar os requisitos de SoS. Ou ainda, pode analisar mudanças nos documentos de requisitos de SoS que precisam ser adaptadas. O fluxo das tarefas é representado no diagrama BPMN na Figura 4.4.

##### 4.5.1 *Task: Identify CS Requirements Changes*

Com o objetivo de identificar mudanças nos requisitos dos Sistemas Constituintes, o SoS precisa se conectar aos processos que gerenciam os requisitos dos seus sistemas constituintes, e identificar as mudanças que são capazes de impactar seus objetivos de capacidade.

A evolução do SoS está diretamente relacionada à evolução de seus sistemas constituintes, por isso é importante manter um registro de todas as mudanças e evoluções que ocorrem nos SC e seus impactos no SoS (DOD-USA, 2008). Os responsáveis por essa tarefa são o Engenheiro de Requisitos do SoS e dos SC. A tarefa segue os seguintes passos.

1. **Obter relatório de mudanças nos SC:** Os engenheiros dos SC são responsáveis (conforme os acordos de colaboração) por informar ao engenheiro de requisitos do SoS, sempre que houver mudanças em seus requisitos individuais.



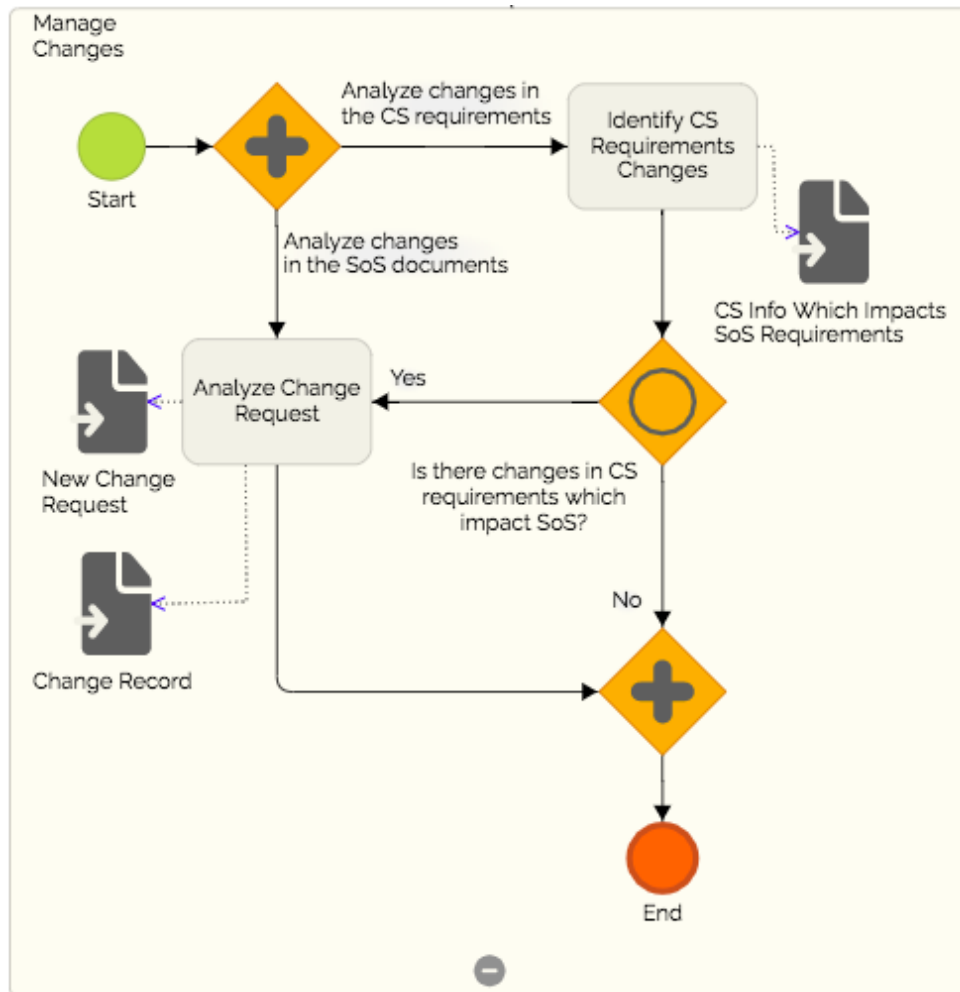


Figura 4.4 – Subprocesso - *Manage Changes*

2. **Avaliar impacto das mudanças:** O engenheiro de requisitos do SoS analisa as mudanças nos requisitos do SC e avalia o impacto das mesmas no SoS.
3. **Documentar as mudanças que impactam o SoS:** O engenheiro de requisitos filtra todas as mudanças que causam impacto no SoS e as documenta para que estas possam ser analisadas posteriormente.
4. **Gerar uma solicitação de mudança, se necessário:** Caso alguma mudança identificada nos requisitos dos SC seja de alto impacto nos requisitos de SoS, é possível gerar uma *Change Request* a ser analisada na tarefa *Analyze Change Request*.

O artefato gerado é um relatório sobre as mudanças nos requisitos dos SC que são capazes de impactar o SoS, chamado de *CS Information Which Impacts SoS Requirements*. Ou uma *Change Request*, caso exista alguma mudança significativa nos requisitos de SoS. Os templates estão disponíveis no apêndice B.

#### 4.5.2 *Analyze Change Request*

O objetivo é analisar problemas e propostas de mudanças nos requisitos. A tarefa começa com a análise de um problema identificado pela *Review SoS CO*, *SoS Conops and SoS ReqSpace*, ou por uma proposta específica de mudança nos requisitos dos SC. Nesse ponto, o *Change Control Board* realiza uma análise do problema ou da proposta, a fim de verificar sua validade.

Se a mudança for validada, avalia-se o efeito da mudança e seu impacto em termos de custos para o projeto. Uma vez que a análise é concluída, a decisão sobre implementar ou não a mudança é tomada. Os responsáveis por essa tarefa são os membros do *Change Control Board* (SOMMERVILLE, 2011). A tarefa segue os seguintes passos:

1. **Identificar o problema ou proposta de mudança:** Identificar o requisito que originou o problema ou a requisição de mudança e suas relações (através das políticas de rastreabilidades estabelecidas).
2. **Validar o problema ou proposta de mudança:** Após análise do problema ou requisição de mudança, verifica-se a sua validade.
3. **Avaliar efeito e impactos da mudança no SoS:** O efeito da mudança é avaliado por meio de informações de rastreabilidade e de conhecimentos sobre os requisitos do SoS.
4. **Estimar custos:** O custo é estimado em termos de modificações nos documentos de requisitos.
5. **Tomar decisão sobre implementação da mudança:** Uma vez que a análise é concluída, toma-se a decisão de implementar a mudança ou não.

Quando uma requisição de mudança é aceita e aprovada, é necessário que todas as modificações nos documentos de requisitos sejam feitas de modo coerente. Para isso, essa tarefa gera como artefato o *Change Record* (template disponível no apêndice B), e envia para tarefa de *Update SoS CO*, *SoS Conops and SoS ReqSpace* todos os dados necessários para implementação da mudança, bem como a descrição de seus impactos no desenvolvimento do SoS.

Caso a mudança proposta não seja aprovada, a análise é transmitida a quem solicitou a mudança, que pode adaptar e gerar uma nova *Change Request*, que será entrada para uma nova análise de mudança, ou retirar a solicitação.

#### 4.6 SUBPROCESSO 4 - MANAGE SOS REQUIREMENTS

Este último subprocesso tem o objetivo de atualizar as versões atuais dos documentos (*SoS Capability Objectives*, *SoS Conops e SoS Requirements Space*) e assim usá-los como base

para definição, priorização e recomendação de requisitos para o próximo incremento do SoS. O fluxo das tarefas está ilustrado no diagrama BPMN da Figura 4.5.

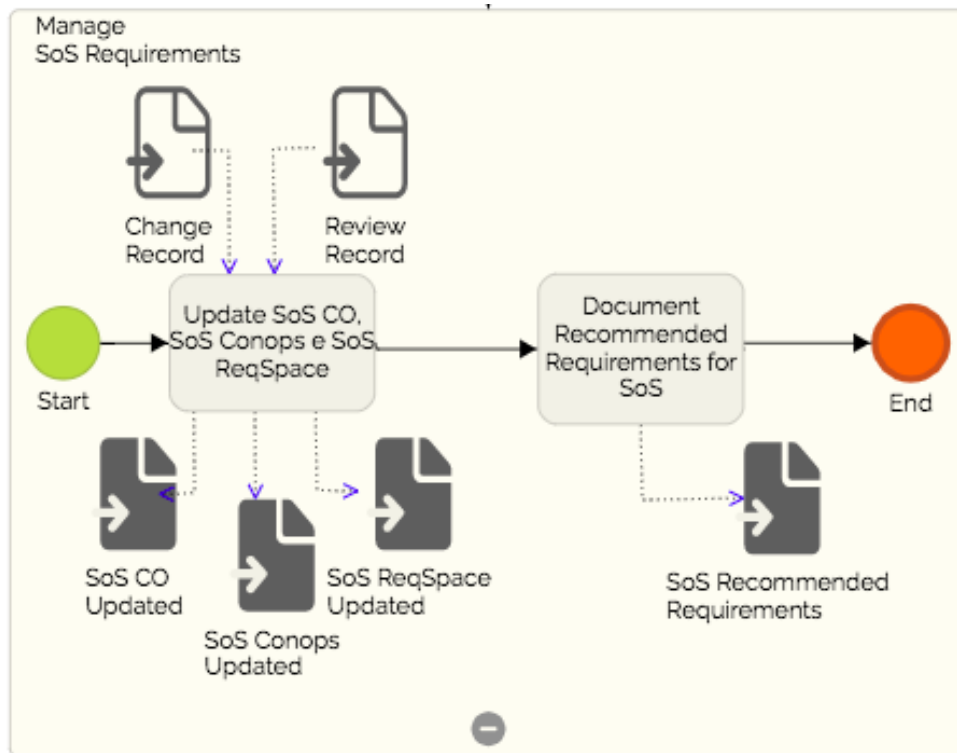


Figura 4.5 – Subprocesso - *Manage SoS Requirements*

#### 4.6.1 *Task: Update SoS CO, SoS Conops and SoS ReqSpace*

Essa tarefa visa atualizar os documentos com base nas correções sugeridas pelo *Review Record* ou incluir e adaptar as mudanças recomendadas pelo *Change Record*. O responsável pela tarefa é o *SoS Requirements Engineer*, que deve seguir os seguintes passos.

1. **Identificar as atualizações a serem efetuadas:** Analisar os artefatos *Change Record* ou *Review Record* e obter as instruções de correção ou mudança.
2. **Revisar o documento a ser atualizado:** Analisar se a correção ou implementação de mudança deve ser feita no *SoS Capability Objectives*, *SoS Conops* ou *SoS Requirements Space*.
3. **Atualizar o documento:** Seguindo os passos de sua atividade específica, fazer as correções ou implementar a mudança em cada documento, conforme necessário.

O resultado dessa tarefa são os documentos *SoS Capability Objectives*, *SoS Conops* e *SoS Requirements Space* atualizados.

#### 4.6.2 *Task: Document Recommended Requirements for SoS*

Em qualquer conjunto de requisitos, alguns serão mais importantes que outros (SOMMERVILLE et al., 2003). Essa tarefa envolve a interação entre os *stakeholders* para descobrir os requisitos mais importantes que serão atendidos no próximo incremento do SoS.

Os engenheiros dos SC trabalham em conjunto com o gerente e engenheiro de requisitos do SoS para revisar, priorizar e recomendar quais requisitos devem ser implementados a cada iteração. Os requisitos selecionados são movidos para *baseline* funcional da iteração. Essa tarefa é de responsabilidade do Engenheiro de Requisitos do SoS, que deve seguir os seguintes passos:

1. **Analisar SoS CO, SoS Conops and SoS ReqSpace:** Analisar os três documentos em conjunto a fim traduzir os objetivos em funcionalidades que possam ser cumpridas pelos SC candidatos.
2. **Analisar RE Work Plan:** Com base no RE *Work Plan* e análise do *Technical Plan* é possível ter ciência das necessidades para o próximo incremento.
3. **Classificar os requisitos:** Classificar os requisitos do SoS em: funcionais, não-funcionais e de interoperabilidade.
4. **Priorizar os requisitos:** Priorizar os requisitos que devem ser atendidos na próxima iteração.
5. **Estabelecer rastreabilidade:** Estabelecer a rastreabilidade entre os requisitos e suas origens.
6. **Recomendar os requisitos:** Recomendar os requisitos escolhidos a serem atendidos no próximo incremento do processo de desenvolvimento do SoS e quando possível sugerir quais os SC que podem atender a estes requisitos.

Como resultado, a tarefa gera o artefato *SoS Recommended Requirements*, que recomenda quais requisitos devem ser atendidos na próxima iteração. Seu template está disponível no apêndice B.

Através da execução destas treze tarefas divididas em quatro subprocessos, é possível identificar os *stakeholders* envolvidos no processo de gerenciamento de requisitos, bem como identificar os objetivos de capacidade do SoS e traduzi-los em requisitos de alto nível assim como validá-los e gerenciar as possíveis mudanças. Os artefatos gerados ou usados com entradas das tarefas são descritos na Seção 4.7.

É importante ressaltar que as características do RMP-SoS remetem à ideia de que o processo é mais adequado para ser aplicado no contexto de SoS dos tipos Reconhecido, Dirigido ou Colaborativo (com algumas adaptações), cujas definições já foram apresentadas no capítulo 2.

#### 4.7 ARTEFATOS

Esta seção descreve a função de cada artefato que faz parte da execução do processo RMP-SoS, alguns deles são produtos gerados externamente e servem como entrada para as tarefas do RMP-SoS, os outros são documentos gerados durante a execução do processo, e seus templates podem ser visualizados no apêndice B.

