

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL FACULDADE DE
INFORMÁTICA PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MATEUS PRESOTTO BALEN

**PROCESSO E MODELO PARA IDENTIFICAÇÃO DOS PRINCIPAIS STAKEHOLDERS DE
SISTEMA-DE-SISTEMAS ACKNOWLEDGED**

Porto Alegre
2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

Ficha Catalográfica

B183p Balen, Mateus Presotto

Processo e Modelo para Identificação dos Principais Stakeholders de Sistema-de-Sistemas Acknowledged / Mateus Presotto Balen . – 2018.
118p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina dos Santos Marczak.

1. Stakeholder. 2. Identificação de Stakeholder. 3. Sistema-de-Sistemas. 4. Engenharia de Requisitos. 5. Arquitetura de Software. I. Marczak, Sabrina dos Santos. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Mateus Presotto Balen

Processo e Modelo para Identificação dos Principais Stakeholders de Sistema-de-Sistemas Acknowledged

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 12 de março de 2018.

BANCA EXAMINADORA:

Profa. Dra. Elisa Yumi Nakagawa (USP)

Prof. Dr. Tiago Coelho Ferreto (PPGCC/PUCRS)

Profa. Dra. Sabrina dos Santos Marczak (PPGCC/PUCRS - Orientador)

AGRADECIMENTOS

Primeiramente, eu gostaria de agradecer à Deus, por ter me dado saúde, foco e forças para seguir meu objetivo previamente traçado. Agradecer à minha família, Élide, Irineu, Manuela e Marcelo, que me apoiaram e acreditaram que eu era capaz de alcançar esta conquista. Tenho conhecimento dos sacrifícios que todos fizeram para que eu alcançasse meu objetivo.

Gostaria de agradecer aos meus amigos, pela compreensão com meus compromissos no mestrado, aos colegas do grupo Munddos, em especial à Carolina Toscani, Júlia Couto, Luis Vaz e Olimar Borges pelos conselhos, companheirismos e ajudas que foram extremamente importantes na caminhada, a colega da USP Cristiane Aparecida Lana pelo tempo disponibilizado e debates que me ajudaram a refletir o futuro do trabalho e a uma pessoa muito especial que em pouco tempo ao meu lado me ajudou e apoiou muito. Gostaria de expressar o meu apreço especial à minha orientadora, Prof. Dra. Sabrina Marczak, por me permitir alcançar o sonho do mestrado e por ter me guiado no caminho correto.

PROCESSO E MODELO PARA IDENTIFICAÇÃO DOS PRINCIPAIS STAKEHOLDERS DE SISTEMA-DE-SISTEMAS ACKNOWLEDGED

RESUMO

Sistemas de software têm se tornado maiores e mais complexos com considerável disseminação em muitas áreas. Devido à necessidade de se tornarem mais abrangentes, eventualmente, os sistemas operacionalmente e gerencialmente independentes tendem a interoperar entre eles. Esta integração entre sistemas independentes com natureza evolutiva e geograficamente distribuído resulta uma classe de sistemas chamada Sistema-de-Sistemas. Uma vez que a Arquitetura de Software define a estrutura base de um sistema e a Engenharia de Requisitos identifica as funcionalidades necessárias para um sistema, entende-se que essas atividades são essenciais para qualquer sistema, assim tornando a identificação dos *stakeholders* de um sistema uma atividade primordial para elicitar os requisitos necessários e, conseqüentemente, construir o sistema com as funcionalidades esperadas. Esta dissertação apresenta um processo e um modelo para a identificação dos principais *stakeholders* de Sistema-de-Sistemas *Acknowledged* com o intuito de minimizar os problemas causados nas definições dos requisitos do sistema e da arquitetura de software por desconhecer os principais *stakeholders*.

Palavras-Chave: *Stakeholder*, Identificação de *Stakeholder*, Sistema-de-Sistemas, Engenharia de Requisitos, Arquitetura de Software, Processo, Modelo.

PROCESS AND MODEL TO IDENTIFY OF KEY STAKEHOLDERS OF ACKNOWLEDGED SYSTEM-OF-SYSTEMS

ABSTRACT

Software systems have become larger and more complex with considerable dissemination in many industries. Due to the need to become more comprehensive, eventually, operationally and managerially independent systems tend to interoperate with themselves. This integration between independent systems with evolutionary nature and geographically distributed results a class of systems called System-of-Systems. Considering that the architecture of software defines a base structure of a system and the engineering of requirements identifies the necessary functionalities for a system, these activities are essential for any system, as a consequence, the identification of the stakeholders of a system becomes a primordial activity to elicit the necessary requirements and consequently build the system with the expected functionalities. This study presents a process and a model for the identification of the main stakeholders of Acknowledged System-of-Systems with the objective of minimizing the problems caused in the definitions of requirements system and software architecture in this category system.

Keywords: Stakeholder, Identify of Stakeholders, System-of-Systems, Requirements Engineering, Software Architecture, Process, Model.

LISTA DE TABELAS

Tabela 2.1 – Comparação entre Sistema Monolítico e SoS. Fonte: [37], p.2	24
Tabela 2.2 – Comparação dos aspectos. Fonte: [27] p.23	25
Tabela 4.1 – Perfis de <i>stakeholders</i> e seus interesses	47
Tabela 6.1 – Exemplo das respostas do modelo	70
Tabela 7.1 – Perfil dos Entrevistados na entrevista piloto	76
Tabela 7.2 – Perfil dos Entrevistados	79
Tabela A.1 – Trabalhos selecionados	102

LISTA DE SIGLAS

- ASC – Preocupações Arquiteturalmente Significantes, do inglês Architectural Separation of Concerns
- ASR – Requisitos Arquiteturalmente Significantes, do inglês Architecturally Significant Requirements
- DICODE – Colaboração e Tomada de Decisão com uso intensivo de dados, do inglês Data-intensive Collaboration and Decision Making
- DOD – Departamento de Defesa americano, do inglês Department of Defense
- DSAN – Matriz de Rede de Espaço Profundo, do inglês Deep Space Array Network
- ER – Engenharia de Requisitos, do inglês Requirements Engineering
- ES – Engenharia de Software, do inglês Software Engineering
- GEOSS – Sistemas de Sistemas de Observação Global da Terra, no inglês Global Earth Observation Systems of Systems
- PEP – Plano de Estudo e Pesquisa
- QP – Questão de Pesquisa
- RMM – Modelo de Monitoramento de Requisitos, do inglês Requirements Monitoring Model
- RSG – Grupo de Avaliação de Requisitos, do inglês Requirements Screening Group
- SAST – Superfície e Teste de Hipóteses Estratégicas, do inglês Strategic Assumptions Surfacing and Testing
- SNA – Análise de Redes Sociais, do inglês Social Network Analysis
- SISOS – Sistemas-de-Sistemas Intensivos de Software, do inglês Software-intensive SoS
- SOA – Arquitetura Orientada a Serviços, do inglês Service-Oriented Architecture
- SOS – Sistemas-de-Sistemas, do inglês System-of-Systems
- SOSE – Engenharia de Sistema-de-Sistemas, do inglês System-of-System Engineering
- SSMN – Rede de Mobilidade do Sistema Solar, do inglês Solar System Mobility Network
- TSOS – Sistema-de-Sistemas de Telemedicina, do inglês Telemedicine System of Systems
- ULS – Grande Escala, do inglês Ultra-Large-Scale
- UML – Linguagem de Modelagem Unificada, do inglês Unified Modeling Language

LISTA DE FIGURAS

Figura 1.1 – <i>Design</i> da pesquisa	16
Figura 3.1 – Atividades e Artefatos de um Modelo Geral de <i>Design</i> Arquitetural. Fonte: [29] p.113	29
Figura 3.2 – Modelo conceitual de arquitetura para SoS. Fonte: [25] p.3	31
Figura 3.3 – Meta-processo para SoS. Fonte: [26] p.4	32
Figura 4.1 – Comparação entre as abordagens de <i>design</i> .Fonte: [49] p.4	41
Figura 4.2 – Modelo de Freeman. Fonte: [23] p.292	43
Figura 4.3 – Modelo de Mitchell, Angle e Wood. Fonte: [45] p.874	44
Figura 4.4 – Uma hierarquia de <i>stakeholders</i> , clientes, usuários e classes de usuários. Fonte: [67] p.103	45
Figura 4.5 – Digrama do processo de identificação de <i>stakeholders</i> e comunicação com eles. Fonte: [55] p.2	55
Figura 6.1 – Diagrama de Atividades para uma solicitação	71
Figura 6.2 – Processo para construção do modelo	72
Figura 6.3 – Exemplo de representação gráfica do modelo de <i>stakeholders</i>	75
Figura 7.1 – Tempo de experiência dos entrevistados	80
Figura 7.2 – Cargos/Funções dos entrevistados	80
Figura 7.3 – Respostas da Pergunta 1 - A identificação dos <i>stakeholders</i> de um SoS é importante?	81
Figura 7.4 – Respostas da Pergunta 2 - Caso já tenha participado de um processo de identificação de <i>stakeholders</i> em SoS. Como esse processo ocorreu, quais técnicas foram utilizadas?	81
Figura 7.5 – Respostas da Pergunta 3 - SOMENTE as técnicas de identificação de requisitos existentes na engenharia de software tradicional, como as listadas pela ISO 1233, são suficientes para identificar os <i>stakeholders</i> de um SoS?	82
Figura 7.6 – Respostas da Pergunta 4 - É importante se ter um processo para realizar a identificação dos <i>stakeholders</i> de SoS?	84
Figura 7.7 – Respostas da Pergunta 5 - O Repositório de Requisitos, sendo construído de forma colaborativa, constitui em uma atividade viável de ser realizada no contexto de SoS <i>Acknowledged</i> ?	85
Figura 7.8 – Respostas da Pergunta 6 - O arquiteto do SoS ser o responsável pela seleção de ASC e ASR, constitui em uma atividade viável de ser realizada no contexto de SoS <i>Acknowledged</i> ?	86

Figura 7.9 – Respostas da Pergunta 7 - O arquiteto do SoS ser o responsável pela entrevista com os <i>stakeholders</i> que definem ASRs, constitui em uma atividade viável de ser realizada no contexto de SoS <i>Acknowledged</i> ?	87
Figura 7.10 – Respostas da Pergunta 8 - O resultado do processo é um modelo. Este modelo que representa o resultado da execução do processo é importante?	88
Figura 7.11 – Respostas da Pergunta 9 - Você acredita que este processo e o modelo resultante funcionariam num SoS real?	89
Figura 7.12 – Consolidação de respostas em escala Likert	90

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS E QUESTÃO DE PESQUISA	14
1.2	ESCOPO DO TRABALHO	15
1.3	METODOLOGIA DE PESQUISA	16
1.3.1	FASE 1: REVISÃO DA LITERATURA	16
1.3.2	FASE 2: PROPOSTA DO PROCESSO E DO MODELO	17
1.3.3	FASE 3: AVALIAÇÃO DO PROCESSO E DO MODELO	17
1.4	ESTRUTURA DO TRABALHO	18
2	SISTEMA-DE-SISTEMAS	19
2.1	DEFINIÇÕES DE SOS	19
2.2	TAXONOMIA E HEURÍSTICA DE SOS	20
2.3	EXEMPLOS DE SOS	22
2.4	DIFERENÇAS ENTRE SISTEMA MONOLÍTICO E SOS	24
3	ARQUITETURA DE SOFTWARE	27
3.1	ARQUITETURA DE SOFTWARE	27
3.2	ARQUITETURA DE SOFTWARE EM SOS	30
4	ENGENHARIA DE REQUISITOS E <i>STAKEHOLDERS</i>	34
4.1	ENGENHARIA DE REQUISITOS	34
4.2	ENGENHARIA DE REQUISITOS EM SOS	36
4.3	ABORDAGENS DE DESENVOLVIMENTO	37
4.4	ENGENHARIA DE REQUISITOS E ARQUITETURA DE SOFTWARE EM SOS	41
4.5	<i>STAKEHOLDERS</i>	43
4.6	<i>STAKEHOLDERS</i> E ARQUITETURA DE SOFTWARE	46
4.7	TRABALHOS RELACIONADOS COM PROCESSOS DE IDENTIFICAÇÃO DE <i>STAKEHOLDERS</i>	51
5	EMBASAMENTO PARA CONCEPÇÃO DO PROCESSO E MODELO	56
6	PROCESSO E MODELO DE IDENTIFICAÇÃO DOS PRINCIPAIS <i>STAKEHOLDERS</i> EM SOS <i>ACKNOWLEDGED</i>	67
6.1	PREMISSAS PARA CONSTRUÇÃO DO PROCESSO E MODELO	67

6.2	APRESENTAÇÃO DO PROCESSO	69
6.3	UTILIZAÇÃO DO PROCESSO PARA CRIAÇÃO DO MODELO	72
6.4	FORMATOS DE REPRESENTAÇÃO DO MODELO	74
7	AVALIAÇÃO DO PROCESSO E DO MODELO PROPOSTOS	76
7.1	PLANEJAMENTO	76
7.2	EXECUÇÃO	77
7.3	CARACTERIZAÇÃO DOS ENTREVISTADOS	78
7.4	RESULTADOS	78
7.5	COMENTÁRIOS E OBSERVAÇÕES GERAIS	90
8	CONSIDERAÇÕES FINAIS	93
8.1	CONCLUSÕES	93
8.2	TRABALHOS FUTUROS	94
	REFERÊNCIAS	95
	ANEXO A – Execução da Revisão da Literatura	101
	ANEXO B – Questionário aos Especialistas e Arquitetos de SoS	106
B.1	TERMO DE CONSENTIMENTO	106
B.2	DADOS DO PERFIL DO PARTICIPANTE	107
B.3	QUESTIONÁRIO PARTE 1 - SOBRE A NECESSIDADE	110
B.4	QUESTIONÁRIO PARTE 2 - SOBRE O PROCESSO E MODELO PROPOSTOS ...	112

1. INTRODUÇÃO

Atualmente, sistemas de software tem-se tornado maiores e mais complexos com considerável disseminação em diversos setores [47]. Devido à necessidade de tornarem-se mais abrangentes, em determinadas vezes, sistemas operacionalmente independentes se integram [47]. Esta nova classe emergente de sistemas, constituída pela integração de sistemas independentes, se distingue de grandes sistemas monolíticos pela independência de seus componentes, natureza evolutiva, comportamento emergente e extensão geográfica que limita a troca de informações entre seus componentes [42]. A esta nova classe de sistemas chama-se Sistema-de-Sistemas (SoS do inglês *System-of-Systems*). Exemplos de SoS podem ser encontrados em diversas áreas, como por exemplo sistemas de aeroportos, automotivos, aviônicos, robóticos e médicos [47]. Um bom exemplo de um SoS é um sistema médico complexo que integra sistemas de diagnóstico, de tratamento e de gerenciamento de pacientes.