- *Information About Stakeholders* - Artefato externo de entrada para a tarefa *Identify Stakeholders*, apresenta as informações profissionais de cada pessoa envolvida no processo de gerenciamento de requisitos.
- *Stakeholders Register* - Lista e identifica todos os stakeholders das atividades de requisitos, bem como suas habilidades e conhecimento, níveis de influência e os classifica por ordem de importância.  
Saída de: *Identify Stakeholders*;  
Entrada para: *Assign Roles and Responsibilities* e *Establish Collaboration Agreements*.
- *Stakeholders Roles and Responsibilities* - Apresenta os papéis e responsabilidades alocados a cada *stakeholder* de acordo com seus conhecimentos e habilidades, seguindo o padrão escolhido pelo gerente do projeto.  
Saída de: *Assign Roles and Responsibilities*;  
Entrada para: *Establish Collaboration Agreements* e *Prepare RE Work Plan*.
- *Collaboration Agreements* - Documentos que apresentam acordos de colaboração entre os *stakeholders* dos SC e do SoS sobre o compartilhamento de informações, condições de acesso e restrição das mesmas.  
Saída de: *Establish Collaboration Agreements*;  
Entrada para: *Prepare RE Work Plan* e *Identify CS Requirements Changes*.
- *Requirements Engineering Work Plan* - Neste plano é indispensável que seja identificado o tipo de SoS e seu ambiente operacional, as missões e objetivos do projeto, a proposta de trabalho, as restrições, a política de rastreabilidade de requisitos, e um glossário dos termos e siglas usados nos documentos produzidos.  
Saída de: *Prepare RE Work Plan*;  
Entrada para: *Identify SoS Capability Objectives*, *Develop SoS Conops*, *Define SoS Requirements Space*, *Analyze Change Request* e *Document Recommended Requirements for SoS*.
- *Implementation Options* - Elemento Externo - gerado pelas atividades dos processos do Elemento Central 6 (*Addressing Requirements and Solution Options*).  
Entrada para: *Establish Collaboration Agreements* e *Review SoS CO, SoS Conops and SoS ReqSpace*.

- *Information About Constituent Systems* - Elemento Externo - gerado pelas atividades dos processos do Elemento Central 2 (*Understanding Systems and Relationships*).  
Entrada para: *Establish Collaboration Agreements, Get Initial Requirements from Architecture, Identify CS Requirements Change, Define SoS Requirements Space* e *Develop SoS Conops*.
- *Project General Information* - Artefato externo, informações de cunho geral sobre o escopo do projeto do SoS. Serve de entrada para as tarefas *Prepare RE Work Plan* e *Establish Collaboration Agreements*.
- *SoS Capability Objectives* - Apresenta na linguagem do usuário o que precisa ser feito pelo SoS, ou seja, as missões primárias e secundárias, as restrições operacionais, e uma visão geral do ambiente.  
Saída de: *Identify SoS Capability Objectives* e *Update SoS CO, SoS Conops and SoS ReqSpace*;  
Entrada para: *Develop SoS Conops, Define SoS Requirements Space, Review SoS CO, SoS Conops and SoS ReqSpace, Update SoS CO, SoS Conops and SoS ReqSpace* e *Document Recommended Requirements for SoS*.
- *SoS Conops* - (Conceito de Operações do SoS) - é usado para entender as intenções dos *stakeholders*. Descreve o modo como os usuários planejam operar e usar o SoS para atingir seus objetivos. Inclui a descrição do usuário e resume as necessidades, metas e características da comunidade de usuários do sistema dentro do ambiente operacional.  
Saída de: *Develop SoS Conops* e *Update SoS CO, SoS Conops and SoS ReqSpace*;  
Entrada para: *Define SoS Requirements Space, Review SoS CO, SoS Conops and SoS ReqSpace, Update SoS CO, SoS Conops and SoS ReqSpace* e *Document Recommended Requirements for SoS*.
- *SoS Requirements Space* - Define as funcionalidades necessárias para atender as capacidades desejadas do SoS.  
Saída de: *Define SoS Requirements Space* e *Update SoS CO, SoS Conops and SoS ReqSpace*;  
Entrada para: *Review SoS CO, SoS Conops and SoS ReqSpace, Update SoS CO, SoS Conops and SoS ReqSpace* e *Document Recommended Requirements for SoS*.
- *SoS Architecture* - Arquitetura do SoS produzida pelas atividades dos processos do Elemento Central 4 (*Developing and Evolving SoS Architecture*). É entrada para tarefa *Get Initial Requirements from Architecture*.
- *SoS Initial Architecture Requirements* - Requisitos necessários para prover a interoperabilidade e a associação consistente dos Sistemas Constituintes.

Saída de: *Get Initial Requirements from Architecture*

Entrada para: *Review SoS CO, SoS Conops and SoS ReqSpace*

- *CS Information which Impacts SoS Requirements* - Registro de todas as mudanças e evoluções que ocorrem nos SC e seus impactos no SoS.

Saída de: *Identify CS Requirements Changes*

Entrada para: *Analyze Change Request*

- *Review Record* - Registro de revisão que pode apresentar uma correção ou confirmação de consistência, ambiguidade, validade, completude ou realismo.

Saída de: *Review SoS CO, SoS Conops and SoS ReqSpace*

Entrada para: *SoS Update CO, SoS Conops and SoS ReqSpace*

- *Change Request* - Apresenta um problema identificado durante a revisão que precisa ser corrigido, ou uma proposta específica de mudança a ser analisada.

Saída de: *Review SoS CO, SoS Conops and SoS ReqSpace, Analyze Change Request* ou *Identify CS Requirements Changes*.

Entrada para: *Analyze Change Request*

- *Change Record* - Apresenta todos os dados necessários para implementação da mudança, bem como a descrição de seus impactos no desenvolvimento do SoS.

Saída de: *Analyze Change Request*

Entrada para: *Update SoS CO, SoS Conops and SoS ReqSpace*

- *SoS Recommended Requirements* - Conjunto de requisitos recomendados para serem atendidos em determinada iteração no desenvolvimento evolutivo do SoS.

Saída de: *Document Recommended Requirements for SoS*

Entrada para: Atividades dos processos do Elemento Central 6 (*Addressing Requirements and Solution Options*).

Ressaltamos que alguns artefatos de entrada são oriundos de fontes externas, ou seja, gerado por atividades de outros processos relacionados aos diversos Elementos Centrais da ESoS, e que não possuem uma ligação direta com o processo de Gerenciamento de Requisitos aqui proposto, mas que contribuem para geração dos artefatos deste processo.

## 5 AVALIAÇÃO E VALIDAÇÃO DO PROCESSO

A validação do Processo RMP-SoS foi feita em duas etapas. Na primeira delas, uma Avaliação de Conformidade foi desenvolvida a fim de verificar se o processo está de acordo com as recomendações existentes no âmbito da Engenharia de Sistemas e Engenharia de Requisitos, e esta será descrita e analisada na Subseção 5.1. A segunda etapa, trata de verificar a aplicabilidade do processo em um estudo de caso combinado com uma pesquisa-ação conduzidos no Projeto SIS-ASTROS. A descrição desta etapa, os resultados obtidos e discussão dos mesmos são apresentados na Subseção 5.2.

### 5.1 AVALIAÇÃO DE CONFORMIDADE

Esta seção apresenta a avaliação de conformidade do processo RMP-SoS, conduzida com o objetivo de identificar e explicar como o processo atende as cláusulas especificadas nos padrões internacionais aceitos no âmbito da Engenharia de Sistemas (ISO/IEC/IEEE 15288) e Engenharia de Requisitos (ISO/IEC/IEEE 29148).

#### 5.1.1 ISO/IEC/IEEE 15288 (2011) - *Systems and software engineering — System life cycle processes*

Esta norma estabelece um *framework* comum de descrições de processos, para descrever o ciclo de vida de sistemas criados por humanos, definindo um conjunto de processos e uma terminologia associada. Esses processos podem ser aplicados a qualquer nível de hierarquia na estrutura de um sistema. Em seu anexo G, a norma traz uma aplicação desses processos de ciclo de vida em ambientes de SoS.

A avaliação de conformidade começa pelos *Agreement Processes (G.3.2)*, a norma diz que “estes processos são cruciais porque eles estabelecem os modos de controle de desenvolvimento e operacional entre as organizações responsáveis pelo SoS e pelos SC independentes”. Nesse caso, o RMP-SoS tem uma tarefa chamada *Establish Collaboration Agreements*, que tem o objetivo de gerar acordos de colaboração entre as partes interessadas para expressar o modo (formal ou informal) em que os SC serão associados e como os *stakeholders* irão colaborar entre si, de forma que a independência gerencial e operacional dos SC sejam respeitadas.

Sobre os *Organizational project-enabling processes (G.3.3)*, “em um SoS, os proprietários dos SC geralmente retêm a responsabilidade pela engenharia de seus sistemas assim como cada um deles tem seus próprios processos organizacionais. A organização responsável pelo SoS implementa esses processos em consideração às questões que se aplicam ao SoS



como um todo”. Por exemplo, o RMP-SoS preocupa-se em atender a esse quesito conhecendo todos os *stakeholders* envolvidos nas atividades de requisitos do SoS e distribuindo papéis e responsabilidades adequados a eles através das tarefas *Identify Stakeholders* e *Assign Roles and Responsibilities* respectivamente.

Os *Technical Management Processes (G.3.4)* são aplicados a considerações particulares da ESoS: “planejamento, análise, organização e integração de capacidades de uma mistura de sistemas novos e já existentes em uma nova capacidade de SoS”. Além disso, “a organização responsável pelo SoS deve planejar um ciclo de vida integrado que reconheça as mudanças independentes dos SC, em adição às mudanças iniciadas no SoS em um ciclo de vida que o trate como o sistema de interesse”. Nesse contexto, considera-se que o RMP-SoS atende a essa recomendação, pois o processo usa artefatos produzidos ao longo das iterações do ciclo de vida do SoS. Por exemplo, consideramos que a arquitetura do SoS é ela mesma uma geradora de requisitos, mesmo que seu projeto esteja fora do escopo do RMP-SoS. Além disso, o processo proposto pode ser integrado ao ciclo de vida que considera as mudanças nos SC através da tarefa *Identify CS Requirements Changes* que busca pelas mudanças nos requisitos dos SC que tenham algum impacto nos requisitos do SoS.

Sobre os *Technical Processes (G.3.5)*, “eles preocupam-se com ações técnicas através do ciclo de vida, e alguns deles podem ser relacionados a ambientes SoS”. *Stakeholder needs and requirements definition* foca no SoS em alto nível, mas também considera como as diferentes necessidades dos *stakeholders* para os sistemas individuais que podem levar a restrições no SoS. *System Requirements Definition* para SoS tende a ser definido no nível necessário para satisfazer as necessidades dos *stakeholders* e objetivos de missões, para serem traduzidos em requisitos para os SC, com o SoS fazendo o “papel de *stakeholder*” para novos requisitos dos SC. E por fim, os *Design Definition Process* fornecem informações e dados suficientes para a implementação do SoS.

Esses *Technical Processes* diretamente relacionados a requisitos citados acima, são atendidos em nossas tarefas relacionadas à ERSoS (subprocessos 2 e 4 do RMP-SoS) e a explicação sobre elas esta detalhada na subseção 5.1.2.

### **5.1.2 ISO/IEC/IEEE 29148 (2011) - *Systems and software engineering - Life cycle processes - Requirements engineering***

Essa norma fornece um tratamento unificado dos processos envolvidos na Engenharia de Requisitos ao longo do ciclo de vida de softwares e sistemas. Não há menção específica ou aplicação a SoS nesta norma; portanto, consideramos a aplicação dessas recomendações no contexto de SoS adaptando a terminologia específica.

Em *Stakeholder requirements definition process (6.2)*, a proposta é definir os requisitos para um sistema que possam fornecer os serviços necessários pelos usuários e outros *sta-*

*keholders* em um ambiente definido. Nesse sentido, o RMP-SoS atende a todas as atividades propostas pela norma, pois é possível elicitar os requisitos dos *stakeholders* (*Elicit stakeholder requirements*), começando pela tarefa *Identify Stakeholders*, na qual todos os *stakeholders* são identificados, bem como seus interesses ao longo do ciclo de vida do SoS. E, depois, por meio da tarefa *Identify SoS Capability Objectives*, na qual é possível elicitar os objetivos de capacidades desejados pelos envolvidos, e estabelecer a rastreabilidade dos mesmos.

Para *Define stakeholder requirements* (6.2.3.2), o RMP-SoS usa a tarefa *Develop SoS Conops* para “identificar um conjunto de atividades sequenciais representativo que identifique todos os serviços requeridos que correspondem a cenários e ambientes operacionais antecipados” e “identificar a interação entre os usuários e o sistema”, por meio da descrição do ambiente operacional e de cenários operacionais pela visão dos usuários que irão operar o SoS.

Questões relacionadas ao *Analyze and maintain stakeholders requirements* (6.2.3.3) são atendidas em parte pela tarefa *Review SoS CO, SoS Conops and SoS ReqSpace*, a qual revisa os documentos gerados pelas tarefas anteriormente citadas e garante que não há requisitos “conflitantes, faltantes, incompletos, ambíguos, inconsistentes ou não verificáveis”. Se houver algum problema, a tarefa gera um *Review Record* ou uma *Change Request* para corrigir isto.

Em *Requirements Analysis Processes* (6.3) a proposta é “transformar a visão dos serviços desejados pelos *stakeholders* em uma visão técnica de um produto requerido que possa entregar estes serviços”. RMP-SoS atende a essa recomendação pela definição do espaço de requisitos do SoS. Então, para definir os requisitos de sistema (*Define system requirements - 6.3.3.1*), a tarefa *Define SoS Requirements Space* tem o objetivo de limitar as necessidades primárias dos *stakeholders* e definir quais funções são necessárias para atender a essas necessidades, estabelecendo a rastreabilidade entre capacidades e funcionalidades. *Analyze and maintain system requirements* (6.3.3.2) também é atendida em parte pela *Review SoS CO, SoS Conops and SoS ReqSpace* para garantir que requisitos de sistemas especificados reflitam os objetivos de capacidade adequadamente a fim de atender as necessidades e expectativas dos *stakeholders*.

A norma também enfatiza que existem atividades da ER em outros processos técnicos. Por exemplo, em *Requirements in architectural design* (6.4.1), a definição de uma arquitetura *Definition of the architecture* (6.4.1.1) está fora do escopo do RMP-SoS. Entretanto, ele engloba a *Analyze and evaluate the architecture* (6.4.1.2) de forma que é possível identificar quais requisitos de sistema são necessários para fazer a integração dos SC de forma consistente. Isso ocorre através das tarefas *Get Initial Requirements from Architecture* e *Identify CS Requirements Changes* do RMP-SoS.

Sobre *Requirements in verification* (6.4.2), “a proposta é confirmar que os requisitos de design especificados são atendidos pelo sistema”. No contexto de um SoS, essas questões de verificação devem ser feitas em tempo de execução, e as atividades são de responsabilidade dos processos relacionados aos Elementos Centrais *Assessing Performance to Capability Objectives* e *Addressing Requirements and Solution Options*. O RMP-SoS apenas fornece entradas para planejar e conduzir um plano de verificação de requisitos em tempo de execução. A mesma

suposição é válida para *Requirements in validation* (6.4.3).

Por fim, a cláusula relacionada a *Requirements management* (6.5) fala sobre a importância de “estabelecer processos para definição, controle e publicação dos requisitos em uma *baseline* técnica para todos os níveis do sistema-de-interesse”. O RMP-SoS garante que todas as tarefas para identificar, definir, manter e priorizar requisitos consistentes sejam cumpridas ao longo do ciclo de vida evolutivo do SoS.

E sobre *Change Management* (6.5.2), a norma diz que “mudanças devem ser gerenciadas para garantir que propostas de mudanças passem por uma avaliação de impacto, revisão e processo de aprovação, e aplicando requisitos cuidadosos para rastreamento e gerenciamento de versões”. Essas questões, no RMP-SoS, são tratadas pelas tarefas *Analyze Change Request* e *Update SoS CO, SoS Conops and SoS ReqSpace*. A primeira analisa e avalia a validade e o impacto da mudança nos requisitos do SoS. E a segunda implementa a mudança quando aprovada ou implementa uma correção apontada pela *Review SoS CO, SoS Conops and SoS ReqSpace*.

### 5.1.3 Discussão dos Resultados

Considerando a presente avaliação de conformidade, assumimos que o processo de gerenciamento de requisitos proposto atende a maioria das recomendações sugeridas pelas duas normas internacionais relacionadas a esse tópico. É importante ressaltar que não há uma norma específica desenvolvida diretamente para a Engenharia de SoS. (DAHMAN; ROEDLER, 2016) fala sobre essa falta de padronização da ESoS e que, até agora, o único modo de se obter um padrão é a aplicação de normas já existentes como a ISO/IEC/IEEE 15288.

Portanto, nessa avaliação foram utilizados o Anexo G da ISO/IEC/IEEE 15288 e a ISO/IEC/IEEE 29148 que é especificamente sobre processos da ER para sistemas e softwares. Nesse caso, para torná-la aplicável ao contexto do SoS, adaptamos alguns conceitos e terminologias. Por exemplo, os “requisitos dos *stakeholders*” mencionados na norma são considerados os “objetivos de capacidades” no contexto de um SoS.

Entende-se que alguns dos processos apresentados por esta norma, não são aplicáveis ao escopo deste processo de gerenciamento de requisitos para SoS. Como exemplo, *Measurement for requirements* (6.5.3) que apresenta atividades que podem ser entendidas como parte dos processos relacionados ao elemento *Assessing Performance to Capability Objectives*, mesmo que a ER como uma disciplina possa se beneficiar desses processos, eles não estão diretamente relacionados ao propósito do RMP-SoS.

Além disso, os *Information items* (7), definem três documentos como parte dos processos da ER. Entretanto, em um ambiente SoS, mesmo que sejam similares e tenham quase a mesma proposta, estes documentos têm uma nomenclatura diferente e diferentes modos de serem preenchidos. Por exemplo, o *Stakeholder requirements specification document* seria o documento *SoS Capability Objectives* do RMP-SoS. Da mesma forma, o *System requirements*

*specification document* é a união dos documentos *SoS Conops* e *SoS Requirements Space*.

A Tabela 5.1 apresenta um resumo sobre quais das recomendações são completamente atendidas pelo RMP-SoS, as que são em parte e as que não são atendidas, bem como aquelas que estão fora do escopo do processo proposto.

Cláusula	Sim	Em parte	Não	Fora do Escopo
<b>ISO/IEC/IEEE 15288 (2015) - Annex G (Application of system life cycle processes to a SoS)</b>				
G.3.2 Agreement processes	X			
G.3.3 Organizational project enabling processes		X		
G.3.4 Technical management processes	X			
G.3.5 Technical processes		X		
<b>ISO/IEC/IEEE 29148 (2011) - Processes (Clause 6)</b>				
6.2 Stakeholder requirements definition process (6.2.3 Activities and Tasks)				
6.2.3.1 Elicit stakeholder requirements	X			
6.2.3.2 Define stakeholder requirements	X			
6.2.3.3 Analyze and maintain stakeholder requirements		X		
6.3 Requirements Analysis process (6.3.3 Activities and tasks)				
6.3.3.1 Define system requirements	X			
6.3.3.2 Analyze and maintain system requirements		X		
6.4 Requirements engineering activities in other technical processes (6.4.1 Requirements in architectural design)				
6.4.1.1 Define the architecture				X
6.4.1.2 Analyze and evaluate the architecture		X		
6.4.2 Requirements in verification		X		
6.4.3 Requirements in validation	X			
6.5 Requirements Management				
6.5.2 Change Management	X			
6.5.3 Measurement for requirements				X

Tabela 5.1 – Resumo da Avaliação de Conformidade

## 5.2 VALIDAÇÃO VIA ESTUDO DE CASO E PESQUISA-AÇÃO

Validar um novo processo ou uma nova abordagem não é uma tarefa trivial, pois esta visa desenvolver evidências convincentes de que o modelo proposto pode ser seguido e que ele apresenta etapas, atividades e procedimentos que sejam de interesse dos envolvidos no desenvolvimento do software ou do sistema (COOK; WOLF, 1999). Com intuito de desenvolver estas evidências, esta subseção apresenta o detalhamento da validação executada neste trabalho.

Com base na revisão de literatura identificou-se um problema comumente encontrado na Engenharia de Requisitos para SoS, este problema, através de análise de documentos, também foi identificado no Projeto SIS-ASTROS. Através desta validação objetiva-se verificar a realidade do problema e se a solução proposta é válida neste contexto. O método utilizado para esta validação foi uma abordagem mista a partir da combinação de um estudo de caso confirmatório com uma pesquisa-ação (já definidos no Capítulo 3) e a coleta de dados foi efetuada em uma entrevista e discussão em grupo. A identificação do estudo, as etapas da validação e os resultados obtidos são apresentados a seguir.

### 5.2.1 Identificação do Estudo:

- *Caso de Estudo:* Projeto SIS-ASTROS (Sistema Integrado de Simulação ASTROS).
- *Unidade de Análise:* Engenharia de Requisitos do Projeto SIS-ASTROS.
- *Dono do problema:* Equipe de desenvolvimento do projeto (UFSM), representada neste estudo pela coordenadora e por três pesquisadores do projeto.

#### 5.2.1.1 O Projeto SIS-ASTROS

O Projeto SIS-ASTROS é um projeto desenvolvido pela UFSM em parceria com o Exército Brasileiro (EB), que teve início no ano de 2014 e tem previsão de conclusão em 2019. O Programa Estratégico ASTROS 2020 (PrgEE ASTROS 2020) tem como objetivo modernizar a tecnologia de lançamento de foguetes existente (veículos lançadores, de planejamento e apoio, entre outros), dispor de sistema de Lançamento de Mísseis Táticos de Cruzeiro e munições guiadas de precisão e a implantação do Forte Santa Bárbara em Formosa, GO (UFSM, 2014).

Dentro desse contexto, é de extrema importância a criação de uma especificação global do Sistema Integrado de Simulação do Programa Estratégico ASTROS 2020 – SIS-ASTROS - para que ocorra uma adequada e oportuna inserção da artilharia de mísseis e foguetes na estrutura da Força Terrestre. O Sistema Integrado de Simulação engloba a pesquisa e o desen-

volvimento de diferentes simuladores, sendo que há necessidade de coordenação e interface dos novos simuladores entre si e com aqueles simuladores já em uso pelo Exército (UFSM, 2014).

É importante ressaltar que de acordo com os documentos de requisitos do Projeto SIS-ASTROS, o sistema como um todo é considerado uma Federação de Sistemas (*FoS - Federation of Systems*). Portanto, para fins de aplicação e estudo, este trabalho considera-o como um *System of System* com base na definição dada pelo *Guide to the Systems Engineering Body of Knowledge* - (SEBOK, 2016), que diz que uma Federação de Sistemas é “um SoS que apresenta as seguintes três dimensões de forma muito significativa: autonomia, heterogeneidade e dispersão. E onde cada sistema componente escolhe por vontade própria se irá participar da federação ou não, ou seja, é uma coalizão de vontade”. Neste caso, o SIS-ASTROS também pode ser considerado um SoS do tipo Colaborativo, o qual os sistemas constituintes interagem entre si voluntariamente para cumprir os objetivos centrais acordados (DOD-USA, 2008).

O SIS-ASTROS é constituído por diferentes simuladores novos e já em uso pelo EB, como mostra a Figura 5.1, que representa a arquitetura inicial projetada para o SoS. Apesar de ser descrita em um nível de detalhamento simplificado, a arquitetura permite identificar os principais elementos de simulação envolvidos no SIS-ASTROS. Além de identificar os sistemas constituintes do sistema de simulação, a arquitetura identifica a plataforma de integração a ser utilizada no desenvolvimento das simulações conjuntas, ou seja, no modo de operação integrado.

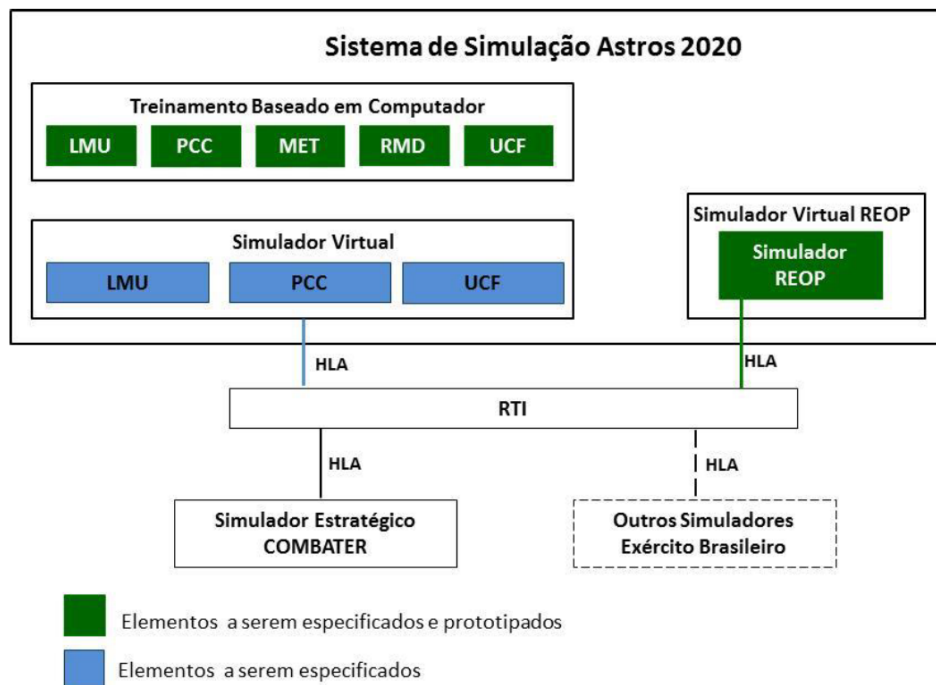


Figura 5.1 – Arquitetura do SIS-ASTROS

Os simuladores que compõem o Sistema Integrado de Simulação ASTROS são:

- **Simulador Tático (SVTát REOP):** Tem como objetivo adestrar militares em doutrinas

táticas relativas ao emprego de uma bateria ASTROS. Consiste de uma ferramenta de simulação virtual que apresenta a capacidade de simular as atividades de um exercício de REOP (Reconhecimento, Escolha e Ocupação de Posição) (UFSM, 2016).

- **Simuladores Virtuais Técnicos (SVTéc):** Consistem em ferramentas de simulação virtual com objetivo de apresentar a capacidade de simular as atividades do uso das seguintes viaturas do Sistema ASTROS: LMU, PCC, VCC e UCF. Esta capacidade será aproveitada tanto para o uso dos simuladores de forma individual quanto em conjunto, bem como integrada com demais elementos do SIS-ASTROS (UFSM, 2017).
- **Simulador Estratégico COMBATER:** Simulador construtivo para exercícios de simulação, desenvolvido a partir do software francês *Sword*, criado pela empresa *RustCon*. O Combater pode ser utilizado em exercícios de nível Unidade, Brigada e Divisão. Permite simular operações de combate nos diversos ambientes operacionais do território brasileiro em exercícios que duram vários dias (TECNODEFESA, 2016).

### 5.2.2 Etapa 1: Observação, leitura e análise de documentos

A primeira etapa deste estudo consistiu em uma leitura e análise dos documentos de requisitos que já haviam sido produzidos pela equipe do projeto SIS-ASTROS usando as técnicas da Engenharia de Requisitos tradicional. Enfatizamos que estes documentos são de uso restrito do Projeto SIS-ASTROS e seu conteúdo não será reproduzido neste trabalho. Os documentos analisados foram:

- Projeto do Sistema de Simulação ASTROS 2020 (SIS-ASTROS) / EB - UFSM;
- Documento “Visão do Projeto” relativo ao Simulador Virtual Tático de REOP do Projeto SIS-ASTROS;
- Documento “Visão dos Simuladores Virtuais Técnicos” relativo aos Simuladores Virtuais Técnicos do Projeto SIS-ASTROS;
- Documento “Requisitos Técnicos Básicos para o Simulador Virtual Técnico da VC-C/PCC”;
- Documento “Requisitos Técnicos Básicos RTB para o Simulador Virtual Técnico da LMU”;
- Documento “Requisitos Técnicos Básicos para o Simulador Virtual Técnico da UCF”;
- Documento “Plano de Integração entre Simulador Virtual Técnico e Simulador Virtual Tático”.

Além disso, houve observação de duas reuniões que ocorreram entre a equipe de desenvolvimento UFSM e membros da RC Consultoria, parceira do projeto, que tratava sobre questões de integração entre o Simulador Tático REOP e o Simulador Construtivo COMBATER.