Não distante, as arquiteturas de software desempenham um importante papel na qualidade de um sistema, uma vez que constituem a espinha dorsal do mesmo sistema [47]. Dessa forma, a atenção dada pelas arquiteturas dos sistemas que envolvem um SoS é fundamental para o seu sucesso [47]. Por sua vez, a Engenharia de Requisitos (ER), é uma atividade essencial que compõe a engenharia de sistemas tradicional para identificar e gerenciar funcionalidades, atributos físicos e interfaces necessárias para um sistema [48]. Assim, tanto os requisitos quanto a arquitetura são primordiais para o sucesso de um software, demonstrando uma forte relação entre elas.

A qualidade dos requisitos de software afeta a qualidade da arquitetura de um software [12]. Se um documento de requisitos possui problemas como requisitos incorretos, conflitantes, ambíguos ou com falta de detalhes, este produzirá uma arquitetura de software com erros e problemas [12]. Como os requisitos são oriundos das informações dos *stakeholders* relevantes para depois serem traduzidos para requisitos técnicos [27], fica evidente a importância de ter acesso aos *stakeholders* relevantes do sistema no momento da elicitación dos requisitos. A definição de um conjunto incompleto de *stakeholders* pode levar a resultados incompletos [50]. E em SoS estes problemas se tornam ainda mais latentes, uma vez que SoS frequentemente envolvem um grande número de *stakeholders* e com interesses diversos. Isto faz com que bons métodos conhecidos de identificação de *stakeholders* possam funcionar em projetos pequenos ou monolíticos mas ficam aquém dos projetos de SoS e em projetos interorganizacionais devido a incapacidade de atender a complexidade e ao grande número de *stakeholders* envolvidos [41] [3].

Finalmente, requisitos de SoS são frequentemente expressos em termos mais amplos, exigindo engajamento da equipe com o gerente, *stakeholders* e usuários do SoS [27]. Este engajamento ajudará a derivar os requisitos a partir das amplas necessidades e então abordá-los utilizando as funcionalidades dos sistemas constituintes, denominação utilizada para indicar um sistema independente que compõe o SoS. Dessa maneira, a equipe do SoS tem o desafio adicional de trabalhar com os sistemas constituintes e conhecer os principais *stakeholders* envolvidos com o SoS para então compreender seu contexto organizacional e ter o perfil de cada *stakeholder* e usuário no SoS ao

longo do tempo. Porém, para isso, precisa-se de um processo adequado que atenda as necessidades, compreenda as características e dificuldades de um SoS e seja reutilizável por outros SoS.

1.1 Objetivos e Questão de Pesquisa

Inicialmente o objetivo desta pesquisa era a identificação dos principais *stakeholders* de SoS e seu papel no desenvolvimento e evolução do mesmo. Autores como Pouloudi e Whitley [53] ressaltam que é reconhecida a importância de ter-se processos de identificação de *stakeholders*, que os *stakeholders* emergem como um produto final do processo de identificação, mas esses processos de identificação não são explicados. Outros autores como Lim e Ncube[41] e Ballejos e Montagna [3] afirmam que abordagens tradicionais para identificação de *stakeholders* podem funcionar em projetos pequenos ou monolíticos mas ficam aquém dos projetos de SoS e em projetos interorganizacionais. Keating [37] e Ncube [48] ressaltam que a Engenharia de Sistema-de-Sistemas (SoSE do inglês *System-of-System Engineering*) trata de um domínio de problema que está destinado modificar a forma tradicional de pensar sobre engenharia de sistemas complexos, assim como destinado a mudar a forma de pensar sobre os requisitos. Por este motivo, o sucesso obtido pela ER na Engenharia de Software(ES) tradicional pode não ter o mesmo sucesso na SoSE devido as divergências que aparecem em todas as áreas elencadas. Ncube ainda afirma que são necessários novos processos de ER, métodos de gestão, técnicas e ferramentas que possam responder dinamicamente a requisitos instáveis, fragmentados e em constante mudança [48].

Assim, percebeu-se que havia a necessidade de definição de um processo para identificação dos principais *stakeholders* de SoS e um modelo para apresentação do resultado gerado pelo processo. Porém, durante a execução do trabalho percebeu-se que o escopo estava muito amplo. O termo principais *stakeholders* precisava estar mais claro. Então, em reuniões de orientação e com a colega da Universidade de São Paulo (USP-ICMC) Cristiane Aparecida Lana, que conduz uma tese de doutorado que pretende estabelecer um *framework* interativo composto por método para a elicitação de requisitos de SoS, um método para validação de requisitos de SoS, decidiu-se que os principais *stakeholders* são os *stakeholders* que tem impacto no *design* da Arquitetura. Esta decisão de selecionar estes como os principais *stakeholders* do SoS tem origem no trabalho de Nakagawa, Gonçalves, Guessi, Oliveira e Oquedo [47] que afirmam que para se ter sucesso na construção de um SoS é fundamental dar atenção a arquitetura. Então identificou-se que *stakeholders* que impactam a arquitetura de um SoS são os *stakeholders* que estão envolvidos na definição dos requisitos de software que influenciam na arquitetura, esses chamados de Requisitos Arquiteturalmente Significantes (ASRs).

Também os diversos tipos de SoS possuem estruturas e comportamentos muito distintos, isso dificulta a criação do processo e modelo de identificação dos principais *stakeholders*, uma vez que uma premissa que define um tipo de SoS acabava por invalidando outra. Então, em conjunto com Professora Dra. Sabrina dos Santos Marczak e com a colega Cristiane Aparecida Lana, optou-se para em primeiro momento, o modelo proposto focar num dos tipos de SoS. Foi escolhido o

Acknowledged por ser o tipo mais utilizado e que tem maior crescimento de utilização comparado aos demais.

Assim, o objetivo da pesquisa é definir um modelo teórico para identificar os principais *stakeholders* de SoS *Acknowledged*, os *stakeholders* que estão envolvidos na definição dos ASRs. Para alcançar este objetivo definiu-se a questão de pesquisa:

Questão de Pesquisa (QP):

Como definir os principais *stakeholders* de um SoS *Acknowledged*?

Para responder a essa questão de pesquisa refinou-se o *design* da pesquisa. Este *design* define a pesquisa como uma abordagem mista, baseada em revisão da literatura para a proposição do processo e modelo de identificação dos principais *stakeholders* de SoS *Acknowledged* e empírica para a avaliação prévia do mesmo, buscando considerar a experiência prática de especialistas sobre o assunto. Para responder a questão de pesquisa, definiu-se os objetivos específicos listados abaixo:

- Pesquisar os principais conceitos de SoS, *stakeholders*, arquitetura de sistemas e engenharia de requisitos.
- Entender a relação entre os conceitos aplicados ao contexto de SoS.
- Investigar os processos e modelos existentes de identificação de *stakeholder* em sistemas tradicionais, distribuídos, inter-organizacionais, entre outros.
- Estabelecer as características que definem o perfil dos principais *stakeholders* de SoS *Acknowledged*.
- Estabelecer de que forma pode-se identificar os *stakeholders* que definem os ASRs de um SoS *Acknowledged*.
- Estabelecer um processo para identificação dos principais *stakeholders* de SoS *Acknowledged* e modelo que apresente o resultado gerado pelo processo.
- Avaliar com especialistas o processo e modelo propostos.
- Refinar o processo e modelo propostos baseado nos resultados da avaliação preliminar.

1.2 Escopo do Trabalho

Esta seção descreve os limites deste trabalho, para adequar essa pesquisa dentro dos prazos do programa de mestrado. Além disso, são apresentadas algumas variáveis extras que poderiam ser pesquisadas em trabalhos futuros.

Este trabalho assume que exista atividades de colaboração entre os principais *stakeholders* do SoS e a equipe do SoS durante a elicitaco de requisitos. Tambm devido as diferenas entre as categorias de SoS, decidiu-se por focar na categoria que fosse mais abrangente escolhendo SoS *Acknowledged*, seguindo os requisitos e configurao desta categoria. Tambm se pode dizer que o processo e modelo propostos no resolvem conflitos entre *stakeholders*, muito menos realiza a priorizao entre os requisitos. Porm o processo proposto de identificao dos principais *stakeholders* possui caractersticas como o repositrio de requisitos e listagem de respostas coletadas nas entrevistas que podem tornar-se insumos para que processos de resoluo de conflitos sejam desenvolvidos e tambm processos que faam com que os *stakeholders* sejam devidamente priorizados.

1.3 Metodologia de Pesquisa

A metodologia de pesquisa adotada foi dividida em 3 fases. A primeira fase consiste em uma reviso da literatura. A segunda fase tem-se a construo do processo e modelo para identificao dos principais *stakeholders* de SoS *Acknowledged*. E a terceira e ltima fase apresenta-se a avaliao preliminar do processo e modelo por especialistas. O *design* da pesquisa, que implementou estas trs fases,  apresentado na Figura 1.1.

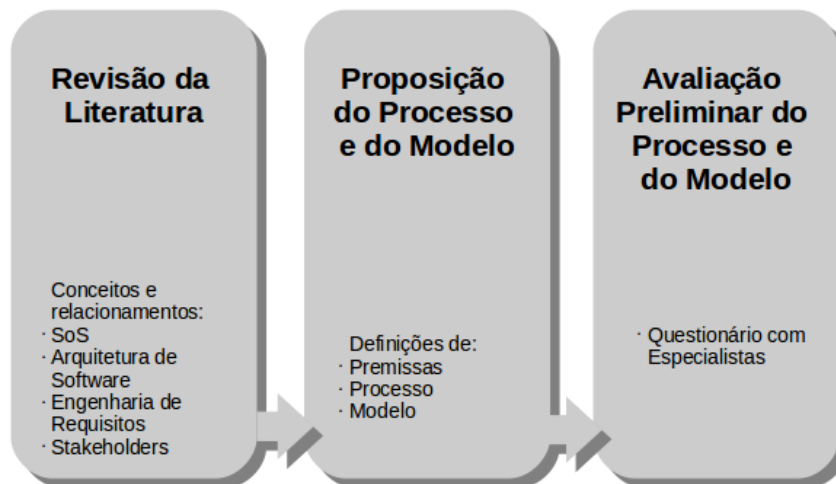


Figura 1.1 – *Design* da pesquisa

1.3.1 Fase 1: Reviso da Literatura

A primeira etapa da primeira fase foi definida como exploratria e documental, considerando que foca no entendimento das definies de *stakeholders*, engenharia de requisitos, arquitetura de software, SoS e os inter-relacionamentos desses conceitos, com o objetivo de buscar a compreenso do estado da arte para formar conhecimento base para execuo da pesquisa. A partir desse conhecimento exploratrio, na segunda etapa realizou-se uma Reviso da Literatura, atravs de

leitura de livros e artigos relacionados ao tema que satisfazem os critérios de seleção definidos previamente a execução da mesma.

1.3.2 Fase 2: Proposta do Processo e do Modelo

Após a execução da Revisão da Literatura, foram extraídas e descritas as informações relevantes dos trabalhos. Com base no referencial teórico selecionado, em especial nos trechos destacados na Seção 6.1, foi proposto um processo para identificação dos principais *stakeholders* de SoS *Acknowledged* e um modelo para apresentação do resultado do processo. Tanto o processo quanto o modelo se baseiam em premissas extraídas da Revisão de Literatura para alcançar seu objetivo. Através do processo estruturado que pode ser reproduzido em qualquer SoS *Acknowledged* os principais *stakeholders* são identificados bem como seu envolvimento com os sistemas constituintes. Por fim, os resultados da execução deste processo podem ser representados de forma gráfica, demonstrando em uma espécie de mapa de *stakeholders* do SoS *Acknowledged* avaliado através do modelo de identificação dos principais *stakeholders*.

Para a elaboração do processo foi preciso identificar claramente as características que compõe um SoS *Acknowledged*. De acordo com DoD [27], as características específicas do SoS *Acknowledged* são: os sistemas constituintes reconhecem os objetivos do SoS; e o SoS *Acknowledged* possui gestão e equipe própria. Assim, respeitando essas características e com base na Revisão de Literatura, estruturou-se um processo e um modelo que serão apresentados no Capítulo 6.