### 5.2.3 Etapa 2: Identificação do problema e verificação da realidade

Durante a análise feita na Etapa 1, observou-se que os documentos produzidos tinham uma certa limitação no que diz respeito à especificação de requisitos do SIS-ASTROS no modo integrado, dessa forma o seguinte problema foi identificado:

**Problema:** O SIS-ASTROS compreende diferentes simuladores gerenciados e desenvolvidos independentemente, mas que podem atuar no modo integrado entre si e com outros simuladores de propriedade do EB já existentes. A definição dos requisitos de cada simulador constituinte foi feita seguindo técnicas da Engenharia de Requisitos tradicional por meio da especificação dos “Documentos Visão”. Entretanto, usando a ER tradicional com foco nos simuladores de forma independente (sistemas constituintes), se considerarmos o SIS-ASTROS como um *System of System*, as missões, os objetivos de capacidade e os requisitos específicos ao SIS-ASTROS integrado não podem ser plenamente especificados por meio destes documentos.

Este problema foi reconhecido como real pelos donos do problema (equipe de desenvolvimento da UFSM) e estes enfatizaram que sentiram a necessidade de haver uma especificação de requisitos do SIS-ASTROS, ou seja do modo integrado, pois não houve essa abordagem de visão geral desde o início do projeto, e sim uma visão específica de cada simulador constituinte, para depois se pensar no modo integrado. Além disso, os participantes da pesquisa apontaram que se houvesse uma melhor distribuição de papéis e responsabilidades entre os diversos *stakeholders*, a execução das tarefas e desenvolvimento dos produtos seria efetivamente melhor.

Assim, a partir da identificação do problema e do reconhecimento de sua autenticidade pela equipe, foi formulada a seguinte **hipótese**: O Processo para Gerenciamento de Requisitos de *System of Systems* (RMP-SoS) aplicado ao contexto do projeto SIS-ASTROS é capaz de auxiliar na melhor organização do trabalho em equipe e gerência dos requisitos de integração, objetivos e missões do SIS-ASTROS operando em modo integrado.

### 5.2.4 Etapa 3: Aplicação do processo RMP-SoS no contexto do projeto SIS-ASTROS

Na prática o Processo RMP-SoS foi aplicado pela autora deste trabalho, com o auxílio da Coordenadora do Projeto SIS-ASTROS. A aplicação se deu por meio de quatro iterações do processo, via preenchimento e revisão dos documentos, seguindo as orientações disponibilizadas no *website* do RMP-SoS ([www.rmpos.com](http://www.rmpos.com)) e nos templates (disponíveis no apêndice B). É



válido destacar que a aplicação foi desenvolvida entre os meses de outubro de 2017 e fevereiro de 2018, conforme os recursos de documentação do projeto foram sendo disponibilizados. A descrição das iterações segue o modelo: *Tarefa (Versão do artefato gerado)*.

Na primeira iteração que foi desenvolvida entre os dias 05/10/17 a 29/10/17, o processo seguiu o seguinte fluxo de trabalho: *Identify Stakeholders (Stakeholders Register v.1.0) > Assign Roles and Responsibilities (Stakeholders Roles and Responsibilities v.1.0) > Prepare RE Work Plan (RE Work Plan v.1.0) > Identify SoS Capability Objectives (SoS Capability Objectives v.1.0) > Develop SoS Conops (SoS Conops v.1.0)*.

Já na segunda iteração, desenvolvida entre os dias 01/11/17 a 24/11/17, o fluxo das tarefas foi o seguinte: *Identify Stakeholders (Stakeholders Register v.1.1) > Assign Roles and Responsibilities (Stakeholders Roles and Responsibilities v.1.1) > Prepare RE Work Plan (RE Work Plan v.1.1) > Identify SoS Capability Objectives (SoS Capability Objectives v.1.1) > Develop SoS Conops (SoS Conops v.1.1) > Define SoS Requirements Space (SoS Requirements Space v.1.0) > Review SoS CO, SoS Conops and SoS ReqSpace (Review Record v.1.0)*.

A terceira iteração aconteceu entre os dias 03/12/17 e 21/01/18 seguindo o seguinte fluxo: *Establish Collaboration Agreements (Collaboration Agreements) > Update SoS CO, SoS Conops and SoS ReqSpace (SoS Capability Objectives v.1.2, SoS Conops v.1.2, SoS Requirements Space v.1.1) > Identify Stakeholders (Stakeholders Register v.1.2) > Assign Roles and Responsibilities (Stakeholders Roles and Responsibilities v.1.2) > Establish Collaboration Agreements (Collaboration Agreements) > Prepare RE Work Plan (RE Work Plan v.1.2) > Get Initial Requirements from SoS Architecture (SoS Initial Architecture Requirements v.1.0) > Review SoS CO, SoS Conops and SoS ReqSpace (Review Record v.1.1)*.

E a quarta e última iteração, entre os dias 05/02/18 e 23/02/18: *Prepare RE Work Plan (RE Work Plan v.1.3) > Get Initial Requirements from SoS Architecture (SoS Initial Architecture Requirements v.1.1) > Update SoS Co, SoS Conops and SoS ReqSpace (SoS Capability Objectives v.1.3, SoS Conops v.1.3, SoS Requirements Space v.1.2) > Document Recommended Requirements for SoS (Recommended Requirements for SoS v.1.0)*.

É importante ressaltar que as iterações não acompanharam o ciclo de desenvolvimento do projeto SIS-ASTROS em tempo real, pois a pesquisa iniciou quando o projeto se encontrava em um estágio já avançado de desenvolvimento, neste caso houve uma adaptação dos cenários de iteração para que o processo RMP-SoS pudesse ser aplicado de forma coerente.

#### **5.2.5 Etapa 4: Análise da Aplicação e Apresentação para Equipe**

Durante reunião realizada no dia 23 de março de 2018, com participação de quatro profissionais, entre eles um doutor e três mestres, integrantes da equipe de desenvolvimento do Projeto SIS-ASTROS, foram relatadas as etapas do estudo de caso e pesquisa-ação e apresentada a forma como o RMP-SoS foi aplicado ao contexto do projeto.

Os participantes da pesquisa tiveram acesso às últimas versões dos documentos produzidos durante a aplicação do processo, bem como puderam observar o *website* do RMP-SoS e entender o fluxo de trabalho desenvolvido durante as iterações desta pesquisa-ação. Após esta apresentação os participantes tiveram a oportunidade de esclarecer dúvidas e fazer algumas considerações sobre os documentos produzidos. Os resultados da avaliação são apresentados na próxima subseção.

### 5.2.6 Etapa 5: Avaliação do Estudo de Caso e Pesquisa-ação

Existem diferentes métodos e critérios para a avaliação de pesquisas deste tipo, no caso da pesquisa-ação, considera-se que a relevância da prática seja mais importante do que o rigor metodológico, por isso na falta de metodologias de avaliação e coleta de dados específicos para esse tipo de pesquisa, os autores geralmente desenvolvem ou adaptam seus próprios *frameworks* de avaliação baseados na sua prática (KHAN; TZORTZOPOULOS, 2016).

Neste trabalho usamos como referência o *framework* criado por (KHAN; TZORTZOPOULOS, 2016) para definição de um roteiro de entrevista adaptado para a coleta de dados com os participantes da pesquisa para avaliação da mesma. O roteiro é composto por 14 questões abertas e flexíveis conforme o andamento da discussão, e estas são usadas com a intenção de obter sugestões de melhorias e avaliar os seguintes critérios:

- **Corretude:** do diagnóstico inicial, da ação tomada e da forma como a ação foi tomada.
- **Usabilidade:** adequação da ação para a situação, facilidade de uso da ação em termos de simplicidade e clareza e adaptação da ação à prática atual.
- **Efetividade:** habilidade da ação para alcançar os resultados desejados, habilidade da ação para inspirar confiança em seu valor para a prática.

A íntegra do roteiro pode ser visualizada no apêndice C, as respostas dos participantes foram registradas por meio de uma gravação de áudio e transcritas posteriormente. Os resultados da avaliação são descritos e analisados a seguir.

#### 5.2.6.1 Corretude

Quanto à corretude do diagnóstico inicial, os pesquisadores foram questionados se “o diagnóstico sobre as limitações associadas à ER tradicional desenvolvida no Projeto SIS-ASTROS foi preciso”. Todos os participantes concordaram que sim, o diagnóstico foi preciso e acrescentaram alguns comentários sobre as dificuldades encontradas na etapa de especificação de requisitos: “A ER tradicional não se preocupa com todos os elementos que a ERSoS considera:

*os objetivos de capacidade, as metas do SoS, pois ela se preocupa somente com os requisitos dos SC de forma limitada” e “houve dificuldade na interação entre os diversos stakeholders do SIS-ASTROS por estes serem de contextos muito diferentes: Exército, empresa privada e Universidade”.*

Quanto à corretude da ação tomada, foi questionado se os pesquisadores concordavam que “a engenharia de requisitos do SIS-ASTROS seria complementada de forma eficiente com o uso do RMP-SoS”. Os participantes concordaram que a eficiência da ER do SIS-ASTROS seria beneficiada com o uso do RMP-SoS, quando dizem que *“Eu acho que sim, seguindo essas tarefas seria mais fácil fazer a especificação dos requisitos. Já começando pelo gerenciamento dos stakeholders melhoraria bastante”*. E também concordam que o RMP-SoS poderia ser realmente implantado no contexto do SIS-ASTROS, quando questionados se *“o RMP-SoS seria facilmente adaptável ao contexto de desenvolvimento do Projeto SIS-ASTROS”*. Entretanto, os participantes ressaltam que atualmente não haveria tempo hábil para esta implementação, nesse caso eles dizem que: *“Se o processo fosse seguido à risca, seria fácil de ser usado dentro do projeto”*.

#### 5.2.6.2 Usabilidade

Quanto à usabilidade, sobre a “facilidade de uso do RMP-SoS em termos de simplicidade e clareza das tarefas e da organização delas”, os pesquisadores respondem que as informações e instruções disponibilizadas tanto no *website* do processo quanto nos templates dos documentos são úteis e de fácil entendimento: *“eu gostei desse cabeçalho em todos os templates que diz o que cada seção precisa conter” e “com a visualização dos diagramas BPMN o entendimento fica mais claro”*. E sugerem que *“talvez indicar no documento, de qual tarefa esse artefato é entrada, além de qual é saída também seria interessante”*.

Os critérios de adequação da ação para a situação e adaptação da ação à prática atual também podem ser atendidos pela questão “se O RMP-SoS seria facilmente adaptável ao contexto de desenvolvimento do Projeto SIS-ASTROS”, já mencionada anteriormente e respondida positivamente pelos participantes.

Ainda quanto à usabilidade e facilidade de uso, quando questionados se “a comunicação e sincronização entre as tarefas do RMP-SoS por meio dos artefatos (documentos) gerados é eficaz e de fácil entendimento”, os pesquisadores afirmam que sim, que *“as matrizes de rastreabilidade facilitam o entendimento e identificação das dependências e impacto dos requisitos uns nos outros e que o uso dessas políticas de rastreabilidade no contexto atual do projeto, teria facilitado muito o trabalho na construção dos documentos já produzidos”*. E como mencionado anteriormente, sugerem que a indicação de qual tarefa determinado documento serve como entrada também seria uma informação útil (no próprio template) para entender melhor estas relações entre as tarefas.

### 5.2.6.3 Efetividade

Quanto à habilidade da ação para alcançar os resultados desejados, perguntou-se aos participantes se as “tarefas primárias (*Identify SoS CO, Develop SoS Conops, Define SoS ReqSpace, Get Initial SoS Architecture Requirements e Document Recommended Requirements for SoS*)” cumpriam o objetivo de especificar os requisitos do SIS-ASTROS de forma coerente e eficaz”. Os participantes respondem que os documentos gerados são completos, e que em conjunto permitem especificar todos os requisitos de forma coerente e o entendimento é fácil. Entretanto, eles apontam como ponto fraco, uma repetição de informações que em alguns casos, por uma falta de cuidado, poderia causar redundância entre os requisitos. Por exemplo, “*o campo restrições aparece mais de uma vez em documentos diferentes*”.

No que diz respeito às “tarefas de articulação (*Identify CS Requirements Change, Review SoS CO, SoS Conops and SoS ReqSpace; Update SoS CO, SoS Conops and SoS ReqSpace; Identify CS Requirements Changes e Analyze Change Request*)” e se elas cumpriam o objetivo de manter a consistência dos requisitos do SIS-ASTROS, bem como de incluir e adaptar mudanças conforme seu desenvolvimento evolui”, os participantes não puderam opinar de forma completa, pois nesta aplicação não foram identificadas mudanças significativas para execução das tarefas do subprocesso “*Manage Changes*”. Entretanto, os participantes concordaram que as tarefas de gerenciamento de mudanças, bem como as de revisão e atualização são de extrema importância na execução do processo.

Ainda sobre a habilidade da ação para alcançar os resultados desejados, questionou-se se “a distribuição de responsabilidades sugeridas pelo RMP-SoS cumpre o objetivo de organizar o trabalho e a interação entre os *stakeholders* de forma eficiente”, os participantes acreditam que essa distribuição de responsabilidades por tarefa sugerida pelo RMP-SoS é interessante, porém se ela pudesse ser flexível a cada realidade diferente de cada projeto, facilitaria em situações onde determinados tipos de *stakeholders* não existem ou não estão disponíveis.

Sobre se “O RMP-SoS fornece um controle no fluxo de trabalho da Engenharia de Requisitos do SIS-ASTROS que a Engenharia tradicional não foi capaz de fornecer”, os participantes declaram que “*observando os documentos Visão criados no SIS-ASTROS, estes não proporcionam uma visão de alto nível, uma visão top-down, que poderia ter sido discutida no início do projeto*”, portanto o RMP-SoS é útil nesse sentido pois é “*um processo sistematizado que proporciona essa visão que poderia ter facilitado a definição de requisitos do SIS-ASTROS*”.

E por fim sobre a habilidade da ação para inspirar confiança em seu valor para a prática, os participantes foram questionados se “a aplicação do RMP-SoS poderia encorajar ações ou mudanças a serem tomadas no contexto da ER do SIS-ASTROS” e se “a aplicação do RMP-SoS produziu novos conhecimentos para a equipe de ER do SIS-ASTROS que podem vir a influenciar na mudança da prática”. Eles responderam que se ainda houvesse tempo hábil de projeto para implementação das tarefas propostas pelo processo, teria sido uma abordagem in-

interessante. E que eles recordam de situações em que se já tivessem o conhecimento dessa visão proposta pelo RMP-SoS, poderiam ter desempenhado ações de forma diferente na elicitação de requisitos e com um esforço menor.

Os pesquisadores concluem que “*é possível que em situações futuras de evolução do projeto, o processo possa ser implementado na prática*” e que “*este estudo proporcionou novos conhecimentos para a equipe, e uma visão diferenciada do processo de gerenciamento de requisitos*”.

#### 5.2.6.4 Avaliação geral e sugestões

Por fim, os pesquisadores integrantes da equipe de desenvolvimento do SIS-ASTROS apontaram como pontos fortes do RMP-SoS: “*toda a engenharia de requisitos é bem atendida, e muitos pontos importantes do ponto de vista do desenvolvedor são bem especificados e se o processo fosse seguido à risca, seria de grande valia para o projeto*”. Além disso, eles declaram que “*as políticas de rastreabilidade estabelecidas facilitam a implementação de mudanças e a avaliação do impacto delas*”.

Como pontos-fracos, a equipe indica que: “*a repetição de informações em alguns documentos pode vir a causar redundâncias nos requisitos*” e ainda que “*os templates dos documentos Stakeholders Register e Stakeholders Roles and Responsibilities deixam uma certa dúvida sobre o objetivo de cada um e sobre a diferença de significados entre papéis e responsabilidades, que talvez eles pudessem ser unificados*”. Um outro ponto fraco já mencionado anteriormente é que o processo proposto não é completamente adaptável a todos os tipos de SoS e a diferentes contextos de projetos.

Sobre quais seriam as barreiras para adoção do RMP-SoS de forma efetiva no gerenciamento de requisitos do SIS-ASTROS, eles indicam que, na prática, “*seria difícil de seguir à risca todas as tarefas e orientações devido à diversidade de stakeholders*”, e que haveria uma maior dificuldade na distribuição de papéis e responsabilidades entre os *stakeholders*, pois é complicado ter estes espaços de discussão sobre requisitos com participação de todos. Além disso, enfatizam que a complexidade a nível de SoS é bem maior do que a nível de sistemas constituintes, e isso também seria uma barreira.

Sobre os aspectos do RMP-SoS que tornam sua aplicação fácil, os pesquisadores apontam que o “*acesso ao website com todas as instruções e disponibilidade de diagramas de fluxo e templates torna a prática mais clara*”, o que tornaria a sua adoção fácil dentro do contexto do SIS-ASTROS.

Finalmente, como sugestões de melhoria, os participantes sugerem que uma ferramenta para gerenciamento dos documentos seria interessante, assim como se fosse possível simplificá-los e torná-los mais objetivos, isso também seria um ponto a melhorar. E como já mencionado, sugerem que sejam criadas guias ou orientações para que este processo possa ser aplicado e

adaptado a diferentes contextos e realidades de projetos.

#### 5.2.6.5 *Discussão e Análise dos Resultados da Avaliação*

A validação da pesquisa por meio deste estudo de caso e pesquisa-ação, teve como objetivo avaliar a autenticidade do problema e se a hipótese de resolução deste problema era válida, por meio da aplicação do Processo RMP-SoS no contexto do Projeto SIS-ASTROS. Os resultados da coleta de dados, desenvolvida por meio de uma entrevista com os participantes, foram considerados satisfatórios, pois de modo geral provam que o objetivo do estudo foi atingido.

Em um primeiro momento, os participantes confirmaram a autenticidade do problema identificado por esta pesquisa, e complementaram incluindo alguns relatos de outras limitações que a adoção das técnicas da ER tradicional trouxeram ao projeto. Num segundo momento, a avaliação teve o objetivo de identificar se a prática, ou seja, a aplicação do processo RMP-SoS no projeto SIS-ASTROS, atendia a determinados critérios como corretude, usabilidade e efetividade.

Em termos de corretude de diagnóstico, tratamento e execução da ação tomada, a avaliação foi muito positiva, pois os participantes concordaram que o Gerenciamento e a Engenharia para Requisitos de SoS (através do RMP-SoS), diferente da ER tradicional, proporciona uma visão mais ampla e mais precisa na definição de requisitos. E ainda que tarefas de gerenciamento de *stakeholders* e de suas responsabilidades são muito importantes para um melhor andamento do projeto.

Quanto à usabilidade, a avaliação também foi positiva, e obteve algumas sugestões importantes. Os participantes responderam que o RMP-SoS é de fácil entendimento, e destacam como pontos fortes as instruções disponibilizadas nas tarefas (através do *website*) e nos templates, assim como os recursos visuais (diagramas e matrizes de rastreabilidade) que no ponto de vista deles tornam a prática mais fácil e mais realista. Os participantes sugeriram que seria importante indicar nos cabeçalhos dos templates de qual tarefa aquele artefato pode ser entrada, no entanto essa informação pode ser visualizada no *website*, junto a cada link de *download* do template.

Sobre a efetividade da aplicação, todos concordam que o RMP-SoS cumpre com seus objetivos de definir os requisitos para o SoS e organizar o trabalho dos *stakeholders* de forma coerente e de acordo com seus papéis e responsabilidades. Entretanto, os participantes também destacam que há uma necessidade de adaptação desses recursos a contextos e realidades de projeto específicas, que se o RMP-SoS sugerisse guias de adaptação das tarefas, seria um ponto a melhorar.

Entende-se que a avaliação da pesquisa cumpriu com seus objetivos. Algumas situações não puderam ser avaliadas na prática, por exemplo o gerenciamento de mudanças e a comunicação entre *stakeholders* para cumprimento das tarefas do processo. Entretanto, de modo geral,

o estudo de caso e a pesquisa-ação proporcionaram um melhor entendimento da prática e levantaram algumas limitações e necessidades de melhoria para o processo em trabalhos futuros. Por exemplo, durante a etapa 3, observou-se que nem sempre todas as instruções podem ser seguidas à risca, e que os templates propostos muitas vezes precisam ser adaptados ou modificados de acordo com determinado contexto ou situação. Para a equipe de desenvolvedores do projeto SIS-ASTROS, o estudo também foi considerado positivo, pois contribuiu com a formação de novos conhecimentos e trouxe uma visão diferenciada para a engenharia e gerenciamento de requisitos que antes não havia sido pensada.

### 5.2.7 Lições Aprendidas

O estudo de caso reportado neste capítulo contribuiu para identificar um conjunto de lições que são descritas a seguir.

- Lição Aprendida 1: Os templates devem ser flexíveis e adaptáveis ao contexto do projeto de SoS em que o processo está sendo aplicado.
- Lição Aprendida 2: Clareza, simplicidade e objetividade nas instruções para realização de tarefas são fundamentais.
- Lição Aprendida 3: Recursos visuais (como diagramas de fluxo ou de atividade) melhoraram o entendimento da equipe e por consequência facilitam a execução do processo.
- Lição Aprendida 4: Recursos visuais como matrizes de rastreabilidade facilitam o trabalho da equipe de ERSoS no gerenciamento de mudanças de requisitos.
- Lição Aprendida 5: O *Website* e os templates facilitam o uso e aplicação do processo.

## 6 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar uma revisão dos trabalhos relacionados ao tópico de Engenharia de Requisitos para SoS. Em geral, os estudos relacionados propõem métodos, processos ou ferramentas para suportar a engenharia de requisitos ou a engenharia de capacidades no contexto de *System of Systems*.

### 6.1 ENGENHARIA DE REQUISITOS PARA SOS

(LEWIS et al., 2009) observa que a Engenharia de Requisitos para sistemas tradicionais, por mais que ainda proporcione inúmeros desafios aos engenheiros, já foi amplamente pesquisada e muitas técnicas com sucesso já foram propostas. Nesse caso, os autores fazem uma análise de diversas técnicas de ER existentes e propõem uma aplicação dessas técnicas em um processo de ER para SoS. De forma resumida, a abordagem proposta considera aspectos *top-down* e *bottom-up* e incorpora as seguintes atividades da ER tradicional: identificar o contexto do SoS; identificar os objetivos do SoS e dos sistemas individuais; entender as interações do SoS; identificar as capacidades e restrições dos sistemas individuais; e analisar a lacuna.

Os autores concluem que apesar de técnicas da ER tradicional não serem completamente aplicáveis a SoS, não seria inteligente ignorá-las por completo na construção de novas técnicas específicas para ER de SoS. Entretanto, a conclusão do trabalho diz que o estudo poderia ser complementado com outras técnicas que funcionem melhor em um ambiente descentralizado e multi-organizacional.

### 6.2 UM MÉTODO PARA DEFINIR REQUISITOS DE SOS

Em um estudo publicado em 2014, os autores (WALKER et al., 2014) examinam a natureza das questões relacionadas à definição de requisitos de SoS, discutem e demonstram uma metodologia emergente. O método promete ser adaptável para diferentes domínios de SoS e melhorar as práticas de ER para SoS significativamente.

O método combina abordagens *top-down* e *bottom-up* para discernir e derivar requisitos de SoS unificados e é formado por sete componentes: 0. *Bounding the SoS*; 1. *Synthesizing characteristics*; 2. *Aggregation of capability objectives*; 3. *Extraction of functions*; 4. *Comparison of functional themes*; 5. *Theme review*; 6. *Derivation of requirements*. Como resultado final é gerado um conjunto de requisitos considerado “mais granular” do que os objetivos de capacidades de alto-nível e “mais abstrato” do que requisitos de sistemas detalhados.



### 6.3 CAPACIDADES DE SOS PARA ENGENHARIA DE REQUISITOS

(LANE, 2014) fornece diretrizes adicionais para tradução dos objetivos de capacidade em requisitos, definindo métodos, processos e ferramentas para ESoS que possam suportar essa atividade e ilustra como estes podem ser usados e integrados no contexto de SoS. A autora cita os seguintes métodos, processos e ferramentas: *System modelling language object models*, *Responsibility/Dependability modelling*; *Interoperability matrices*; *Data fusion analyses*; *Level of performance, cost and ROI* e *RACSRS analysis using C2R MPTs*.

A pesquisa sugere que os métodos, processos e ferramentas já existentes podem ser facilmente propostos e usados em conjunto para suportar a engenharia de capacidades e requisitos de SoS, resultando em uma análise técnica bastante rigorosa das opções de capacidade e dos custos necessários para sua implementação.

### 6.4 UM PROCESSO DE ENGENHARIA E GERENCIAMENTO DE REQUISITOS PARA FASE CONCEITUAL DE SISTEMAS COMPLEXOS

(ARNAUT; FERRARI; SOUZA, 2016) propõem um processo para engenharia e gerenciamento de requisitos na fase conceitual de sistemas complexos. O processo integra sistematicamente um conjunto de métodos e técnicas da engenharia de requisitos com gerenciamento de projetos. Uma abordagem de processo de negócio foi usada para modelagem do processo, e sua eficácia foi avaliada em relação a um conjunto de recomendações para lidar com requisitos instáveis, pouco claros e incompletos.

Os resultados da aplicação do processo sugerem que a proposta permite que organizações de ER criem meios eficazes de planejamento, coordenem e engajem as partes interessadas, produzam requisitos que atendam as necessidades do cliente e definam o sistema correto.

### 6.5 ANÁLISE DOS TRABALHOS RELACIONADOS

Entende-se que o RMP-SoS difere das demais abordagens, citadas acima, pois este aborda questões como o comportamento emergente e desenvolvimento evolutivo e respeita a independência operacional e gerencial dos sistemas constituintes, combinando e considerando artefatos de outros elementos centrais da ESoS que não estão diretamente relacionados à definição de requisitos.

Por exemplo, no trabalho de (LEWIS et al., 2009), os autores declaram que sua proposta deve ser complementada com técnicas que funcionem em um contexto mais descentralizado. Além disso, nenhuma das técnicas considera os efeitos do comportamento emergente do SoS na definição dos requisitos. Já o estudo desenvolvido por (WALKER et al., 2014), é um método

	<b>Independência Operacional</b>	<b>Independência Gerencial</b>	<b>Distribuição Geográfica</b>	<b>Desenvolvimento Evolutivo</b>	<b>Comportamento Emergente</b>
Este trabalho	Sim	Sim	Sim	Sim	Sim
LEWIS, 2009	Sim	Sim	Não	Sim	Sim
WALKER et al., 2014	Sim	Sim	Sim	Não	Sim
LANE, 2014	Sim	Não	Sim	Sim	Não
ARNAUT et al., 2016	Não	Não	Sim	Sim	Não

Tabela 6.1 – Comparação entre os Trabalhos Relacionados

que não fornece um conjunto de requisitos pronto para incorporação na *baseline* técnica, ou seja, apenas fornece um conjunto de requisitos menos abstrato do que os objetivos de capacidade do SoS. E também, os autores não consideram o desenvolvimento evolutivo do SoS, uma vez que o método só funciona a partir do momento em que todas as capacidades já são conhecidas.

No estudo de (LANE, 2014), a integração das técnicas também não suporta o comportamento emergente de um SoS, e não considera as interações entre os sistemas constituintes. E o estudo de (ARNAUT; FERRARI; SOUZA, 2016), apesar de propor um processo de gerenciamento de requisitos capaz de acompanhar as mudanças e a interação dos *stakeholders*, não é um processo específico para o domínio de um SoS, por isso não considera suas características intrínsecas, como independência gerencial e operacional dos sistemas constituintes e comportamento emergente.

Um quadro comparativo em respeito às características intrínsecas de um SoS, atendidas ou não pelos trabalhos relacionados é apresentado na Tabela 6.1. A forma como o RMP-SoS atende a cada característica pode ser visualizada na tabela apresentada no Apêndice A.

## 7 CONCLUSÃO

Um *System of System* compreende múltiplos sistemas constituintes que são operados e gerenciados por diferentes *stakeholders*, estes sistemas têm seus próprios requisitos e seus próprios processos de gerenciamento que podem ou não alinhar-se aos objetivos e requisitos do SoS ao qual eles fazem parte.