1.3.3 Fase 3: Avaliação do Processo e do Modelo

Por fim, a terceira e última fase do trabalho, é a avaliação do processo e do modelo propostos. O objetivo desta fase é avaliar cada uma das etapas do processo e um exemplo de modelo gerado com profissionais e especialistas em SoS.

Nesta avaliação preliminar, o primeiro passo foi a apresentação dos conceitos utilizados no trabalho e a explicação do objetivo da pesquisa através de reuniões com cada um dos entrevistados. Após, um questionário foi entregue para o respondente e durante o tempo de respostas, todo o tipo de esclarecimento foi prestado, com o objetivo de não ter nenhum tipo de viés de entendimento das questões, do processo ou do modelo. Assim minimizou-se qualquer dúvida entre o que está sendo proposto e o que é necessário a um SoS *Acknowledged* de acordo com os especialistas.

Os resultados das respostas das entrevistas com os especialistas foram analisados e refinados. A etapa de análise foi composta da consolidação e exibição das respostas para às perguntas. A etapa de refinamento compreendeu apontar as críticas ao processo e modelo propostos numa possível aplicação de um SoS *Acknowledged*. Assim, possibilitando uma autoavaliação dos artefatos propostos com o intuito de atestar a identificação das necessidades do SoS *Acknowledged*. A avaliação preliminar organizada por entrevistas e seus resultados é apresentada no Capítulo 7.

1.4 Estrutura do Trabalho

Este trabalho está organizado em 8 capítulos, sendo o Capítulo 1 (esse capítulo) descrevendo os objetivos, a questão de pesquisa, a metodologia de pesquisa e definindo o escopo do trabalho. O Capítulo 2 apresenta o referencial teórico sobre SoS, trazendo definições, taxonomia, heurísticas, exemplos e a diferença entre um sistema monolítico e um SoS. O Capítulo 3 apresenta o referencial teórico sobre Arquitetura de Software. O Capítulo 4 apresenta conceitos da Engenharia de Requisitos, abordagens de desenvolvimento de sistemas, a relação entre Engenharia de Requisitos e Arquitetura de Software, definições de *stakeholders*, a relação de *stakeholders* e Arquitetura de Software e os trabalhos relacionados com a identificação de *stakeholders* em diversos tipos de sistemas. O Capítulo 5 apresenta as informações extraídas da literatura que embasam a criação do processo e modelo de identificação dos principais *stakeholders* de SoS *Acknowledged*. O Capítulo 6 apresenta o processo e modelo para identificação dos principais *stakeholders* de SoS *Acknowledged*, com as premissas consideradas para a elaboração do processo, bem como a explicação de cada uma das etapas do processo e modelo. O Capítulo 7 apresenta a avaliação preliminar do processo e do modelo para identificação dos principais *stakeholders* de SoS *Acknowledged*, detalhando o planejamento, realização e os resultados deste estudo. E, por fim, o Capítulo 8 que apresenta as considerações finais e trabalhos futuros.

2. SISTEMA-DE-SISTEMAS

Este capítulo apresenta as definições dos conceitos presentes no trabalho e que serão utilizados na pesquisa. Na Seção 2.1, apresenta-se as definições de SoS; na Seção 2.2, apresenta-se a taxonomia e heurísticas de SoS; na Seção 2.3, apresenta-se exemplos de SoS; e na Seção 2.4, apresenta-se as diferenças entre Sistemas Monolíticos e SoS.

2.1 Definições de SoS

Pesquisas na área de SoS vem crescendo com grande velocidade na última década. O mapeamento sistemático da Engenharia de Sistema-de-Sistemas (SoSE, do inglês System-of-System Engineering) de Axelsson [2] demonstra esse crescimento. Este mapeamento, executado no início de 2015, apresenta que até 2003 haviam menos de 50 publicações por ano e que a partir desta data as publicações aumentaram significativamente chegando a mais de 350 por ano em 2006 e 2012. Também se pode salientar que entre os principais resultados deste mapeamento, está a necessidade de melhorar o entendimento dos pesquisadores sobre os princípios chave de SoS, demonstrando que não há clareza no entendimento dos mesmos do que é um SoS.

Portanto, ainda existem muitas definições e interpretações do que são SoS que geram divergências entre os pesquisadores. As definições utilizadas neste trabalho são do Departamento de Defesa Americano (DoD) [27] e de Maier [42]. De acordo com o DoD [27], SoS é um conjunto ou arranjo de sistemas que resultam, quando independentes e úteis, em sistemas integrados que entregam capacidades únicas. Maier [42] afirma que o termo SoS é comumente utilizado sugerindo uma coleção de componentes que são significativamente complexos que podem ser considerados sistemas e que agrupados formam um sistema maior. A esta integração colaborativa dos sistemas, Maier propõe o nome de "*system-of-systems*". E por sua vez, aos sistemas independentes que compõem o SoS, pode-se denominar de sistemas constituintes ou sistemas componentes.

Para poderem ser chamados de SoS, estes sistemas devem respeitar algumas características principais. Inicialmente foram definidas por Maier [42] cinco características: independência operacional, independência gerencial, desenvolvimento evolucionário, comportamento emergente, e distribuição geográfica. Além dessas cinco, Sommerville [63] sugere adicionar mais duas: intensividade de dados e heterogeneidade. Estas características são apresentadas a seguir:

- **Independência Operacional:** se um SoS é desfeito, cada sistema que fazia parte do SoS deve conseguir operar normalmente de forma independente.
- **Independência Gerencial:** o componente de um sistema não só pode operar independentemente como opera independentemente. Os componentes dos sistemas são adquiridos e integrados ao SoS, mas mantém a operação contínua independente do SoS.

- **Desenvolvimento Evolucionário:** O SoS não surge totalmente formado. Seu desenvolvimento e existência é evolucionário, com inclusões, remoções e modificações de funções e propósitos conforme a experiência.
- **Comportamento Emergente:** O SoS executa funções e propósitos que não estão em nenhum sistema constituinte. Esses comportamentos são propriedades emergentes de todo o SoS e não podem ser localizados em nenhum sistema constituinte. Os principais propósitos dos SoSs são cumpridos por esses comportamentos.
- **Distribuição Geográfica:** A extensão geográfica dos sistemas constituintes é grande. Grande é um conceito nebuloso e relativo à medida que as capacidades de comunicação aumentam, mas no mínimo, significa que os componentes podem facilmente trocar apenas informações.
- **Intensividade de Dados:** Como um SoS normalmente depende e gerencia um grande volume de dados, em termos de tamanho isso pode ser dezenas ou centenas de vezes maior que o código de um sistema constituinte.
- **Heterogeneidade:** É improvável que todos os SoS sejam desenvolvidos usando a mesma linguagem de programação ou método de *design*. Esta é uma consequência do ritmo muito rápido de evolução das tecnologias de software. As empresas frequentemente atualizam seus métodos e ferramentas de desenvolvimento à medida que novas versões melhoradas estão disponíveis.

Assim um sistema que tem subsistemas complexos porém não possui as características acima não é considerado um SoS. Por exemplo: a rede de defesa aérea sem uma única autoridade de comando é um SoS, enquanto que um sistema de apenas um míssil individual não pode ser considerado SoS.

2.2 Taxonomia e Heurística de SoS

A taxonomia é necessária para possibilitar a classificação de SoS, bem como para facilitar a identificação do que é um SoS. A taxonomia para a identificação de um SoS possibilita diferenciar um verdadeiro SoS de um grande e complexo sistema porém monolítico [43]. Maier [43] afirma que apenas as características de um SoS não são classificações taxonômicas adequadas. A classificação taxonômica adequada precisa ser mais próxima do desenvolvimento [43]. Com isso, Maier dividiu os SoS em três categorias distinguindo-as principalmente pelo controle gerencial: *Virtual*, *Collaborative* e *Directed*. Mais tarde, o DoD através de Dahmann e Baldwin [17] acrescentou mais uma categoria a esta lista: *Acknowledged*. Cada uma destas categorias está descrita a seguir, porém, deve-se salientar que um SoS, em algumas situações, pode conter comportamentos de mais de uma categoria de SoS.

- **Virtual:** são sistemas sem gerenciamento central e os participantes podem não reconhecer o objetivo global do sistema.

- **Collaborative:** são sistemas que são constituídos por sistemas independentes e são proprietários e gerenciados por diferentes organizações.
- **Directed:** são propriedade de uma única organização e são desenvolvidos pela integração de sistemas que também são pertencentes a essa organização.
- **Acknowledged:** são sistemas que reconhecem os objetivos do SoS, possuem um gerente/gestor e recursos para o SoS. Porém os sistemas que fazem parte do SoS tem independência de propriedade, objetivos, financiamento, desenvolvimento e sustentação. As mudanças nos sistemas são baseadas na colaboração entre o SoS e o sistema.

Assim, na categoria *Virtual* a ausência de gerenciamento indica que os sistemas constituintes são independentes e descoordenados porém as integrações entre eles constitui um SoS. Já na categoria *Collaborative*, o gerenciamento distribuído entre diferentes organizações oferece padrões para que ocorra a colaboração entre os sistemas constituintes. Na categoria *Directed* o gerenciamento central especifica os principais propósitos e direciona os sistemas constituintes. Por fim, na categoria *Acknowledged* o gerenciamento é baseado na negociação entre sistemas constituintes e o SoS, uma vez que são compostos por diferentes organizações.

Conforme o DoD [27], existe um número crescente de SoS *Acknowledged* e assim como os SoS *Directed* esses sistemas tem recursos próprios a nível de SoS. Como o SoS *Acknowledged* compreende sistemas que mantêm objetivos, gerenciamento e recursos independentes durante o ciclo de desenvolvimento, então o processo acaba sendo altamente colaborativo, o que torna a principal distinção entre o SoS *Acknowledged* e o SoS *Directed*, pois o SoS *Acknowledged* tem seu modo de funcionamento normal não subordinado a um gerenciamento central. Além disso, como muitos SoS *Acknowledged* não tem autoridade sobre os sistemas constituintes, eles conduzem seus objetivos através de oportunidade de elevar o desenvolvimento do sistema constituinte para alcançar objetivos que sozinho este sistema não conseguiria ou teria muita dificuldade.

Para objetivar boas práticas e padrões, precisa-se definir heurísticas. Com esse objetivo, Maier [43] apresenta heurísticas de SoS a partir do refinamento das heurísticas propostas por Rehtin [56]. As heurísticas propostas por Maier estão listadas abaixo:

- **Stable Intermediate Forms:** Sistemas complexos se desenvolverão e evoluirão em uma arquitetura geral muito mais rapidamente se houver formas intermediárias estáveis do que se não houver [43]. Esta heurística afirma que há uma arquitetura de transição entre o estado atual e a arquitetura alvo do SoS. Como um SoS não é um sistema monolítico, ele depende da colaboração dos demais sistemas para ser constituído, propõe-se arquiteturas intermediárias que são temporárias e evolutivas com o objetivo de chegar no desenho de arquitetura final.
- **Policy Triage:** Seleciona e dispõe de suporte a componentes do SoS, ignorando quem pode se recuperar sozinho e tratando apenas aqueles que sem ajuda morrerão [43]. Uma equipe de desenvolvimento em um SoS não tem o controle completo sobre o desenvolvimento ou

modo de operação do sistema final. Por isso a equipe de arquitetura do SoS necessita fazer a triagem para distinguir as principais características que precisam ser abordadas.

- **Leverage at the Interfaces:** Discussão do poder das interfaces. A maior influência em uma arquitetura de sistemas está na interface, os maiores perigos também [43]. Historicamente, tem-se maior sucesso dando maior atenção as interfaces do que aos componentes. Mas os componentes de um SoS são altamente independentes, gerenciais e operacionais, a arquitetura do SoS torna-se as interfaces.
- **Ensuring Cooperation:** Se um sistema requer colaboração voluntária, o mecanismo e os incentivos para essa colaboração devem ser planejados [43]. Os custos e benefícios da colaboração devem ser superiores aos custos e benefícios da independência.
- **Primacy of Communications:** A comunicação é a principal tecnologia que habilita a construção de um SoS. O comportamento emergente pode surgir somente se a troca de informações entre os sistemas for suficiente [43]. Se os elementos são produzidos semi-independentes os padrões de comunicação são mais importantes do que qualquer sistema. Ou seja, nenhum SoS pode existir sem comunicação.

2.3 Exemplos de SoS

A revisão da literatura de Axelsson [2] mostra que até hoje as pesquisas sobre SoS são dominadas por pesquisadores norte-americanos com foco em aplicações militares e espaciais. Porém, Axelsson sinaliza o início do desenvolvimento por outras partes da sociedade nos últimos anos, com pesquisas e desenvolvimentos de sistemas *cyber-physical*, Internet das Coisas e ecossistemas de software, que se relacionam com SoS. Por estas diversidades entende-se a afirmação de Muller [46] que a maioria dos sistemas atuais é realmente um sistema constituinte de um SoS.