Nesse cenário, diversos autores apontam a necessidade de clarificar as relações de gerenciamento entre SoS e SC, assim como a necessidade de mecanismos que possam gerenciar requisitos de SoS e adaptar as diversas mudanças que podem surgir durante seu desenvolvimento, por meio de um método sistemático para o gerenciamento de requisitos e que facilite o trabalho de colaboração entre os diversos *stakeholders*. Este problema de pesquisa pode ser identificado após intenso levantamento bibliográfico e com o desenvolvimento de uma revisão sistemática de literatura que contribuiu para elencar desafios de pesquisa e ideias de soluções.

A fim de propor uma solução para estas deficiências no campo da Engenharia de Requisitos para SoS, o desenvolvimento deste trabalho possibilitou avaliar e validar a proposta de uma abordagem para gerenciamento de requisitos de *System of Systems* por meio de um processo nomeado de RMP-SoS, que propôs 4 subprocessos organizados em diferentes tarefas, com o objetivo de organizar o trabalho e as responsabilidades dos envolvidos, elicitar objetivos de capacidade, conceito de operações, espaço de requisitos e requisitos de interoperabilidade, além de gerenciar as mudanças, recomendar e priorizar requisitos para um SoS a fim de permitir que o desenvolvimento evolutivo de um SoS seja mais previsível e adaptável.

O RMP-SoS é disponibilizado por meio de um *website* que apresenta detalhadamente todos seus subprocessos, tarefas com instruções, artefatos e seus templates disponíveis para *download*, além de diagramas de fluxo de trabalho que facilitam a sua execução na prática. Este processo foi criado com base em uma cuidadosa análise de literatura científica relacionada à Engenharia de SoS e seus atributos, tarefas e artefatos foram desenvolvidos para que atendessem as principais propriedades intrínsecas de um SoS.

Na primeira etapa de validação desta proposta, uma avaliação de conformidade foi desenvolvida com o objetivo de identificar e explicar como o RMP-SoS atende as cláusulas especificadas nos padrões internacionais aceitos no âmbito da Engenharia de Sistemas e da Engenharia de Requisitos, considerando que não há uma padronização internacionalmente aceita especificamente para Engenharia de SoS. Os resultados desta avaliação mostram que o processo proposto atende a maioria das recomendações sugeridas pelas normas ISO/IEC/IEEE 15288 (anexo G) e ISO/IEC/IEEE 29148. Entretanto, algumas cláusulas apresentadas pelas normas não são aplicáveis ao escopo do RMP-SoS, e alguns termos precisaram ser adaptados ao contexto de SoS para que pudessem ser avaliados.

A segunda etapa de validação consistiu em um estudo de caso combinado com uma pesquisa-ação, que teve como objetivo avaliar a autenticidade de um problema identificado, e

avaliar se a proposta de solução era válida, por meio da aplicação do processo RMP-SoS no contexto do Projeto SIS-ASTROS. Este estudo foi desenvolvido em seis etapas distintas que envolveram a observação e análise de documentos e da prática atual desempenhada no projeto, a identificação do problema e verificação da realidade do mesmo, a aplicação do RMP-SoS no projeto SIS-ASTROS por meio da produção dos documentos sugeridos pelos templates, uma reunião de apresentação e análise junto da equipe do projeto e a coleta de dados por meio de entrevista e discussão que tinham o objetivo de avaliar a corretude, usabilidade e efetividade das ações práticas, ou seja, a aplicação do processo.

Os resultados obtidos foram satisfatórios nas três dimensões avaliadas, algumas limitações do processo foram levantadas e discutidas com a equipe do projeto, assim como importantes sugestões de melhoria foram coletadas. Ao fim desta validação, pode-se afirmar que o processo para gerenciamento de requisitos de SoS proposto, o RMP-SoS, atende a maioria das recomendações internacionais no âmbito da Engenharia de Sistemas e da Engenharia de Requisitos. E que também cumpre com seu objetivo inicial de apresentar um método sistematizado para gerenciamento de trabalho dos envolvidos com a ER de SoS e das tarefas de elicitação e gerenciamento de requisitos, permitindo a incorporação de mudanças e tornando o processo evolutivo do SoS mais previsível. Assim, contribuindo com o campo de pesquisa de Engenharia de SoS com uma nova e relevante proposta para resolução dos desafios já mencionados.

## 7.1 TRABALHOS FUTUROS

Os resultados da validação permitem observar que este trabalho apresenta algumas limitações que podem ser resolvidas a partir de trabalhos futuros. Em primeiro lugar, uma ferramenta *online* para apoiar a aplicação do processo na prática, na qual os responsáveis pelas tarefas pudessem consultar e gerenciar as versões dos documentos produzidos em tempo real, além de trocar informações e mensagens, facilitaria o trabalho e a comunicação entre os envolvidos e o gerenciamento dos produtos de trabalho.

Um outro ponto a melhorar, que foi sugestão dada pelos participantes da segunda etapa de avaliação, seria definir guias de adaptação do processo para contextos e realidades diferentes de projeto. Um trabalho futuro, poderia refinar os templates já propostos e incluir mecanismos de adaptação para que o desenvolvimento das tarefas possa ser alterado de acordo com cada contexto de projeto.

E por fim, uma próxima versão do RMP-SoS pode atender às atividades propostas pelo sexto Elemento Central da ESoS (*Addressing Requirements and Solution Options*), onde a partir do documento de requisitos gerado (*SoS Recommended Requirements*) seja possível incluir tarefas que permitam encontrar os sistemas constituintes capazes de atender aqueles requisitos e a definir a forma como estas soluções serão implementadas.

## 7.2 PUBLICAÇÕES

Até o presente momento, como resultado deste trabalho foram publicados os seguintes trabalhos:

Um artigo de proposta foi desenvolvido, submetido e aceito no X Workshop de Teses e Dissertações em Sistemas de Informação, evento satélite do XIII Simpósio Brasileiro de Sistemas de Informação e apresentado no dia 08 de Junho de 2017, na cidade de Lavras, MG e publicado nos anais do evento. E está disponível em: <<http://sbsi2017.dcc.ufla.br/download/wtdsi.pdf>>

A Revisão Sistemática de Literatura desenvolvida durante a etapa inicial deste trabalho foi submetida e aceita em um evento de nível internacional, qualificado como *Qualis B1*, a *29th International Conference on Software Engineering and Knowledge Engineering - (SEKE 2017)*. O artigo foi apresentado no dia 05 de Julho de 2017 na cidade de *Pittsburgh (PA), EUA* e publicado nos anais do evento, disponível no endereço: <<https://ksiresearchorg.ipage.com/seke/seke17.html>>

E um artigo sobre o desenvolvimento e descrição do processo RMP-SoS foi submetido e aceito também na *30th International Conference on Software Engineering and Knowledge Engineering (SEKE 2018)*, cujo título é "*An Approach for System of Systems Requirements Management*". O artigo será apresentado no dia 01 de Julho de 2018 na cidade de *San Francisco Bay (CA), EUA* e está publicado nos anais do evento, disponível no endereço: <<http://ksiresearchorg.ipage.com/seke/seke18.html>>

## REFERÊNCIAS BIBLIOGRÁFICAS

ABBOTT, R. Open at the top; open at the bottom; and continually (but slowly) evolving. In: IEEE. **System of Systems Engineering, 2006 IEEE/SMC International Conference on**. [S.l.], 2006. p. 6–pp.

ARNAUT, B. M.; FERRARI, D. B.; SOUZA, M. L. d. O. e. A requirements engineering and management process in concept phase of complex systems. In: IEEE. **Systems Engineering (ISSE), 2016 IEEE International Symposium on**. [S.l.], 2016. p. 1–6.

BOARDMAN, J.; SAUSER, B. System of systems-the meaning of of. In: IEEE. **System of Systems Engineering, 2006 IEEE/SMC International Conference on**. [S.l.], 2006. p. 6–pp.

COOK, J. E.; WOLF, A. L. Software process validation: Quantitatively measuring the correspondence of a process to a model. **ACM Trans. Softw. Eng. Methodol.**, ACM, New York, NY, USA, v. 8, n. 2, p. 147–176, abr. 1999. ISSN 1049-331X. Disponível em: <<http://doi.acm.org/10.1145/304399.304401>>.

DAHMAN, J. et al. System engineering artifacts for sos. **IEEE Aerospace and Electronic Systems Magazine**, IEEE, v. 26, n. 1, p. 22–28, 2011.

DAHMAN, J.; ROEDLER, G. Moving towards standardization for system of systems engineering. In: IEEE. **System of Systems Engineering Conference (SoSE), 2016 11th**. [S.l.], 2016. p. 1–6.

DAHMAN, J. S.; BALDWIN, K. J. Understanding the current state of us defense systems of systems and the implications for systems engineering. In: IEEE. **Systems Conference, 2008 2nd Annual IEEE**. [S.l.], 2008. p. 1–7.

DOD-USA. Systems engineering guide for systems of systems. In: \_\_\_\_\_. [S.l.]: Washington, DC, 2008.

EASTERBROOK, S. et al. Selecting empirical methods for software engineering research. **Guide to advanced empirical software engineering**, Springer, p. 285–311, 2008.

GIL, A. **Como elaborar projetos de pesquisa**. [S.l.]: Atlas, 2010. ISBN 9788522458233.

HOLT, J. et al. Model-based requirements engineering for system of systems. In: IEEE. **System of Systems Engineering (SoSE), 2012 7th International Conference on**. [S.l.], 2012. p. 561–566.

IEEE. Systems and software engineering – life cycle processes –requirements engineering. **ISO/IEC/IEEE 29148:2011(E)**, 2011.

KEATING, C. et al. System of systems engineering. **Engineering Management Journal**, Taylor & Francis, v. 15, n. 3, p. 36–45, 2003.

KHAN, S.; TZORTZOPOULOS, P. A framework for evaluating an action research study on lean design management. In: IGLC. **Proc. 24th Ann. Conf. of the Int'l. Group for Lean Construction. Boston, USA, July 20-22**. [S.l.], 2016. p. 113–122.

LANE, J. A. System of systems capability to requirements engineering. In: IEEE. **System of Systems Engineering (SOSE), 2014 9th International Conference on**. [S.l.], 2014. p. 91–96.

LEWIS, G. A. et al. Requirements engineering for systems of systems. In: IEEE. **Systems Conference, 2009 3rd Annual IEEE**. [S.l.], 2009. p. 247–252.

LIMA, R. M.; VARGAS, D.; FONTOURA, L. System of systems requirements: A systematic literature review using snowballing. In: SEKE. **Software Engineering and Knowledge Engineering, 2017 SEKE / International Conference on**. [S.l.], 2017. p. 10–pp.

LIMA, R. M. de; FONTOURA, L. M. Um processo para gerenciamento de requisitos de sistema de sistemas. In: XIII SBSI. **X Workshop de Teses e Dissertações em Sistemas de Informação**. [S.l.], 2017. p. 85–87.

MAIER, M. W. Architecting principles for systems-of-systems. In: WILEY ONLINE LIBRARY. **INCOSE International Symposium**. [S.l.], 1996. v. 6, n. 1, p. 565–573.

NIELSEN, C. B. et al. Systems of systems engineering: Basic concepts, model-based techniques, and research directions. **ACM Computing Surveys (CSUR)**, ACM, v. 48, n. 2, p. 18, 2015.

PMI. **A Guide to the Project Management Body of Knowledge (PMBOK Guide), 4th Edition**. [S.l.]: Project Management Institute, 2008.

PRESSMAN, R. **Engenharia de Software**. AMGH, 2011. ISBN 9788563308337. Disponível em: <<https://books.google.com.br/books?id=eRIOuQAACAAJ>>.

PRINCE2. **Roles and responsibilities**. PRINCE2 Wiki, 2017. Disponível em: <[http://prince2.wiki/Roles\\_and\\_responsibilities](http://prince2.wiki/Roles_and_responsibilities)>.

SAYÃO, M.; LEITE, J. C. S. do P. Rastreabilidade de requisitos. **RITA**, v. 13, n. 1, p. 57–86, 2006.

SEBOK. System engineering body of knowledg. In: \_\_\_\_\_. [S.l.]: System Engineering, 2016.

SOMMERVILLE, I. **Engenharia de software**. Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>.

SOMMERVILLE, I. et al. **Engenharia de software**. [S.l.]: Addison Wesley São Paulo, 2003. v. 6.

TECNODEFESA. **O Exército Brasileiro e a Simulação de Combate @ONLINE**. 2016. Disponível em: <<http://tecnodefesa.com.br/o-exercito-brasileiro-e-a-simulacao-de-combate/>>.

UFMS. **Projeto de Simulação ASTROS 2020**. [S.l.], 2014.

\_\_\_\_\_. **SIMULADOR VIRTUAL TÁTICO - Documento Visão**. [S.l.], 2016.

\_\_\_\_\_. **SIMULADORES VIRTUAIS TÉCNICOS SVTEC - Documento Visão**. [S.l.], 2017.

WALDEN, D. D. et al. **Systems engineering handbook: A guide for system life cycle processes and activities**. [S.l.]: John Wiley & Sons, 2015.

WALKER, R. G. et al. A method to define sos requirements. **International Journal of System of Systems Engineering**, Inderscience Publishers, v. 5, n. 4, p. 289–323, 2014.

YIN, R. K. **Estudo de Caso-: Planejamento e Métodos**. [S.l.]: Bookman editora, 2010.