Os exemplos que Maier [43] e DoD [27] apresentam representações amplas e por vezes distantes da ES como por exemplo a internet exemplificando um SoS *Collaborative* e a *World Wide Web* e economias nacionais como exemplos de SoS Virtual. Sommerville [63] procura exemplos mais próximos de sistemas de software e cita alguns como: um sistema de gerenciamento de nuvem que controla o gerenciamento de nuvem privada e gerenciamento de servidores em nuvens públicas, como Amazon e Microsoft; um sistema *on-line* bancário que controla solicitações de empréstimo e que se conecta a um sistema de referência de crédito fornecido por uma agência de referência de crédito para verificar o crédito dos candidatos; um sistema de informação de emergência que integra informações da polícia, ambulâncias, bombeiros e guarda costeira sobre os recursos disponíveis para tratar emergências, como inundações e acidentes em grandes proporções; um sistema de aprendizagem digital que oferece uma variedade de suporte de aprendizagem integrando sistemas de software distintos, como o Microsoft Office 365, ambientes de aprendizagem virtual, como o Moodle, fer-

ramentas de conteúdo, como arquivos de jornais. Assim, buscou-se na literatura exemplos de SoS reais, listados abaixo:

Global Earth Observation Systems of Systems (GEOSS): SoS constituído por um conjunto de sistemas de observação, informação e processamento da Terra que interagem e provem acesso a diversas informações para uma ampla gama de usuários nos setores público e privado [24]. Os últimos dados apontam que o GEOSS está presente em cerca de 90 países e 67 organizações participantes, tem o objetivo de aumentar a relevância das observações da Terra para questões globais através da conexão dos produtores de dados ambientais e de ferramentas de apoio à decisão. Entrega dados para cerca de 186 países.

Solar System Mobility Network (SSMN): um SoS no contexto de exploração espacial, por enquanto apenas o conceito. Tal conceito do SSMN baseia-se na hipótese de que, através da distribuição de capacidades de suporte logístico para missões de exploração espacial, maior eficácia pode ser alcançada no futuro programa nacional de exploração espacial através das simulações de atividades humanas e robóticas [62]. Assim, este SoS utiliza simulações de sistemas e atividades humanas e robóticas na exploração do sistema solar.

Deep Space Array Network (DSAN): um SoS projetado para ser a próxima geração de redes de antenas terrestres que fornecem serviços de comunicação para as missões espaciais da NASA [20]. Atualmente a rede consiste de 1.200 antenas de recepção de 12 metros, divididas igualmente entre três longitudes ("regiões"), a fim de manter o rastreamento à medida que a Terra gira. O DSAN emprega processamento de sinal digital para combinar sinais de uma série de antenas para alcançar a abertura efetiva de uma antena muito maior [20]. Com 400 antenas em cada uma das três regiões, inúmeras aeronaves podem ser rastreadas ao mesmo tempo, alocando grupos de antenas para cada alvo e roteando seus sinais adequadamente através do sistema de processamento de sinal [20]. O DSAN é composto por vários sistemas complexos, como por exemplo: matriz de recebimento; matriz de transmissão; telemetria; monitoramento e controle das comunicações entre os sistemas.

Telemedicine System of Systems (TSoS): um SoS utilizado para monitorar pacientes em suas casas que estão distribuídos em grandes áreas geográficas e geralmente incluem componentes com diferentes características operacionais. De acordo com Petcu e Petrescu [52], estes SoS de telemedicina possuem componentes como: monitoramento de pacientes, atendimento hospitalar, orientação de serviço de ambulância para intervenção e assistência de paciente em deslocamento, primeiro atendimento de emergência. Conforme os autores, o sistema permanece monitorando os sinais vitais do paciente, bem como dados relacionados aos padrões de atividades diárias (quando aplicável) e transmite esses sinais para uma central. Esta central de monitoramento é responsável por fazer a análise de dados, avaliação de dados, arquivamento e representação de dados do paciente expondo as informações para os *stakeholders*.

Pode-se dizer que estes exemplos se encaixam bem nas categorias de SoS *Acknowledged* e *Directed* pois ou tem equipe própria do SoS com sistemas constituintes de diferentes organizações ou tem os sistemas constituintes e o SoS pertencendo a mesma organização. Exemplos reais de

Tabela 2.1 – Comparação entre Sistema Monolítico e SoS. Fonte: [37], p.2

Comparação	Sistema Monolítico	SoS
Foco	Único Sistema Complexo	Integração de Múltiplos Sistemas Complexos
Problema	Bem definido	Emergente
Fronteira	Fixa e Definida	Dinâmica
Expectativa do Stakeholder	Solução	Resposta Inicial
Ciclo de Vida	Previsível	Evolucionário e Contínuo

SoS *Virtual* e *Collaborative* não foram encontrados na literatura, porém outros exemplos, além dos propostos por Maier, de SoS *Virtual* podem ser GRIDs computacionais colaborativos e de SoS *Collaborative* podem ser comunidades de interesses como projetos *Open Source*. Gerrit Muller [46], em seu recente trabalho, afirma que muitos SoSs são sistemas constituintes de um SoS, ou seja, que muitos SoS que são compostos por outros SoS. Esta configuração acaba possibilitando que ocorram fusões de categorias de SoS dificultando mais ainda esta categorização.

2.4 Diferenças entre Sistema Monolítico e SoS

Um grande e complexo sistema monolítico se difere de um SoS em diversos aspectos. Para apresentar as diferenças entre eles, compara-se através de uma adaptação da comparação feita por Keating [36], que comparou a Engenharia de Sistemas tradicional com a Engenharia de SoS, conforme a Tabela 2.1.

Conforme Keating [36], enquanto um Sistema Monolítico é composto por um único sistema, um SoS é constituído pela integração de sistemas complexos com gerenciamento e operação independentes. O problema a ser solucionado num Sistema Monolítico tem escopo bem detalhado, definido e objetivos relativamente claros, já em SoS essas condições são a exceção dos problemas que normalmente emergem do contexto onde o SoS está inserido. Como o SoS entrega capacidades que os sistemas constituintes não conseguem realizar sozinhos, os problemas que surgem nem sempre são bem claros e definidos. Quanto as fronteiras do sistema, num Sistema Monolítico essas são claras, fixas e definidas, correspondem exclusivamente aos limites do sistema. Por sua vez, num SoS as fronteiras são dinâmicas com base nos sistemas constituintes que fazem parte do SoS e podem mudar conforme ocorrerem mudanças nos sistemas constituintes do SoS ou alterações como inclusão ou remoção de sistemas constituintes que compõe o SoS. A expectativa do *stakeholder* num Sistema Monolítico é com a resolução de um problema, a implementação de uma solução. Num SoS, a expectativa do *stakeholder* está na resposta inicial que a comunicação entre os sistemas constituintes vai gerar, uma vez que a criação de novas comunicações podem gerar novas necessidades direcionando as implementações para problemas complexos e a solução final não é necessariamente a expectativa final. O Ciclo de Vida de um Sistema Monolítico pode ser previsível

e definido pelo proprietário do sistema. Num SoS, a constante evolução que está submetido devido seu comportamento e características fazem com que seu ciclo de vida não seja facilmente previsível.

Ainda assim, com o intuito de detalhar mais a comparação entre os diversos níveis de um Sistema Monolítico e um SoS *Acknowledged* tem-se a Tabela 2.2, desenvolvida pelo DoD [27]. Nesta tabela tem-se a aplicação de cada princípio da ES e por sua vez, cada princípio possui aspectos que estão sendo comparados.

Analisando as comparações da Tabela 2.2, percebe-se que a comunicação é o fator mais importante em um SoS. Maier [42] ressalta que SoS são definidos por padrões de comunicação. Além disso, afirma que na maioria dos SoS a arquitetura do SoS é a comunicação. Assim ratificando o que Nakagawa [47] afirma, para o sucesso de um SoS é fundamental dar atenção à arquitetura de software.

Tabela 2.2 – Comparação dos aspectos. Fonte: [27] p.23

Aspecto de Desenvolvimento	Sistema Monolítico	SoS <i>Acknowledged</i>
Gestão e Supervisão		
Envolvimento dos <i>Stakeholders</i>	Claro conjunto de <i>stakeholders</i>	<i>Stakeholders</i> de sistemas constituintes e de SoS (incluindo os proprietários dos sistemas) com competição de interesses e prioridades. Em alguns casos os <i>stakeholders</i> do sistema constituinte não tem nenhum interesse no SoS. Nem todos os <i>stakeholders</i> podem ser reconhecidos.
Governança	Gerenciamento do Projeto e Financiamento alinhados	Mais níveis de complexidade devido à gestão e financiamento tanto para o SoS e sistemas constituintes. SoS não tem autoridade sobre todos os sistemas constituintes.
Ambiente Operacional		
Foco Operacional	<i>Design</i> e Desenvolvimento para atender objetivos operacionais	Conhecido por alcançar um conjunto de objetivos operacionais usando sistemas constituintes cujos objetivos podem não estar alinhados com os objetivos do SoS.
Implementação		
Aquisição	Marcos alinhados e requisitos documentados	Maior complexidade devido aos múltiplos ciclos de vida do sistema em todos os sistemas legados, sistemas em desenvolvimento, novos desenvolvimentos e inserção de tecnologias. Normalmente têm objetivos de capacidade inicial que podem precisar ser traduzidos em requisitos formais.

Teste e Avaliação	Testes e avaliações do sistema são geralmente possíveis	O teste é mais desafiador devido à dificuldade de sincronização em vários ciclos de vida de sistemas. Dada a complexidade de todas as partes e o potencial de problemas para consequências não intencionais.
Considerações de Engenharia e Design		
Fronteiras e Interfaces	Foco em interfaces e nas fronteiras para um sistema único	Com foco em identificar os sistemas que contribuem para os objetivos de SoS e permitindo o fluxo de dados, controle e funcionalidade em todo o SoS enquanto equilibra as necessidades dos sistemas.
Performance e Comportamento	Desempenho do sistema para atender a objetivos específicos	Desempenho do SoS que satisfaz as necessidades de capacidade do usuário SoS enquanto equilibra as necessidades dos sistemas constituintes.

3. ARQUITETURA DE SOFTWARE

Este capítulo apresenta as definições dos conceitos de Arquitetura de Software presentes no trabalho. Na Seção 3.1, apresenta-se as definições de Arquitetura de Software; e na Seção 3.2, apresenta-se as definições de Arquitetura de Software em SoS.

3.1 Arquitetura de Software

Arquitetura de sistemas de software é a organização essencial, que unifica conceitos e princípios de um sistema incluindo seus componentes, seus relacionamentos entre os componentes e com o ambiente (desenvolvimento, operacional, etc) e seus princípios que orientam seu *design* e evolução [34]. Bass [5] define de uma forma mais direta, afirmando que arquitetura de sistemas de software é o conjunto de estruturas necessárias para pensar no sistema, que inclui elementos de software e como esses elementos se relacionam. Nestas definições, percebe-se a importância da arquitetura para um sistema, não existe sistema que não tenha uma arquitetura.

Devido a complexidade crescente dos sistemas e conseqüentemente da arquitetura de sistemas de software, esforços foram realizados para a criação de uma ISO de conceitos e padrões de arquitetura de sistemas de software. Essa ISO 42.010 [34] estabelece além de conceitos, um modelo conceitual de arquitetura. Os principais conceitos definidos neste padrão estão descritos abaixo:

- **Sistema de *Software-intensive***: Qualquer sistema onde o software contribua com essencial influência no *design*, construção, implantação e evolução do sistema como um todo, englobando aplicações individuais, sistemas tradicionais, subsistemas, sistemas de sistemas, linhas de produtos, famílias de produtos, empresas inteiras e outras agregações de interesse.
- **Sistema de Interesse**: Faz referência ao sistema cuja arquitetura está sendo considerada na preparação de uma descrição de arquitetura.
- **Descrição de Arquitetura**: Produto de trabalho usado para expressar uma arquitetura.
- **Visões de Arquitetura**: Produto de trabalho que expressa a arquitetura de um sistema na perspectiva das preocupações específicas do sistema.
- **Pontos de Vista de Arquitetura**: Produto de trabalho que estabelece as convenções para a construção, interpretação e uso de Visões de Arquitetura para estruturar preocupações específicas do sistema.
- **Stakeholders**: Indivíduos, equipe, organização ou classes disso, tendo interesse num sistema.
- **Preocupações**: Interesse em um sistema relevante para um ou mais *Stakeholders*. As preocupações surgem ao longo do ciclo de vida das necessidades e requisitos do sistema, das

escolhas de *design* e das considerações de implementação e operação. As preocupações podem influenciar em diferentes aspectos de um sistema.

- **Decisões Arquiteturais:** As decisões dizem respeito a preocupações do sistema. No entanto, muitas vezes não existe um mapeamento simples entre os dois. Uma decisão pode afetar a arquitetura de várias maneiras.

A arquitetura de um sistema é composta de informações essenciais desse sistema em relação ao seu ambiente [34]. Não há uma definição explícita do que é essencial ou fundamental para um sistema, mas essa definição pode: pertencer aos elementos ou sistemas constituintes; demonstrar como os elementos do sistema estão dispostos ou relacionados; estar relacionada aos princípios da organização ou *design* do sistema; pode estar relacionada aos princípios que regem a evolução do sistema ao longo do seu ciclo de vida [34]. Uma arquitetura é abstrata, consiste de conceitos e propriedades, enquanto que Descrições de Arquitetura são trabalhos que são usados para expressar arquiteturas para Sistemas de Interesses.

Durante anos, pesquisadores e organizações desenvolveram técnicas, processos, métodos, guias, boas práticas, entre outros documentos para a construção de *design* de arquitetura de software. Cada um desses documentos desenvolvidos possui características próprias, porém, conforme Hofmeister [29], essas diversas documentações definem métodos para *design* de arquitetura de software com o mesmo objetivo: manter o controle intelectual sobre o *design* de sistemas de software. Esses sistemas exigem envolvimento e negociação entre múltiplos *stakeholders*. Muitas vezes são sistemas desenvolvidos por grandes equipes distribuídas durante longos períodos de tempo. Outras vezes abordam múltiplos objetivos e preocupações possivelmente conflitantes. Além de terem a necessidade de serem mantidos por um longo período de tempo.

Assim, ao analisar os métodos para *design* de arquitetura de sistemas de software, pode-se considerar o trabalho de Hofmeister [29], no qual comparou cinco (5) métodos da indústria, destacando as similaridades e diferenças entre os métodos. Mas mais relevante que isso para este trabalho é a definição de um modelo geral para *design* de arquitetura de software. Este modelo geral começa com a classificação dos tipos de atividades executadas durante o *design* de arquitetura. A análise arquitetural articula os requisitos arquiteturalmente significantes (ASR do inglês *Architecturally Significant Requirements*) com base nas preocupações e contextos arquiteturais. A síntese arquitetural resulta em soluções arquiteturais candidatas que atendem a esses requisitos. A avaliação arquitetural garante que as decisões arquitetônicas utilizadas sejam as corretas. Na Figura 3.1, tem-se as atividades e os artefatos propostos.

Neste modelo há três atividades principais que são executadas, não necessariamente sequencialmente. Mas, em vez disso, evoluem passo-a-passo enquanto que os arquitetos se movem constantemente de uma atividade para outra, fazendo com que a arquitetura evolua progressivamente ao longo do tempo [29]:

- **Análise Arquitetural:** Analisa as preocupações de arquitetura para estabelecer, em termos de ASRs, quais problemas do sistema que a arquitetura de software pode resolver.

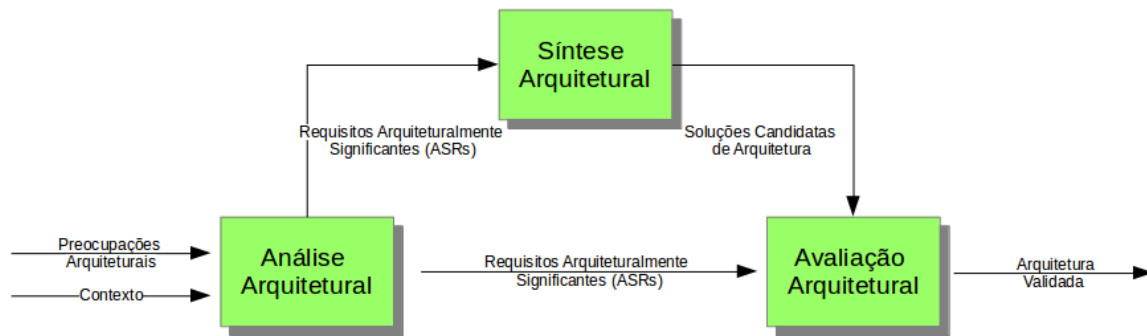


Figura 3.1 – Atividades e Artefatos de um Modelo Geral de *Design* Arquitetural. Fonte: [29] p.113

- **Síntese Arquitetural:** Propõe soluções de arquitetura para resolver um conjunto de ASRs, assim passa do problema para a solução. É a atividade principal do *design* da arquitetura.
- **Avaliação Arquitetural:** Avalia as soluções de arquiteturas candidatas contra os ASRs. Embora sejam esperadas várias iterações, o resultado final da avaliação arquitetural é uma arquitetura validada. Assim, o objetivo é verificar se as decisões de *design* da arquitetura são as corretas.

Relacionadas as atividades, estão os artefatos envolvidos que compõe o fluxo das informações conforme listado abaixo [29]:

- **Preocupações Arquiteturais:** Essas preocupações são definidas como interesses no desenvolvimento da arquitetura do sistema. Esses interesses estão relacionados ao desenvolvimento, operação e qualquer outro aspecto relevante a um ou mais *stakeholder*. A maioria das preocupações arquiteturais são expressas em requisitos do sistema e em decisões de *design* obrigatórias.
- **Contexto:** Conjunto de elementos e circunstâncias que influenciam de alguma forma o sistema. A diferença entre Contexto e Preocupações Arquiteturais é que se se for especificamente relacionada ao sistema é Preocupação Arquitetural, se for uma característica geral ou objetivo da organização ou do *stakeholder*, então é Contexto. Por exemplo: o objetivo de negócio da arquitetura é uma preocupação, enquanto um objetivo de negócio da empresa é o contexto.
- **Requisitos Arquiteturalmente Significantes (ASRs):** São requisitos de software que influenciam na arquitetura. Nem todos os requisitos do sistema influenciam a arquitetura.
- **Soluções Candidatas de Arquitetura:** podem representar soluções parciais ou alternativas. Expressam opções de decisões de *design* alternativas sobre a estrutura do software;
- **Validação da Arquitetura:** Soluções Candidatas de Arquitetura que foram avaliadas e escolhidas que estão consistentes com os ASRs. Além disso, essas soluções devem ser mutuamente consistentes.

3.2 Arquitetura de Software em SoS

Arquiteturas de sistemas de software são essenciais para promover o sucesso e a qualidade dos sistemas, ainda mais em um SoS [26]. A Arquitetura de Software em SoS segue os mesmos princípios da Arquitetura de Software tradicional, porém com mais restrições e desafios, uma vez que SoS são complexos e incluem características que não são típicas dos projetos de sistemas tradicionais [11]. A arquitetura de SoS define como os sistemas funcionam em conjunto para atender aos objetivos SoS e considera os detalhes dos sistemas constituintes e seu impacto no desempenho ou funcionalidade do SoS [26]. A arquitetura de um SoS endereça os conceitos e operação para o SoS e engloba as funções, relações e dependências dos sistemas constituintes, internos e externos. Isso inclui funcionalidades *end-to-end* (de uma ponta a outra do sistema) e fluxo de dados, bem como as comunicações [27].

Como exemplo dessas dificuldades, Butterfield, Pearlman e Vickroy [11], os sistemas que se tornam sistemas constituintes do SoS normalmente são projetados sem o conhecimento de outros sistemas constituintes. Sistemas legado do SoS foram projetados usando as expectativas e limitações que fornecem uma capacidade para satisfazer uma necessidade ou requisito, mas sem conhecer os recursos e limitações de outros sistemas constituintes do SoS. Por necessidades como estas, que o DoD [27] afirma que a arquitetura de um SoS é um *framework* persistente para governar a evolução do SoS ao longo do tempo. Um exemplo de um *framework* de arquitetura é a Arquitetura Orientada a Serviços, do inglês *Service Oriented Architectures* (SOA). SOAs podem criar SoS através do mapeamento de serviços de sistemas, orquestrando a comunicação entre os serviços dos sistemas constituintes [19].

Assim, Gonçalves, Cavalcante, Batista, Oquendo e Nakagawa [25] propõem o modelo conceitual que tem por objetivo apresentar as características de um *Software-intensive* SoS (SiSoS), e também fazer a distinção de SiSoS com os demais tipos de sistemas. Conforme o modelo [25], um SiSoS é composto de sistemas constituintes, que contribuem para a realização da missão global do SiSoS e podem ter independência gerencial e um objetivo individual. Além disso, os sistemas constituintes têm independência operacional e apresentam a capacidade de se conectar uns aos outros, permitindo assim sua cooperação para produzir funcionalidades emergentes. Assim, tanto um SiSoS quanto seus sistemas constituintes podem ter seus respectivos *stakeholders* (*stakeholders* de SoS e *stakeholders* dos sistemas constituintes). Este modelo conceitual de arquitetura pode ser analisado na Figura 3.2.

Outro trabalho de Gonçalves, Oquendo e Nakagawa [26] colaborando com o modelo SiSoS, onde a principal contribuição deste trabalho é disponibilizar um meta-processo para construção de arquiteturas de software em SoS *Acknowledged* chamado de SOAR. Neste contexto, as mudanças no SoS dependem da negociação entre os *stakeholders* e os desenvolvedores de SoS e sistemas constituintes. O SOAR combina as experiências de Engenharia de Sistemas em arquitetura de SoS com o conhecimento de Engenharia de Software no desenvolvimento de arquiteturas de software. Assim sendo, arquitetos de software, gerentes de processos e outros *stakeholders* do SoS podem

7. **Apoiando múltiplos provedores de sistemas constituintes:** devido aos sistemas constituintes poderem ter diferentes interesses e estarem em diferentes fases de desenvolvimento, cria a necessidade de trabalhar com diferentes times de Arquiteturas.
8. **Apoiando desenvolvimento de arquitetura distribuído:** esta atividade deve englobar o planejamento/suporte necessários para que ocorra o trabalho colaborativo entre equipes heterogêneas.
9. **Propondo uma nova versão de Arquitetura:** nesta atividade soluções arquiteturais são propostas com base nos ASCs do SoS e capacidades dos sistemas constituintes na ordem de encontrar o conjunto de ASRs do SoS. O principal resultado desta atividade é estabelecer uma nova versão de arquitetura para futura avaliação.
10. **Descrevendo uma nova versão de Arquitetura:** nesta atividade a arquitetura do SoS é descrita de acordo com o contexto de desenvolvimento de cada SoS.
11. **Prevendo Comportamento Emergente:** comportamento emergente de um SoS não são simplesmente soma de partes do sistemas constituintes, eles devem ser previstos também considerando a identificação de comportamentos desejáveis e indesejáveis.
12. **Avaliando e Validando versões de Arquitetura:** nesta atividade verifica-se se as decisões de *design* arquitetural estão corretas. A Arquitetura do SoS é também verificada contra os ASRs, ASCs e qualquer outro elemento relevante ao contexto do SoS, através de múltiplas iterações.

Percebe-se a importância do *stakeholder* no processo de arquitetura de um sistema. Através de suas preocupações, eles vão ajudar a definir os requisitos relevantes para a arquitetura que por sua vez determinarão os atributos de qualidade de um sistema. Além do *stakeholder*, os requisitos relevantes para a arquitetura, neste trabalho representados por ASRs, bem como as preocupações dos *stakeholder*, ASCs, são os principais insumos para a elaboração de uma arquitetura consistente e objetiva. A identificação dos principais *stakeholders* é essencial para alcançar o melhor desenho de arquitetura, tanto que Bass [5] aconselha a todos os arquitetos que conheçam seus *stakeholders*, que falem com eles, envolva-os e ouça-os.

4. ENGENHARIA DE REQUISITOS E *STAKEHOLDERS*

Este capítulo apresenta as definições dos conceitos de Engenharia de Requisitos e *stakeholders* que foram utilizados neste trabalho. Na Seção 4.1, apresenta-se as definições de Engenharia de Requisitos; na Seção 4.2, apresenta-se o comportamento da Engenharia de Requisitos em SoS; na Seção 4.3, apresenta-se abordagens de desenvolvimento de sistemas de software; na Seção 4.4, apresenta-se a intersecção de Engenharia de Requisitos e Arquitetura de Software em SoS; na Seção 4.5, apresenta-se as definições de *stakeholders*; na Seção 4.6, a relação entre os *stakeholders* e Arquitetura de Software dos Sistemas; e na Seção 4.7, apresenta-se os trabalhos relacionados com Processos de Identificação de *stakeholders*.

4.1 Engenharia de Requisitos

A ER é o processo pelo qual os requisitos são determinados [14]. O sucesso desse processo envolve entender as necessidades do usuário, cliente e outros *stakeholders*; o entendimento do contexto no qual o software desenvolvido será utilizado; modelagem, análise, negociação e documentação dos requisitos dos *stakeholders*; validação de que os requisitos documentados combinam com os requisitos negociados; e evolução do gerenciamento de requisitos [14].

Existem diversos tipos de tarefas na ER como a elicitación, modelagem, análise de requisitos, validação e verificação, gerenciamento de requisitos, entre outras. Focando-se na elicitación de requisitos que compreende, conforme Cheng e Atlee [14], atividades que permitem o entendimento das metas, objetivos e motivações para a construção de um sistema de software. A elicitación dos requisitos também envolve a identificação dos requisitos que o sistema resultante deve satisfazer a fim de atingir esses objetivos. De acordo com Cheng e Atlee [14], grande parte das tarefas de elicitación de requisitos concentra-se em técnicas para melhorar a precisão, acurácia e variedade dos detalhes dos requisitos como as técnicas a seguir:

- **Técnicas para identificar *stakeholders*:** ajudam a garantir que todos os que possam ser afetados pelo software sejam consultados durante a elicitación.
- **Técnicas analógicas, como metáforas e personas:** ajudam os *stakeholders* a serem mais precisos, irem mais a fundo nos seus requisitos.
- **Técnicas de RE contextual e pessoal:** analisam os requisitos dos *stakeholders* em relação a um determinado contexto, ambiente e, talvez, usuário individual, para ajudar a garantir que o sistema final está adequado para uso nesse ambiente.
- **Técnicas para sugerir requisitos (atividades de *brain-storming*):** ajudam a identificar requisitos não essenciais que tornam o produto final mais atraente.

- **Técnicas de *feedback***: o uso de modelos, simulações, animações para elicitare *feedbacks* positivos e negativos sobre as representações iniciais do sistema proposto.