## APÊNDICE A – TABELA DE ATRIBUTOS x TAREFAS DO RMP-SoS

Característica do SoS	Atributo Requerido no Processo	Tarefas do RMP-SoS	Artefatos do RMP-SoS
<p><b>Independência Operacional</b> – Qualquer sistema que faz parte de um SoS é independente e capaz de operar de forma eficaz, mesmo que o SoS seja desmembrado (MAIER, 1996).</p> <p><b>Independência Gerencial</b> - Apesar de colaborar com os outros membros dos SoS, os sistemas constituintes são autônomos e gerenciados individualmente para que eles "não só possam operar de forma independente, mas que eles realmente operem de forma independente" (MAIER, 1996).</p>	<p>Quando uma nova capacidade ou requisição de mudança aparece ela deve ser analisada a qual nível pertence e direcionada ao responsável. (HOLT et al., 2012) (SOMMERVILLE, 2011)</p>	<p><i>Analyze Change Request</i></p>	<p><i>Change Request</i></p> <p><i>Change Record</i></p>
	<p>SC deve identificar quais mudanças em seus requisitos são capazes de impactar o SoS. (DOD, 2008)</p> <p>SC devem informar ao gerente do SoS sobre as mudanças que ocorrerem nos requisitos. (DOD, 2008)</p> <ul style="list-style-type: none"> <li>• O SoS precisa se conectar aos processos do sistema constituintes.</li> <li>• A equipe de engenharia do SoS precisa estar ciente dos requisitos dos sistemas, bem como dos planos de financiamento e mudanças agendadas, para que possam antecipar os impactos das mudanças dos sistemas no SoS.</li> </ul> <p>Em um SoS, pode ser difícil gerenciar recursos redundantes em sistemas individuais, caso os sistemas precisem da capacidade redundante para atender às suas próprias necessidades quando operam separadamente ou as necessidades de outros SoS em que participam (DOD, 2008).</p>	<p><i>Identify SC Requirements Changes</i></p>	<p><i>CS Info Which Impacts SoS Requirements</i></p>
	<p>O ambiente operacional do SoS visa atender a um conjunto de objetivos usando sistemas cujos objetivos podem ou não alinhar-se aos objetivos do SoS (DAHMANN; BALDWIN, 2008).</p>	<p><i>Identify SoS Capability Objectives</i></p> <p><i>Develop SoS Conops</i></p> <p><i>Define SoS Requirements Space</i></p>	<p><i>SoS Capability Objectives</i></p> <p><i>SoS CONOPS</i></p> <p><i>SoS Requirements Space</i></p>
		<p><i>Identify SC Requirements Changes</i></p>	<p><i>CS Info Which Impacts SoS Requirements</i></p>
	<p>O fato de que muitos sistemas constituintes desempenham um papel em múltiplos SoS significa que o clarificar as relações de gerenciamento não é uma tarefa</p>	<p><i>Establish Collaboration Agreements</i></p>	<p><i>Collaboration Agreements</i></p>



	<p>simples. Por isso acordos formais devem ser estabelecidos (DAHMANN; BALDWIN, 2008).</p> <p>Pode ser uma tarefa difícil estabelecer confiança entre os <i>stakeholders</i> que podem não querer compartilhar detalhes de seu sistema por razões de confidencialidade ou competitividade (NIELSEN et al., 2015).</p>		
	<p>Fatores de escalabilidade como múltiplas interações entre SC, grande número de usuários e <i>stakeholders</i> aumentam a flexibilidade requerida dos SC e também o número de restrições impostas nas soluções de engenharia (LEWIS et al., 2009)</p> <p>Identificar o tipo de SoS deve ser o primeiro passo da ER, pois isso é o que dita a forma que os requisitos tomarão tanto para os SC quanto para o SoS (LEWIS et al., 2009).</p>	<i>Prepare RE Work Plan</i>	<i>RE Work Plan</i>
<p><b>Distribuição Geográfica</b> – As partes que colaboram em um SoS são distribuídas em uma grande extensão geográfica. Mesmo que essa extensão seja vagamente definida, enfatiza-se que os sistemas constituintes só podem trocar informações e não quantidades consideráveis de energia ou massa (MAIER, 1996).</p>	<p>Identificar todos os <i>stakeholders</i> e seus papéis. (HOLT et al., 2012)</p>	<i>Identify Stakeholders</i>	<i>Stakeholders Register</i>
		<i>Assign Roles and Responsibilities</i>	<i>Stakeholders Roles and Responsibilities</i>
	<p>Estabelecer acordos formais e não formais entre SoS e SC. (DOD, 2008)</p>	<i>Establish Collaboration Agreements</i>	<i>Collaboration Agreements</i>
	<p>Engajamento dos engenheiros dos SC e do SoS para entender a natureza das mudanças e impactos no SoS (DOD, 2008)</p> <p>Tipicamente, existem mais de uma autoridade de tomada de decisão controlando vários aspectos do SoS. Por isso, a engenharia de requisitos precisa ser realizada através da colaboração entre os múltiplos <i>stakeholders</i> durante todo contínuo e evolutivo desenvolvimento do SoS. (LEWIS et al., 2009).</p>	<i>Analyze Change Request</i> (Change Control Board)	<i>Change Record</i>
	<p>Estabelecer um canal de comunicação entre engenheiros dos SC e do SoS. (DOD, 2008)</p> <p>Troca de dados de requisitos no contexto do SoS (LEWIS et al., 2009).</p> <p>Existem desafios relacionados a troca de conhecimento e informações entre os diversos <i>stakeholders</i> (NIELSEN et al., 2015).</p>	Meio de troca de mensagens - Ferramenta	Não atendido pela primeira versão do RMP-SoS.
<p><b>Comportamento Emergente</b> – Através da colaboração entre os SC de</p>	<p>Estabelecer uma relação entre os requisitos do SoS e dos SC (DOD, 2008).</p>	<i>Prepare RE Work Plan</i> (Traceability Policies)	<i>RE Work Plan</i>

<p>um SoS é alcançada uma sinergia em que o comportamento do sistema atende a um propósito que não pode ser alcançado ou atribuído a nenhum dos sistemas individuais (MAIER, 1996).</p>	<p>Controlar <i>top-level requirements changes</i> para manter estabilidade e coerência (DOD 2008).</p> <p>Identificar redundâncias nos requisitos entre os SC.</p>	<p><i>Review SoS CO, SoS Conops and SoS ReqSpace</i></p>	<p><i>Review Record or Change Request</i></p>
	<p>Um grande desafio para o designer de SoS é saber, ou aprender como, enquanto o SoS progride através de sua série de estados estáveis, criar um clima no qual a emergência possa florescer, e uma agilidade para detectar e destruir rapidamente comportamentos não intencionais (BOARDMAN; SAUSER, 2006)</p> <p>A evolução de qualquer parte do SoS pode levar a um comportamento emergente indesejável, que precisará ser identificado, analisado e tratado (LOCK, 2012).</p>	<p><i>Analyze Change Request</i></p> <p><i>Identify SC Requirements Change</i></p>	<p><i>Change Record</i></p> <p><i>CS Info which impacts SoS Requirements</i></p>
	<p>A interoperabilidade refere-se à capacidade dos SoS de incorporar uma série de sistemas constituintes heterogêneos. Isso envolve a integração e adaptação de interfaces, protocolos e padrões para permitir a ponte entre sistemas legados e novos (NIELSEN et al., 2015).</p> <p>Comportamento dinâmico, comportamento emergente e evolução podem requerer mudanças na arquitetura, pois essas alterações influenciam na segurança, dependabilidade dos sistemas e manipulação de tempo (CECCARELLI et al., 2015).</p> <p>Após a construção ou simulação de uma arquitetura, caso o designer tenha aplicado uma boa estratégia, é possível identificar e premeditar alguns comportamentos emergentes, que serão interessantes para definição de requisitos (CECCARELLI et al., 2015).</p>	<p><i>Get Initial Requirements from Architecture</i></p>	<p><i>SoS Initial Architecture Requirements</i></p>
<p><b>Desenvolvimento Evolucionário</b> – A existência e o desenvolvimento do SoS são evolutivos, no sentido de que os objetivos e as funcionalidades podem estar em constante mudança, pois podem ser modificados, adicionados, ou removidos. Assim, um SoS nunca parece estar completamente formado</p>	<p>Revisão contínua dos requisitos conforme o SoS evolui. (ISO 15288)</p>	<p><i>Review SoS CO, SoS Conops and SoS ReqSpace</i></p>	<p><i>Review Record or Change Request</i></p>
	<p>Identificar mudanças nos requisitos dos SC e impactos das mudanças no SoS (DOD, 2008).</p>	<p><i>Identify SC Requirements Change</i></p>	<p><i>CS Info which impacts SoS Requirements</i></p>
	<p>Os engenheiros dos SC trabalham em conjunto com o gerente e engenheiro de SoS para revisar, priorizar e recomendar quais requisitos devem ser implementados a cada iteração. Os requisitos selecionados para cada</p>	<p><i>Document Recommended Requirements for SoS</i></p>	<p><i>SoS Recommended Requirements</i></p>

(MAIER, 1996).	<p>iteração são movidos para a <i>baseline</i> funcional da iteração (DOD, 2008).</p>		
	<p>Obter os requisitos oriundos da arquitetura (DOD, 2008).</p> <p>A formação da arquitetura de um SoS pode ser considerada uma fonte de requisitos não funcionais, pois esta descreve “os SC que compõem um SoS e suas regras de interoperabilidade, dependências e interações” (CECCARELLI et al., 2015).</p>	<p><i>Get Initial Requirements from Architecture</i></p>	<p><i>SoS Initial Architecture Requirements</i></p>
	<p>Incluir novos requisitos com o menor impacto ao SoS (DOD, 2008).</p>	<p><i>Analyze Change Request</i></p>	<p><i>Change Record</i></p>
		<p><i>Update SoS CO, SoS Conops and SoS ReqSpace</i></p>	<p><i>SoS Capability Objectives, SoS Conops and SoS Requirements Space updated</i></p>
	<p>Documentar a origem de cada requisito (por um conjunto de atributos e um contexto) (HOLT et al., 2012).</p> <p>Garantir a rastreabilidade dos requisitos (HOLT et al., 2012)].</p> <p>Compreender o ambiente ajuda os sistemas individuais e os SoS a antecipar as necessidades futuras e a engenharia de mecanismos apropriados (por exemplo, da arquitetura) para se adaptar a essas mudanças no ambiente (LEWIS et al., 2009).</p>	<p><i>Prepare RE Work Plan (Traceability Policies)</i></p>	<p><i>RE Work Plan</i></p>
	<p>O Engenheiro de SoS determina as opções para mudanças ou novos requisitos para atender aos objetivos do SoS trabalhando em conjunto com os engenheiros e gerentes dos SC . Essas decisões são tomadas em conjunto pois esses <i>stakeholders</i> são tomadores de decisão chave no processo. (DAHMAN; BALDWIN, 2008).</p>	<p><i>Analyze Change Request</i></p>	<p><i>Change Record</i></p>
	<p>Objetivos de capacidades devem ser ajustados de forma realista a fim de refletir as opções de implementação viáveis. (DOD, 2008)</p> <p>Os objetivos de um SoS geralmente são definidos em termos de capacidades necessários e o engenheiro de sistemas é responsável por trabalhar com o gerente de SoS e os usuários para traduzi-los em requisitos de alto nível que podem fornecer a base para o planejamento técnico e evoluir as capacidades ao longo do tempo (DOD, 2008)</p>	<p><i>Identify SoS Capability Objectives</i></p> <p><i>Develop SoS CONOPS</i></p> <p><i>Define SoS Requirements Space</i></p>	<p><i>SoS Capability Objectives</i></p> <p><i>SoS CONOPS</i></p> <p><i>SoS Requirements Space</i></p>

	(LANE, 2014).		
	<p>SoS provavelmente evoluirá baseado nas mudanças incrementais dos sistemas individuais. (DOD, 2008).</p> <p>A evolução pode se manifestar como atualizações nos sistemas constituintes, o que requer constante verificações de conformidade (NIELSEN et al., 2015).</p>	<p><i>Identify CS Requirements Change</i></p>	<p><i>CS Info Which impacts SoS Requirements</i></p>
	<p>A evolução dos requisitos é conduzida por uma variedade de fontes: mudanças no ambiente do SoS; comportamento emergente; mudanças nos sistemas constituintes; problemas de atualização no SoS; perspectivas e necessidades do usuário; oportunidades tecnológicas (NIELSEN et al., 2015).</p>	<p><i>Review SoS CO, SoS Conops and SoS ReqSpace</i></p> <p><i>Update SoS CO, SoS Conops and SoS ReqSpace</i></p>	<p><i>Review Record or Change Request</i></p> <p><i>SoS Capability Objectives, SoS Conops and SoS Requirements Space updated</i></p>

## APÊNDICE B – TEMPLATES DO RMP-SoS

<b>RMP-SoS - v.1/2018</b>
<b>Subprocess 1:</b> Manage Stakeholders <b>Task:</b> Identify Stakeholders <b>Template:</b> Stakeholders Register

### Version History

Data	Version - Description	Responsible	Reviewer

### References

<Lista de documentos usados como referência nesta tarefa.>

Stakeholder Identification	
<b>Name:</b> <nome do stakeholder>	<b>ID:</b> <identificação única do stakeholder>
<b>Project Position:</b> <sua posição dentro da organização ou projeto responsável pelo SoS ou pelo SC>	
<b>Phone:</b> <telefone principal>	<b>e-mail:</b> <e-mail principal>
Abilities and Knowledge	
<b>Graduation:</b> <formação principal e área de conhecimento>	
<b>Abilities:</b> <habilidades relacionadas que serão úteis ao processo>	
<b>Knowledge:</b> <conhecimentos que serão úteis ao processo>	
Classification	
<b>Influence level:</b> <capacidade de alavancar ou paralisar o projeto, de 0 a 5>	
<b>Importance level:</b> <necessidade de seu conhecimento para o projeto, de 0 a 5>	

**RMP-SoS - v.1/2018****Subprocess 1:** Manage Stakeholders**Task:** Assign Roles and Responsibilities**Template:** Stakeholders Roles and Responsibilities**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa.>

<b>Stakeholder ID:</b> <ID única do representante>	<b>RMP-SoS Role:</b> <papel desempenhado pelo stakeholder no processo>
<b>Stakeholder Name:</b> <nome do representante>	
<b>Responsibilities:</b> <responsabilidades no projeto e tarefas que poderão ser desempenhadas pelo stakeholder no processo>	
<b>Work Products:</b> <listagem dos produtos de trabalho (de entrada e saída) de responsabilidade do stakeholder ou relacionados às suas responsabilidades>	

## RMP-SoS - v.1/2018

**Subprocess 1:** Manage Stakeholders

**Task:** Prepare RE Work Plan

**Template:** Requirements Engineering Work Plan

### Version History

Data	Version - Description	Responsible	Reviewer

### References

<Lista de documentos usados como referência nesta tarefa.>

#### 1. Introduction

<Identificar o tipo de SoS e descrever o ambiente operacional>

#### 2. Project Scope Summary

<Resumir as missões e objetivos principais do projeto do SoS>

#### 3. RE Work Propose

- a. Atividades e Tarefas
- b. Ordem de Execução (workflow)
- c. Planejamento (tempo e custo)

#### 4. Constraints

<Identificar e descrever quaisquer restrições que condicionem a execução das atividades da ER do SoS>