Requisitos surgem de ideias ou conceitos que podem originar-se por uma ameaça a segurança, de uma imposição de lei, do desejo de criar algo novo, da necessidade de melhorar o processo ou sistemas ou de alguma outra necessidade percebida além dessas mencionadas. São através dessas ideias ou conceitos que podem originar o conjunto de requisitos que podem ser obtidos por diferentes técnicas. A IEEE em seu Guia para o desenvolvimento de especificações de requisitos de sistema define as técnicas de identificação de requisitos conforme a seguir [31]:

- (a) **Workshops estruturados**: reuniões estruturadas de um grupo de pessoas interessadas em um determinado assunto.
- (b) **Brainstorming ou sessões para resolução de problemas**: reuniões de um grupo de pessoas interessadas em explorar a potencialidade criativa de um indivíduo ou de um grupo para resolução de problemas.
- (c) **Entrevistas**: conversas entre pessoas interessadas para obter informações necessárias.
- (d) **Surveys ou questionários**: questionários estruturados com o objetivo de coletar informações.
- (e) **Observação de padrões de trabalho**.
- (f) **Observação de sistemas ambiente socio-organizacional**.
- (g) **Revisão de documentação técnica**.
- (h) **Análise de mercado**: análise do que seus clientes querem e do que seus concorrentes oferecem.
- (i) **Avaliação de sistemas concorrentes**.
- (j) **Engenharia Reversa**: processo de descobrir os princípios e o funcionamento de um sistema, através da análise de sua estrutura, função e operação. Desconstruir o que existe para entender como funciona.
- (k) **Simulações**: processo de simular a interação do usuário no sistema.
- (l) **Prototipações**: visão inicial de parte do sistema que possibilita demonstrar conceitos e experimentar opções de implementação.
- (m) **Processos e sistemas de Benchmarking**: comparação de processos e de sistemas.

4.2 Engenharia de Requisitos em SoS

Os requisitos de SoS são de propriedade de diversos *stakeholders* e existem em diferentes níveis, para diferentes sistemas e em diferentes artefatos. Eles geralmente se sobrepõem ou entram em conflito, e atribuí-los aos sistemas constituintes específicos nem sempre é possível [66]. Por esses motivos, aplicar as técnicas da ER tradicional em SoS torna-se um desafio, por serem potencializadas pelas características de um SoS como distribuição geográfica, comportamento emergente, desenvolvimento evolucionário, e independência operacional e gerencial dos constituintes. As técnicas de ER tradicional são definidas para desenvolvimento de novos sistemas, seguindo uma ordem natural para construção de um sistema, iniciando pela definição dos requisitos dos *stakeholders*, passando pela execução da análise de requisitos e pela elaboração do desenho de arquitetura que será implementado. Ncube [48], por exemplo, afirma que as técnicas de requisitos atuais são inapropriadas para abordar completamente as características do SoS, visando principalmente comportamento emergente e desenvolvimento evolutivo, bem como as várias formas de colaboração entre os sistemas constituintes. Isto se dá visto que a complexidade de SoS representa uma nova realidade que exige da comunidade de engenharia de requisitos mudanças na forma de pensar sobre processos de engenharia de requisitos atuais [48].

Os principais motivos da afirmação de Ncube são a natureza e as características dos requisitos para SoS, que abrangem muitas disciplinas, são fragmentadas, às vezes conflitantes, instáveis, difíceis de reconhecer e podem não ser totalmente definidas [48]. Além disso, estes requisitos são definidos como requisitos de SoS e são implementados através das capacidades dos sistemas constituintes ou dos requisitos em nível do sistema constituintes que são alocados a um ou vários sistemas constituintes específicos [48] [27]. Esses tipos de requisitos mudam constantemente e isso torna a divisão dos requisitos muito complexa.

Keating [37] concorda com Ncube quando ressalta que a SoSE trata de um domínio de problema que está destinado a modificar a forma tradicional de pensar sobre engenharia de sistemas complexos, assim como destinado a mudar a forma de pensar sobre os requisitos. Por este motivo, o sucesso obtido pela ER na ES tradicional pode não ter o mesmo sucesso na SoSE devido as divergências que aparecem em todas as áreas elencadas. Ncube vai além e afirma que é necessário que novos processos de ER, métodos de gestão, técnicas e ferramentas que possam responder dinamicamente a requisitos instáveis, fragmentados e em constante mudança [48].

Para minimizar estas dificuldades, Barden e Evans [4] sugerem a criação de um grupo de pessoas capacitadas para trabalhar com requisitos em SoS, a este grupo eles chamam de Grupo de Avaliação de Requisitos (RSG), do inglês *Requirements Screening Group*. Este grupo é formado pelos representantes dos *stakeholders*: Contratante, Cliente e Usuários. Estes representantes são autorizados e responsáveis por responder as decisões tomadas além de fornecer orientação as atividades de desenvolvimento. Das atividades previstas como responsabilidade do RSG as principais estão listadas a seguir [4]:

- Interpretação da intenção do usuário como um guia para o desenvolvimento de requisitos;
- Desenvolvimento e documentação dos requisitos dos *stakeholders*;
- Desenvolvimento e documentação dos requisitos técnicos (performance do SoS);
- Rastreabilidade de requisitos entre o documento de requisitos dos *stakeholders* e do SoS e especificações dos sistemas constituintes;
- Rastreabilidade entre requisitos SoS e requisitos de sistemas constituintes;
- Solicitações de mudanças nos requisitos dos *stakeholders*;
- Solicitações de mudanças dos requisitos técnicos (performance do SoS);
- Revisão e aprovação dos requisitos; e
- Garantir que os requisitos de um grupo não contradizem ou impõe um problema para outro.

Um SoS surge quando os sistemas envolvidos interagem com hardware e com os demais sistemas constituintes. Este comportamento exige monitoramento do SoS, para certificar se os requisitos estão sendo atendidos. Assim, além dos requisitos que surgem em tempo de projeto, também se precisa estar atendo aos requisitos que surgem em tempo de execução. Dessa forma, pode-se dizer que existem diversos tipos de requisitos num SoS, os mais triviais são os requisitos que surgem em tempo de projeto nos sistemas constituintes, estes podem ser detectados por mecanismos da ER tradicional, são requisitos que tem origem na concepção do SoS. Com maior complexidade, têm-se os requisitos emergentes do SoS, que são novos requisitos que emergem dos sistemas constituintes, uma vez que um novo requisito é necessário a um sistema constituinte ele torna-se requisito do SoS. Também com alto grau de complexidade, tem-se os requisitos que surgem em tempo de execução. Estes requisitos são percebidos através de monitoramento do SoS e podem surgir em tempo de execução do SoS [65].

Para identificar os diversos tipos de requisitos de um SoS, precisa-se conhecer as abordagens de desenvolvimento existentes e como ocorre o ciclo de vida de uma solicitação dentro da abordagem. A Seção 4.3 apresenta os detalhes que vincula o tipo de requisito com as abordagens de desenvolvimento.

4.3 Abordagens de Desenvolvimento

Devido as características do SoS, o ciclo de vida de uma solicitação de implementação é uma atividade mais complexa do que num sistema tradicional. Em um sistema tradicional, a implementação de uma solicitação, segue uma série de processos. Esses processos foram definidos pelas ISO 12.207 e 15.288 [33] [35] como:

1. **Definição de Requisitos de Stakeholders:** processo que define os requisitos dos *stakeholders* para um sistema que pode fornecer os recursos necessários aos usuários e outros *stakeholders* de um ambiente específico.
2. **Análise de Requisitos do Sistema:** processo para transformar a visão das capacidades desejadas pelo usuário em uma visão técnica da solução que atende às necessidades operacionais do usuário.
3. **Design Arquitetural do Sistema:** processo para gerar alternativas arquiteturais, para selecionar uma ou mais alternativas que enquadram as preocupações dos *stakeholders* e atendem aos requisitos do sistema, sendo expressadas em um conjunto de visualizações consistentes.
4. **Processo de Definição de Design:** processo que provém dados e informações detalhadas sobre o sistema e seus elementos, permitindo a implementação consistente com as entidades arquitetônicas conforme definições em modelos e visualizações da arquitetura do sistema.
5. **Processo de Análise do Sistema:** processo que provém uma rigorosa base de dados e informações para entendimento técnico, ajudando na tomada de decisão ao longo do ciclo de vida do sistema.
6. **Processo de Implementação:** processo que realiza a implementação de um elemento específico do sistema.
7. **Processo de Integração do Sistema:** processo que sintetiza um conjunto de elementos no sistema que satisfaz os requisitos, arquitetura e *design* do sistema.
8. **Processo de Teste de Qualidade do Sistema:** processo que garante que a implementação de cada requisito do sistema é testada para avaliar se está em conformidade. Assim, definindo que o sistema está pronto para entrega.
9. **Processo de Instalação do Software:** processo que instala o produto de software, atendendo aos requisitos acordados no ambiente de destino.
10. **Processo de Suporte de Aceite de Software:** processo que ajuda o adquirente (comprador) a confiar em que o produto atende aos requisitos.
11. **Processo de Operação de Software:** processo usado pelo sistema para entregar seus serviços.
12. **Processo de Manutenção de Software:** processo usado para sustentar a capacidade do sistema no fornecimento de seus serviços.
13. **Processo de Eliminação de Software:** processo para acabar com a existência de uma entidade do sistema de software.

Cada sistema constituinte possui estas etapas para construção de uma nova implementação. Enquanto isso, a comunidade da ES aborda muitos problemas e dificuldades causadas pelo conceito emergente dos SoS [13]. A combinação dos ciclos de vida dos sistemas constituintes em um ciclo de vida de SoS apresenta uma série de desafios em ES [13]. É difícil para a comunidade ES desenvolver soluções sistemáticas para abordar questões de engenharia envolvidas na evolução SoS devido a esta dificuldade de combinar os ciclos de vida dos sistemas constituintes com o ciclo de vida do SoS [13]. O gerenciamento da evolução de um SoS requer adaptação de práticas da ES para enfrentar desafios típicos do ambiente de um SoS [13].

Esta necessidade ocorre pois SoS envolvem muitos sistemas que estão em diferentes fases do ciclo de vida [27], enquanto determinados sistemas em fase de implementação, outros estão em fase de manutenção, outros em operação, entre outras possíveis fases. SoS podem compreender sistemas legados, sistemas em desenvolvimento, sistemas relacionados a outras iniciativas [27]. Estes diferentes momentos dos sistemas constituintes se torna uma tarefa complicada para a equipe de SoS que deve navegar nos planos de evolução e prioridades de desenvolvimento dos sistemas constituintes, durante sua agenda de desenvolvimento assíncrona, para planejar e orquestrar a evolução do SoS para alcançar os objetivos do SoS. Além desses desafios encontrados em desenvolvimento, dependendo da complexidade e distribuição dos sistemas constituintes, pode ser impossível ou muito difícil testar e avaliar completamente os recursos SoS [27]. Uma vez que o desenvolvimento é descentralizado e nem sempre haverá um ambiente de testes atualizado de cada sistemas constituinte, comunicando com o SoS e disponível para ocorrer o teste do SoS. Por exemplo, uma solicitação do SoS pode gerar alterações em diversos sistemas constituintes. Assim, nesse exemplo, a implementação de uma nova funcionalidade em um sistema constituinte (A) pode fazer com que seja necessário que outro sistema constituinte (B) se adapte para utilizar esta nova implementada e conseqüentemente indiquem que sejam realizadas alterações tanto em A como em B.

Em tempo de desenvolvimento de sistemas, existem diferentes tipos de abordagens que podem ser utilizadas. Essas abordagens possuem estruturas distintas e podem ser aplicadas a processos específicos do ciclo de desenvolvimento [61]. Das abordagens existentes, as mais relevantes são: *Top-down*, *Middle-out* e *Bottom-up*. Essas abordagens são definidas abaixo:

Top-down: focada no entendimento do problema, é a abordagem mais conhecida [61]. Muito bem aplicada em sistemas tradicionais, no desenvolvimento de um novo sistema. Parte do problema em busca de solução.

Bottom-up: abordagem normalmente utilizada para reengenharia ou engenharia reversa de um sistema [61]. Essa abordagem é ideal para projetos que estão atualizando ou substituindo sistemas. Como os sistemas legados muitas vezes não possuem documentação adequada, a engenharia reversa fornecerá uma análise para deduzir os recursos de *design* de produtos com pouco conhecimento sobre o processo original de projeto de engenharia. A abordagem começa com o foco em componentes de nível inferior e como eles são integrados e eventualmente atravessa os requisitos de sistema originais.

Middle-out: abordagem implementada geralmente em projetos que requerem melhoria ou evolução de sistemas [49]. Esta abordagem geralmente aborda as restrições que ocorrem durante o processo de projeto ao atualizar os sistemas existentes. Os mesmos processos que se aplicam a uma abordagem *top-down* serão usados em uma abordagem *middle-out*, mas a diferença está onde a arquitetura começa. A análise operacional é usada para criar os requisitos do sistema (em alto nível), em seguida, decomposta para identificar o sistema e seus componentes.

Uma tarefa complicada para a equipe de SoS é a de navegar pelos planos de evolução e de prioridades dos sistemas constituintes. Estes sistemas podem possuir cronogramas de desenvolvimento que são assíncronos entre eles e planejar e orquestrar a evolução dos sistemas constituintes para os objetivos do SoS torna-se uma tarefa difícil de ser executada. Além desses desafios de desenvolvimento, dependendo da complexidade e distribuição dos sistemas constituintes, pode ser impossível ou muito difícil testar e avaliar completamente os recursos SoS [27]. O planejamento e implementação da SoSE deve considerar e influenciar os planos de desenvolvimento dos sistemas constituintes [27]. Para entender o ambiente no qual um sistema ou SoS serão desenvolvidos e empregados é fundamental para entender como melhor aplicar os princípios da ES dentro desse ambiente.