#### 5. Traceability Policies

<Definir políticas que estabelecem os relacionamentos entre os objetivos de capacidades desejados e os stakeholders, bem como entre os objetivos de capacidades e os requisitos. Bem como, definir como esses registros serão mantidos>

- a. Objetivos de capacidade desejados x Stakeholders de origem
- b. Matriz de Rastreabilidade 1: Goals x SoS Capability Objectives
- c. Matriz de Rastreabilidade 2: SoS Capability Objectives x SoS Requirements Space
- d. Matriz de Rastreabilidade 3: SoS Requirements x SoS Requirements Space
- e. Matriz de Rastreabilidade 4: SoS Requirements x SoS Performance Objectives
- f. Matriz de Rastreabilidade 5: SoS Requirments x Interoperability Requirements

#### 6. Authorization for the begining of the RE activities

<Instruções para o início das tarefas>

#### 7. Glossary

<Listar termos e abreviaturas, usadas nos documentos produzidos nesse processo e seus respectivos significados>

## RMP-SoS - v.1/2018

**Subprocess 2:** Elicit SoS Requirements  
**Task:** Identify SoS Capability Objectives  
**Template:** SoS Capability Objectives

### Version History

Data	Version - Description	Responsible	Reviewer

### References

<Lista de documentos usados como referência nesta tarefa.>

#### 1. Introduction

<Identificar a proposta do documento>

#### 2. SoS Project General Missions

<Descrever o objetivo geral do SoS, identificar os objetivos específicos e as necessidades que motivam o desenvolvimento e a execução do SoS>

#### 3. Constraints

<Identificar as restrições de desenvolvimento disponíveis e conhecidas. Ex: políticas de negócio, regras, recomendações, leis ou normas>

#### 4. Stakeholders Goals

<Definir e descrever um “resultado desejado” que o esforço do SoS será direcionado para atender>

<b>Goal 01:</b> <descrição do objetivo em alto nível>	<b>Origin:</b> <Stakeholder ID>
---	---------------------------------

#### 5. Capability Objectives

<Descrever as habilidades necessárias do sistema para fazer algo a fim de entregar os objetivos citados>

<b>SoS CO 01:</b> <descrição do objetivo em alto nível>	<b>Origin:</b> <Stakeholder ID> <Goal ID>
<b>Constraint:</b> <possível restrição de desenvolvimento>	

#### 6. Traceability Matriz

<Matriz de rastreabilidade entre Goals X SoS Capability Objectives >

	SoS CO 01	SoS CO 02	SoS CO 03	SoS CO 04	SoS CO 05	SoS CO 06
Goal 01						
Goal 02						



**RMP-SoS - v.1/2018****Subprocess 2:** Elicit SoS Requirements**Task:** Develop SoS Conops**Template:** SoS Conops**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa.>

**1. Introduction:**

<Descrever brevemente a visão geral do SoS, a proposta para implementação do SoS, os recursos necessários e os stakeholders envolvidos>

**2. SoS Overview**

<Descrever em alto nível os requisitos de missão, as inter-relações dos sistemas constituintes-chave, e a interdependência entre sistemas constituintes>

**3. Operation Environment by the User's Point of View**

<Resumir o papel de cada usuário no ambiente operacional do SoS, esclarecer a ordem das operações, resumir procedimentos, e descrever diagramas de fluxo associados a tomada de decisão>

**4. Operational and Support Environments**

<Descrever a infraestrutura associada a cada sistema constituinte em termos de: instalações, equipamentos, hardware, software, pessoal, procedimentos operacionais, requisitos de manutenção, treinamento e suporte>

**5. Operational Scenarios**

<Descrever o sistema "em ação", usando um ou mais cenários de representação para refletir: diversas perspectivas dos stakeholders; uma serie de cenários de falha; circunstancias típicas e extremas>

**RMP-SoS - v.1/2018****Subprocess 2:** Elicit SoS Requirements**Task:** Define SoS Requirements Space**Template:** SoS Requirements Space**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa>

**1. Introduction**

<Fornecer uma visão geral ou um sumário do documento>

**2. Functions Required to provide Capabilities**

<Listar as necessidades de primeira ordem dos usuários e definir as funções necessárias aos sistemas constituintes para fornecer as capacidades desejadas>

**SoS CO 01:** <descrever o objetivo de capacidade relacionado>

**Origin:** <STK 00 – Goal 00>

**ReqSpace 01:** <descrever o espaço de requisito em termos de funcionalidades necessárias para atender a capacidade desejada >

**3. Variability in the User Environment**

<Considerar as variabilidades no ambiente do usuário que impactam a forma como essas funções serão executadas e descrevê-las>

**4. Performance Objectives**

<Descrever os objetivos de performance>

**Perf Obj 01:** <descrever sumariamente o objetivo de performance>

**5. Capability Backlog**

<Listar as capacidades a serem atendidas e suas respectivas funções>

Matriz de Rastreabilidade 2: SoS Capability Objectives x Requirements Space

**6. Problem Reports**

<Identificar problemas ou dificuldades para atender as capacidades>

**RMP-SoS - v.1/2018****Subprocess 2:** Elicit SoS Requirements**Task:** Get Initial Requirements from Architecture**Template:** SoS Initial Architecture Requirements**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa.>

**1. Introduction**

<Fornecer uma visão geral da arquitetura do SoS e um sumário do documento>

**2. Constituent Systems Information**

<Descrever informações dos SC relevantes para a arquitetura>

**3. Constraints**

<Descrever as restrições, regras ou políticas de interoperabilidade entre os sistemas constituintes>

**4. Interoperability Requirements**

<Listar os requisitos de interoperabilidade necessários, ou seja, os atributos e funcionalidades que facilitem o relacionamento entre os SC>

**Int Req 01:****Int Req 02:****Int Req 03:**

## RMP-SoS - v.1/2018

**Subprocess 2:** Elicit SoS Requirements

**Task:** Review SoS CO, SoS Conops and SoS ReqSpace

**Template:** Review Record

### Version History

Data	Version - Description	Responsible	Reviewer

#### 1. Introduction

<Resumir o documento e identificar o(s) tipo(s) de verificação(ões) que foi(ram) executada(s) (de validade, de consistência, de completude ou de realismo) ou descrever a revisão (de interoperabilidade, ou mudanças nos requisitos dos SC)>

#### 2. Work Products Reviewed and Objectives

<Listar o documento que foi revisado e descrever os objetivos da revisão>

#### 3. Review Participantes

<Listar todos os stakeholders participantes da revisão e seus papéis durante a tarefa>

#### 5. Problems Identified and Recommendations for Resolution

<Listar quaisquer problemas ou inconsistências encontradas nos documentos revisados, os revisores devem identificar: problemas que requerem correção (defeitos), problemas de projeto, ou problemas de produtos de trabalho>

#### 6. Action Item Status

<Listar todas as ações destinados a corrigir os problemas identificados e podem incluir:

“Continue work” (produto de trabalho não é considerado completo e seu desenvolvimento deve continuar);

“Indicate work to do” (se o problema requer que uma tarefa seja planejada novamente, mas isso não muda o produto de trabalho);

“Raise Change Request” (se o problema requer que seja feita uma mudança, preencha o documento “Change Request”);

\* Action Items de revisões anteriores devem ser listadas, bem como seu status (open/closed)>

#### 7. Follow-up Review

<Descrever recomendações para revisões futuras e quando esta será necessária, e informações adicionais necessárias>

## RMP-SoS - v.1/2018

**Subprocess 2:** Elicit SoS Requirements

**Subprocess 3:** Manage Changes

**Task:** Review SoS CO, SoS Conops and SoS ReqSpace

**Task:** Analyze Change Request

**Task:** Identify CS Requirements Change

**Template:** Change Request

### Version History

Data	Version - Description	Responsible	Reviewer

### References

<Lista de documentos usados como referência nesta tarefa.>

#### 1. Introduction

<Resumir o documento e identificar o tipo de verificação que foi executada (de validade, de consistência, de completude ou de realismo) ou descrever a revisão (de interoperabilidade, ou mudanças nos requisitos dos SC)

Definir:

**Change Request Number:**

**Change Request Type (problem or enhancement):**

**Date:**

**Originator (for traceability):**

**Change Request Priority:**

>

#### 2. Current Problem

<Descrição do problema atual; Falha Crítica; Aprimoramento; Novo Requisito; Condições sobre as quais o problema foi observado; Ambiente Operacional Atual; Fonte do Problema Atual; Custo e Impactos do problema atual >

#### 3. Proposed Change (from originator)

<Descrição da mudança proposta; Requisito ou objetivo que a originou; custo estimado para implementação da mudança proposta >

#### 4. Proposed Change (from Review Team)

<Ação; descrição da mudança proposta; impactos da mudança na configuração atual do SoS; categoria; correção de erros; melhoria; nova característica; outro>

#### 5. Change Proposed

<Descrever as mudanças propostas aprovadas e aceitas>

**RMP-SoS - v.1/2018****Subprocess 3:** Manage Changes**Task:** Identify CS Requirements Changes**Template:** CS Info which Impacts SoS Requirements**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa.>

**1. Introduction**

<Descrever uma visão geral do documento>

**2. Constituent System Info**

<Identificar e obter informações relevantes do SC que apresentou mudanças nos seus requisitos>

**3. Changes Assessment**

<Avaliar e descrever as mudanças nos requisitos bem como os impactos causados pelas mudanças identificadas>

**4. Possible Solution**

<Com auxílio dos engenheiros do SC, listar possíveis ações para amenizar o impacto e incorporar as mudanças necessárias>

**RMP-SoS - v.1/2018****Subprocess 3:** Manage Changes**Task:** Analyze Change Request**Template:** Change Record**Version History**

<b>Data</b>	<b>Version - Description</b>	<b>Responsible</b>	<b>Reviewer</b>

**References**

<Lista de documentos usados como referência nesta tarefa.>

**1. Introduction**

<Resumir o documento e identificar o tipo de mudança que será implementada

Definir:

**Change Request Number:**

**Change Record Number:**

**Change Record Type (problem or enhancement):**

**Date:**

**Originator (for traceability):**

**Change Implementation Priority:**

>

**2. Problem Resolution**

<Descrever o custo estimado para implementar a mudança proposta; responsável pela implementação; análise; requisito ou objetivo que originou a mudança; implementação>

**3. Instructions**

<Instruções para implementar a mudança em determinado documento seguindo seu template original>

**RMP-SoS - v.1/2018****Subprocess 4:** Manage SoS Requirements**Task:** Document Recommended SoS Requirements**Template:** SoS Recommended Requirements**Version History**

Data	Version - Description	Responsible	Reviewer

**References**

&lt;Lista de documentos usados como referência nesta tarefa.&gt;

**1. Introduction**

&lt;Definir a proposta do SoS e incluir uma visão geral; identificar a atual iteração do ciclo de desenvolvimento do SoS&gt;

**2. SoS Requirements**

&lt;Functional Requirements, Non-functional Requirements, Interoperability Requirements&gt;

<b>SoS Requirement 01:</b> <descrição do requisitos em alto nível>	<b>Origin:</b> <capability objective / requirement space>
<b>Type:</b> <functional, non-functional or interoperability requirement>	

**3. Traceability Matrix**

## SoS Requirements x SoS Requirements Space

	SoS Req	SoS Req	SoS Req	SoS Req	SoS Req
Req Space					
Req Space					
Req Space					
Req Space					
Req Space					
Req Space					
Req Space					

## SoS Requirements x Performance Objectives

	Perf Obj	Perf Obj	Perf Obj	Perf Obj	Perf Obj	Perf Obj	Perf Obj
SoS Req							
SoS Req							
SoS Req							
SoS Req							
SoS Req							



## SoS Requirements x Interoperability Requirements

	IntReq	IntReq	IntReq	IntReq
SoS Req				
SoS Req				
SoS Req				

### 4. Allocation Suggestion

<Listar possíveis SC que possam cumprir as funcionalidades desejadas>

## APÊNDICE C – ROTEIRO DE ENTREVISTA E DISCUSSÃO

Adaptado de: (KHAN; TZORTZOPOULOS, 2016)

1. O diagnóstico dos pesquisadores sobre as limitações associadas a ER tradicional desenvolvida no Projeto SIS-ASTROS foi preciso?
2. A Engenharia de Requisitos do SIS-ASTROS seria complementada de forma eficiente com o uso do RMP-SoS?
3. Em termos de simplicidade e clareza, o RMP-SoS seria fácil de ser implementado no Projeto SIS-ASTROS?
4. O RMP-SoS seria facilmente adaptável ao contexto de desenvolvimento do Projeto SIS-ASTROS?
5. As tarefas primárias (*Identify SoS CO, Develop SoS Conops, Define SoS ReqSpace, Get Initial SoS Architecture Requirements e Document Recommended Requirements for SoS*) cumprem o objetivo de especificar os requisitos do SIS-ASTROS de forma coerente e eficaz?
6. A distribuição das responsabilidades sugeridas pelo RMP-SoS cumpre o objetivo de organizar o trabalho e a interação entre os stakeholders de forma eficiente?
7. As tarefas de articulação (*Identify CS Requirements Change, Review SoS CO, SoS Conops and SoS ReqSpace; Update SoS CO, SoS Conops and SoS ReqSpace; Identify CS Requirements Changes e Analyze Change Request*) cumprem o objetivo de manter a consistência dos requisitos do SIS-ASTROS, bem como de incluir e adaptar mudanças conforme seu desenvolvimento evolui?
8. A comunicação e sincronização entre as tarefas do RMP-SoS por meio dos artefatos (documentos) gerados é eficaz e de fácil entendimento?
9. O RMP-SoS fornece um controle no fluxo de trabalho da Engenharia de Requisitos do SIS-ASTROS que a Engenharia tradicional não seria capaz de fornecer?
10. A aplicação do RMP-SoS pode encorajar ações ou mudanças a serem tomadas no contexto da ER do SIS-ASTROS?
11. A aplicação do RMP-SoS produzirá novos conhecimentos para a equipe de ER do SIS-ASTROS que podem vir a influenciar na mudança da prática?
12. Quais os pontos fortes e fracos do RMP-SoS, no seu ponto de vista?
13. Quais seriam as barreiras para a adoção do RMP-SoS de forma efetiva no gerenciamento de requisitos do SIS-ASTROS? E quais aspectos do RMP-SoS tornariam sua aplicação fácil?
14. Quais são suas sugestões para melhoria do RMP-SoS?