Para entender melhor o ciclo de vida de um SoS, pode-se analisar o exemplo do GEOSS exemplificado na Seção 2.3 deste trabalho. O GEOSS interconecta esses sistemas para fortalecer o monitoramento do Planeta Terra. Um sistema deste porte pode ser considerado de Grande Escala (ULS, do inglês *Ultra-Large-Scale*) conforme Overton [49] e tem desafios de *Design* como competição de requisitos, continua evolução, sistemas heterogêneos e falhas. O GEOSS tem a capacidade de ser muito maior em tamanho do que muitos sistemas em número de linhas, número de usuários, quantidade de dados armazenados, acessados e processados, número de interdependência de componentes de software, entre outros.

No caso do GEOSS, Overton [49] propõe como uma boa prática, a inclusão de mais uma camada ao modelo RUP 4+1 definida como visão do problema. A visão do problema preocupa-se com a redução de problemas complexos em problemas mais simples ou em problemas que já foram resolvidos anteriormente. Utilizando a abordagem de desenvolvimento *middle-out*. Essa abordagem começa com um padrão de *design* apropriado, descrevendo o problema de acordo com esse padrão de *design* e elaboram a solução de acordo com os requisitos deste padrão. Dessa forma, Overton [49] sugere que o uso de padrões pode balancear as forças opostas por definir a estrutura do problema e o desenvolvimento permitir um projeto que equilibre a competição dos requisitos.

Dessa forma, a abordagem *middle-out* é utilizada geralmente em projetos que requerem melhoramento ou evolução de sistemas, uma vez que se baseiam em *Design Patterns*. Utiliza os mesmos processos que se aplicam a uma abordagem *top-down*, porém a grande diferença está na fase de arquitetura de sistema que ocorre no início do processo. Nesta abordagem, a análise operacional é usada para criar os requisitos do sistema (em alto nível), em seguida, decomposta para identificar o sistema e seus componentes [49]. Com essa abordagem pode-se equilibrar a competição pelos requisitos, uma vez que eles são gerados por análise operacional [49]. Na Figura 4.1, Overton [49]

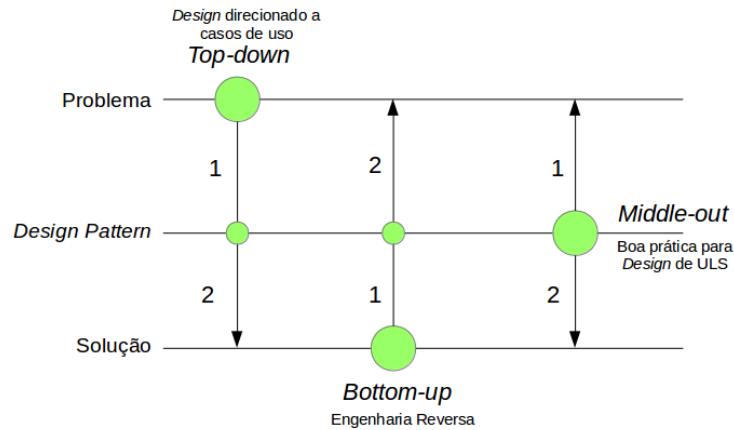


Figura 4.1 – Comparação entre as abordagens de *design*. Fonte: [49] p.4

realiza a comparação entre as diferentes abordagens. A abordagem *middle-out* é implementada no GEOSS como uma boa prática.

Assim, tanto a abordagem de desenvolvimento utilizada pelo SoS *top-down* ou *middle-out* quanto o momento do ciclo de vida de cada sistema constituinte são fatores que precisam fortemente considerados para a construção e evolução de um SoS. E todos os *stakeholders* importantes nestes processos precisam ser considerados no SoS.

4.4 Engenharia de Requisitos e Arquitetura de Software em SoS

Ao longo dos anos vários argumentos foram utilizados para diferenciar ER de Arquitetura de Software. O argumento mais conhecido que os diferencia é: requisitos são definidos como problemas e a arquitetura de sistema como estrutura de um sistema de software que resolve problemas com componentes e conectores como os principais elementos. Apesar dessa distinção, Boer [18] afirma que numa visão emergente de arquitetura há uma similaridade que aproxima estas duas áreas, são os ASR, ou, em outras palavras, requisitos que influenciam profundamente na arquitetura do sistema. Esta visão emergente de arquitetura faz com que a arquitetura não apenas defina a estrutura da solução, mas além disso defina um conjunto de decisões de *design* que conduzem à estrutura da solução.

Nesta visão emergente de arquitetura, técnicas de ER são utilizadas para identificação de ASRs. Como são os *stakeholders* que definem os ASRs, começa-se identificando os ASCs. A listagem de ASCs é uma das entradas para a definição dos ASRs. Estas preocupações dos *stakeholders* estão relacionadas ao sistema de interesse no seu contexto. Conforme a ISO 42.010 [34], essas preocupações podem ser de um *stakeholder* ou de um grupo de *stakeholders*, e surgem ao longo do ciclo de vida de um sistema, na fase de elicitação dos requisitos, nas escolhas de *design* e nas considerações de implementação e operação do sistema. Ainda de acordo com a ISO 42.010 [34], preocupações podem ser manifestadas de muitas formas, como na relação a uma ou mais necessidades,

metas, expectativas, responsabilidades, requisitos, restrições de projeto, suposições, dependências, atributos de qualidade, decisões de arquitetura, riscos ou outras questões relacionadas ao sistema. Alguns exemplos de preocupações que *stakeholders* tem podem ser com as funcionalidades, uso, finalidade do sistema, segurança, desempenho, limitações conhecidas, comportamento do sistema, comunicação entre processos e sistemas, experiência dos clientes, entre outros.

De acordo com Bass [5], nem todos os requisitos tem o mesmo impacto na arquitetura, em muitos casos apenas alguns requisitos tem impacto profundo na arquitetura. ASRs frequentemente são conhecidos como requisitos de atributos de qualidade, performance, segurança, disponibilidade, usabilidade e assim por diante [5]. Arquitetos identificam ASR através de ASRs candidatos. Bass [5] afirma que a primeira percepção que teve quando observou arquitetos experientes trabalhando na identificação de ASR foi que eles procuraram conversar com os *stakeholders* importantes.

Os requisitos candidatos para tornar-se um ASR são originados de documentos de requisitos ou *user stories*, dependendo da técnica de desenvolvimento que é adotada. Porém muitos projetos não criam ou não mantêm as documentações atualizadas [5]. ASRs frequentemente derivam de objetivos de negócios no próprio desenvolvimento. Sendo assim, muitos *stakeholders* não possuem ideia do que é um ASR ou do que é um atributo de qualidade. Assim, Bass [5] ressalta que entrevistar os *stakeholders* importantes é o caminho mais seguro para saber o que eles sabem e precisam. Em síntese, os ASRs podem ser extraídos de documentos de requisitos, capturados junto aos *stakeholders* ou derivados de objetivos de negócio.

Devido a dificuldade dos arquitetos em encontrar os ASRs, Chen, Babar e Nuseibeh [12] desenvolveram um *framework* para caracterizar os ASRs. O *framework* está dividido em quatro conjuntos de características: Definição, Descrição, Indicação e Heurística. A Definição, é a fase na qual precisa-se definir os critérios sobre o que é "significante". Este é o termo chave da definição, ele vai demarcar as fronteiras. Descrição é onde se define o conjunto de características para encontrar os ASRs. Uma vez que os ASRs tendem a serem expressados vagamente, negligenciados inicialmente, escondidos em outros requisitos, subjetivos, variáveis e situacionais. Indicação é onde se apresenta indicadores como custo de um requisito, neste exemplo, o custo de um requisito sendo uma medida de significância. A estimativa de custos geralmente será realizada depois que os requisitos foram reunidos para medir o impacto do requisito no sistema. E por fim, Heurística é onde são listadas as características conhecidas pelos engenheiros de requisitos. As características consideradas pelos autores foram as seguintes: atributos de qualidade, principais recursos, restrições e ambiente de aplicação.

Cada requisito percebido passa por este *framework*. Desses requisitos, o *framework* apontará qual é um ASR. Normalmente um ASR será o requisito que impõe restrições ao sistema, que define o ambiente onde o sistema vai executar, que se refere as principais características do sistema ou o requisito que define um atributo de qualidade, ou seja um requisito não funcional como por exemplo: adaptabilidade, disponibilidade, configuração, flexibilidade, interoperabilidade, desempenho, confiabilidade, responsividade, recuperabilidade, escalabilidade, estabilidade, segurança, extensibilidade, modularidade, portabilidade, reutilização, testabilidade, auditabilidade, manutenibilidade,

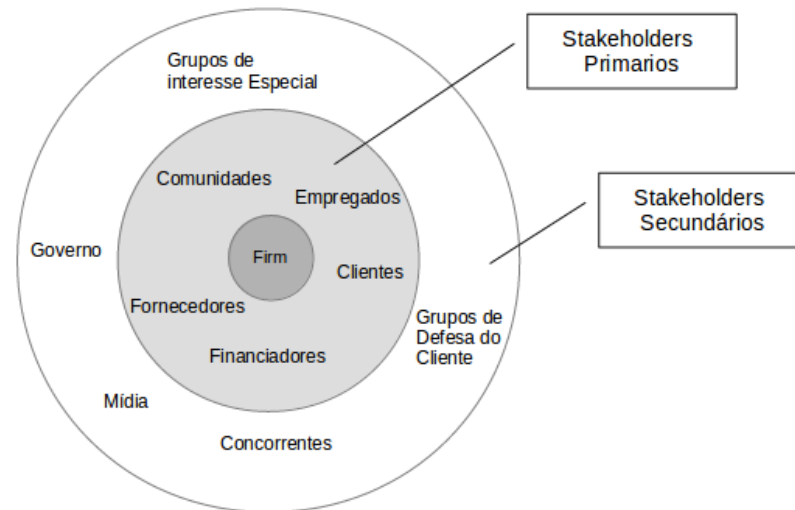


Figura 4.2 – Modelo de Freeman. Fonte: [23] p.292

gerenciamento, sustentabilidade, suporte e usabilidade [12]. Em suma, os ASRs indicam preferências para a direção na qual o sistema deve seguir e delimitam o espaço do *design* descartando alternativas indesejáveis.

Por fim, entende-se que deve ser dada ênfase especial aos processos de engenharia de requisitos para lidar com as competições entre os *stakeholders*, criar SoS interoperáveis, SoS com arquiteturas persistentes, SoS com evolução dinâmica e como os comportamentos emergentes afetam a estabilidade de requisitos [48].

4.5 Stakeholders

O termo *stakeholder* tem origem no livro “Strategic Management: A Stakeholder Approach” de Since Freeman (1984) que foi lançado como parte da série “Business and Public Policy”. Este livro foi um marco no desenvolvimento da teoria de *stakeholders*, onde o autor define *stakeholder* como algum grupo ou indivíduo que pode afetar ou ser afetado pela realização dos objetivos da empresa.

Com o decorrer dos anos passou-se a ver uma variação do termo *stakeholder* na literatura. Mitchell, Angle e Wood [45] apresentam 28 definições de *stakeholder* de diferentes autores, muitas são parecidas e algumas são utilizadas como base para outras. Para exemplificar essas diferentes definições trazidas pelos autores, pode-se falar do trabalho de Alkhafaji (1989) no qual define *stakeholder* como grupos a quem a corporação é responsável; de Brenner (1993) que afirma que *stakeholder* tem algum relacionamento legítimo e não trivial com uma organização como por exemplo: transações de câmbio, ações de impacto e responsabilidades morais; e o trabalho de Nasi (1995) no qual *stakeholders* interagem com a empresa e assim tornam possível a operação da mesma. Conforme Fassin [22], a teoria de *stakeholder* sofre de numerosas deficiências e imperfeições. Em partes

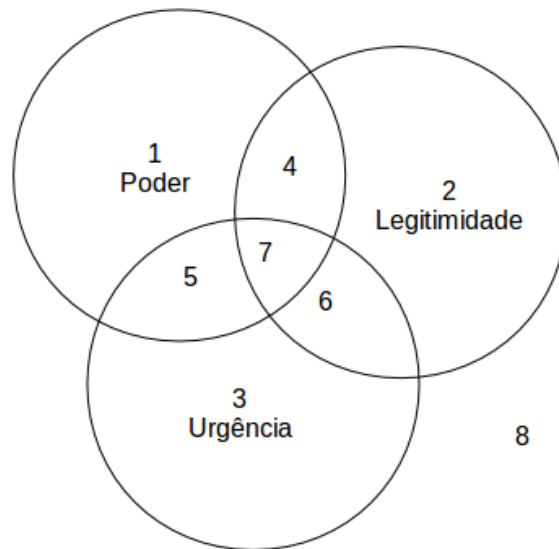


Figura 4.3 – Modelo de Mitchell, Angle e Wood. Fonte: [45] p.874

pela imprecisão, ambiguidade e amplitude da teoria em si e em parte pelo amplo apelo intuitivo, as críticas são frequentemente implícitas.

A definição mais aceita sobre *stakeholder* é a de Freeman. Conforme Freeman [23], *stakeholder* pode ser definido como qualquer grupo ou indivíduo que pode afetar ou ser afetado pela realização de objetivos da organização. Para demonstrar graficamente esta definição Freeman refinou o seu modelo de 1984 para um modelo de duas camadas, conforme a Figura 4.2. Neste modelo, a organização denominada "Firm", está no centro e a partir do centro tem-se duas camadas de modelo, onde a primeira representa o grupo de *stakeholders* primário e a segunda o grupo de *stakeholders* secundários. Nesta representação, os *stakeholders* primários tem um maior grau de importância e urgência na organização.

Mitchell, Angle e Wood [45], por sua vez, começam com uma definição ampla de *stakeholder*, respondendo as perguntas "Quem" e "O que" importa para uma organização como sendo a chave para entender os *stakeholders*, propondo classes para realizar a identificação: o poder de influência dos *stakeholders* da empresa, a legitimidade da relação dos *stakeholders* com a empresa, e a urgência dos pedidos dos *stakeholders* na empresa. Assim essas três palavras: poder, legitimidade e urgência identificam as classes de *stakeholders*. A Figura 4.3 ilustra o modelo proposto por Mitchell, Angle e Wood [45]. Cada número da imagem representa um grupo que pode ser detalhado:

1. **dormant**: *Stakeholders* que tem Poder
2. **discretionary**: *Stakeholders* que tem Legitimidade
3. **demanding**: *Stakeholders* que tem Urgência
4. **dominant**: *Stakeholders* que tem Poder e Legitimidade
5. **dangerous**: *Stakeholders* que tem Poder e Urgência

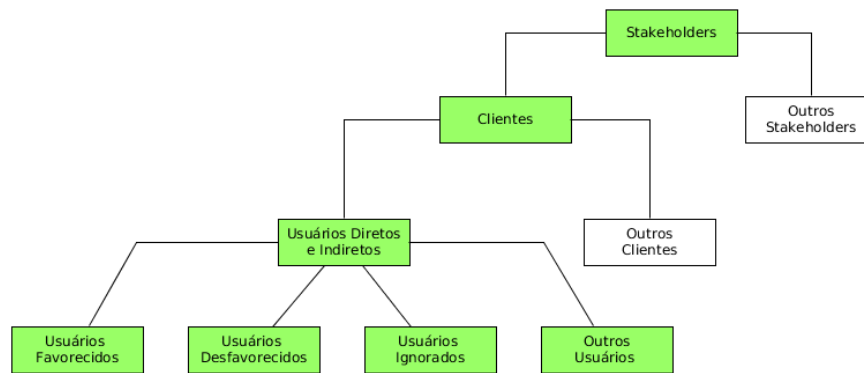


Figura 4.4 – Uma hierarquia de *stakeholders*, clientes, usuários e classes de usuários. Fonte: [67] p.103

6. **dependent**: *Stakeholders* que tem Legitimidade e Urgência
7. **stakeholders**: *Stakeholders* que tem os 3 atributos
8. **non-stakeholders**: não são *stakeholders*

Clarkson [15] define *stakeholder* como portadores de risco voluntários e involuntários, onde *stakeholders* voluntários têm alguma forma de risco, como resultado de ter investido algum tipo de capital humano ou financeiro, algo de valor, em uma empresa. Enquanto que *stakeholders* involuntários são colocados em risco como resultado das atividades de uma empresa. Mas sem o elemento de risco não há nenhuma participação.

A ISO 12.207 [33] também faz sua definição do que é um *stakeholder*. Considera que é um indivíduo ou uma organização com direitos, compartilhamentos, reclamações ou interesses em um sistema ou em sua posse de características que atendam às suas necessidades ou expectativas. Como o termo *stakeholder* é muito amplo, Wieggers e Beatty [67] definem uma hierarquia de *stakeholders* que pode ser representada na Figura 4.4. Nesta hierarquia, uma classe de usuário é um subconjunto de usuários do produto, que por sua vez é um subconjunto dos clientes do produto, que é um subconjunto dos *stakeholders*. Assim um indivíduo pode pertencer a várias classes de usuários.

Os usuários de um produto podem discordar em alguns aspectos, possibilitando agrupá-los em classes de usuários distintas com base em diferenças [67]. Essas diferenças podem ser listadas abaixo:

- privilégio de acesso ou níveis de segurança (como usuário comum, usuário convidado, administrador, etc);
- tarefas que executam durante suas atividades;
- funcionalidades que utilizam;
- frequência com que utilizam o produto;
- experiência no domínio da aplicação e experiência em sistemas computacionais;

- plataformas que estarão usando (*desktops, laptops, tablets, smartphones*, dispositivos especializados)
- idioma nativo; e
- tipo de interação com o sistema: direta ou indireta

Power [54] realça que o termo *stakeholder* é frequentemente usado no desenvolvimento de software, predominantemente limitando-se a clientes, usuários finais ou patrocinadores do projeto. As equipes de sistemas não focam num grupo de *stakeholders*, elas apenas mencionam os *stakeholders* como um cliente ou um único *stakeholder*. Está claro qual é o tipo de entidade que pode ser um *stakeholder*: pessoas, grupos, organizações, instituições, sociedades, etc. Porém não há um único modelo ou um acordo entre os autores para a definição dos *stakeholders* envolvidos. Cada autor utiliza formas distintas para definí-los.

4.6 *Stakeholders* e Arquitetura de Software

Nesta abordagem emergente de arquitetura onde se aumenta o envolvimento do arquiteto com os *stakeholders*, torna-se importante e necessário ter acesso aos *stakeholders* que impactam na arquitetura. Uma vez que apesar de todos terem interesse no sucesso do sistema, normalmente têm diferentes preocupações, por vezes contraditórias, que desejam que o sistema garanta ou otimize. De acordo com Bass [5] essas preocupações são tão diversas quanto os perfis e necessidades dos *stakeholders* e os arquitetos devem conhecê-las e entendê-las o mais cedo possível. Devido a essas preocupações, arquitetos devem identificar e envolver ativamente os *stakeholders* para solicitar suas necessidades e expectativas. E para que o engajamento precoce dos *stakeholders* permita que se compreenda as restrições, gereencie as expectativas, negocie as prioridades e faça a definição de conflitos [5]. Por esses motivos que as preocupações (ASCs) e os requisitos (ASRs) que impactam na arquitetura são importantes para a definição de uma arquitetura eficiente. Para reconhecer os ASCs e os ASRs os arquitetos precisam conhecer todos os *stakeholders* que os envolvem.

Mas para que se possa engajar os *stakeholders* e conseqüentemente se alcance o sucesso, é essencial a elaboração de uma lista dos *stakeholders*. De acordo com Rozanski e Woods [57] essa lista deve ser uma atividade colaborativa. Eles explicam que um bom *stakeholder* é aquele que informado, confirmado, autorizado e representativo. Cada uma dessas características pode ser descrita conforme a seguir:

- **Informado:** *stakeholders* que tem a informação, experiência e entendimento necessário para tomar as decisões corretas.
- **Confirmado:** *stakeholders* que estão dispostos e são capazes de se disponibilizar para participar do processo, e preparados para tomar algumas decisões.

- **Autorizado:** *stakeholders* que tem suas decisões mantidas mais tarde. Suas decisões não são revertidas por outros *stakeholders*.
- **Representativo:** quando um *stakeholders* é mais que uma pessoa, é um representante adequado de um grupo.

Normalmente *stakeholders* são representados por grupos de interesses, cada grupo representa as preocupações e objetivos do grupo com o sistema. Esses grupos, além de representar os *stakeholders*, também abstraem a quantidade de *stakeholders* envolvidos para simplificar relacionamentos entre arquiteto e representantes do grupo de *stakeholders*. Através de reuniões com os representantes dos grupos é mais fácil chegar a um consenso do que envolvendo todos. Em caso de ter um grande grupo de *stakeholders*, os arquitetos precisam gerenciá-lo ativamente, equilibrando e priorizando necessidades, para garantir que seu tamanho não impeça o progresso [57].

A definição dos perfis de *stakeholders* que estão relacionados com arquitetura pode variar em cada sistema, devido a particularidades e exceções praticadas pelo sistema em questão. Na Tabela 4.1 estão os perfis que normalmente podem ser considerados envolvidos ou interessados de alguma forma na arquitetura. A listagem destes perfis organizada pelo autor desta trabalho foi construída com base nos livros de Bass, Clements e Kazman [5], de Rozanski e Woods [57] e no padrão internacional ISO 42.010 [34]. Esta listagem foi dividida em quatro áreas de interesse: Sistemas, Negócio, Infraestrutura e Externo. Na área de Sistemas estão perfis que tem seu principal foco no ciclo de desenvolvimento de um sistema de software. Na área de Negócio estão os perfis que tem foco principal no conhecimento das atividades da empresa. Na área de Infraestrutura estão os perfis que tem foco em manter o ambiente do sistema disponível. E, por fim, a área chamada de Externo contempla os perfis que não fazem parte da organização mas tem relevância no relacionamento com as demais áreas de interesse.

Tabela 4.1 – Perfis de *stakeholders* e seus interesses

Perfil	Descrição	Interesse na Arquitetura
Sistemas		
Analista	Responsável por analisar a arquitetura para garantir que atenda a determinados ASRs críticos. Os analistas são frequentemente especializados, como por exemplo: analistas de desempenho e analistas de segurança.	Analisando a satisfação dos ASRs do sistema com base em na arquitetura.

Arquiteto	Responsável pela criação da arquitetura e sua documentação. O foco e a responsabilidade estão na estruturação do sistema.	Negociação entre os requisitos concorrentes e as abordagens de <i>design</i> . Através de um repositório para armazenar as decisões de design fornece provas de que a arquitetura atende aos seus requisitos.
Engenheiro de Sistemas	Responsável pelo <i>design</i> e desenvolvimento de sistemas ou componentes do sistema.	Lidera tecnicamente a equipe de desenvolvimento e garantem que o ambiente fornecido para o software seja suficiente.
Gerente de Projeto	Responsável pelo planejamento, sequenciação, agendamento e alocação de recursos para desenvolver e entregar os componentes do sistema para atividades de integração e teste.	Ajuda a definir o orçamento e o cronograma. Avalia o progresso em relação ao orçamento e cronograma estabelecidos, identificando e resolvendo a contenção de recursos de tempo de desenvolvimento.
<i>Designer</i>	Responsável pelo <i>design</i> de sistemas a partir da arquitetura, aplicando as decisões arquiteturais para atender aos requisitos específicos das partes para as quais são responsáveis.	Compreendendo como sua parte se comunicará e interagirá com outras partes do sistema.
Desenvolvedor	Responsável pelo desenvolvimento de elementos específicos de acordo com os requisitos de projetos e de arquitetura.	Compreender restrições invioláveis e liberdades exploráveis em atividades de desenvolvimento.
Testador	Responsável pelo teste (independente) e verificação do sistema ou de seus elementos contra os requisitos formais e de arquitetura.	Criando testes com base no comportamento e na interação dos elementos do software.
Integrador	Responsável por assumir componentes individuais e integrá-los, de acordo com a arquitetura e os projetos de sistemas.	Produzindo planos e procedimentos de integração e localizando a origem das falhas de integração.
Verificador de Conformidade	Aponta para compatibilidade com padrões e processos para proporcionar confiança na adequação de um produto.	Arquitetura é a base para a verificação de conformidade, pois garante que as implementações tenham sido fieis às prescrições arquitetônicas.
Negócio		

Gerente de Produtos	Responsável pelo desenvolvimento de toda uma família de produtos, todos construídos usando os mesmos recursos disponíveis.	Determinando se um novo membro potencial de uma família de produtos está dentro ou fora do escopo da arquitetura disponível.
Gerente de Negócios	Responsável pelo funcionamento do negócio/organização proprietária do sistema. Inclui responsabilidade gerencial/executiva, responsabilidade pela definição de processos de negócios, etc.	Compreender a capacidade da arquitetura para atingir os objetivos comerciais.
Cliente	Paga pelo sistema e assegura a sua entrega. O cliente geralmente é ou representa o usuário final. Especialmente em um contexto de aquisição do governo.	Garantir a funcionalidade e a qualidade necessárias serão entregues. Realiza avaliação do progresso, estimativa de custo e estabelece expectativas para o que será entregue, quando e por quanto.
Adquirentes ou Compras	Responsável por supervisionar a aquisição do sistema ou produto.	Podem usar a documentação da arquitetura, para verificar se a funcionalidade desejada está sendo entregue.
Usuário	São os usuários finais do sistema. Pode haver tipos distintos de usuários. Tais como administradores. Superusuários, etc.	Os usuários também podem usar a documentação para entender quais são os principais elementos do sistema, o que pode ajudá-los na maquiagem de campo de emergência.
Comunicador	Responsável por explicar o sistema a outros <i>stakeholders</i> através da documentação e materiais de treinamento.	Utiliza documentação da arquitetura para apresentar a estrutura base do sistema.
Assessor	Responsável por supervisionar a conformidade do sistema com padrões e regulamentos legais.	Utiliza documentação da arquitetura para certificar aplicações de determinadas sanções legais e regulatórias.
Avaliador	Responsável por realizar uma avaliação formal da arquitetura (e sua documentação) contra alguns critérios claramente definidos.	Avaliando a habilidade da arquitetura em entregar o comportamento exigido e os ASRs.
Infraestrutura		

Administrador de Banco de Dados	Envolvido em muitos aspectos de armazenamento de dados, incluindo <i>design</i> do banco de dados, análise de dados, modelagem e otimização de dados, instalação de software de banco de dados, monitoramento e administração de segurança do banco de dados.	Compreender como os dados são criados, usados e atualizados por outros elementos arquitetônicos e quais propriedades os dados e o banco de dados devem ter para que o sistema global atinja seus objetivos de qualidade.
Administrador de Rede	Responsável pela manutenção e supervisão de hardware e software em uma rede informática. Isso pode incluir a implantação, a configuração, a manutenção e monitoramento dos componentes da rede.	Determinando cargas de rede durante vários perfis de uso, entendendo os usos da rede.
Administrador de <i>Middleware</i>	Responsável por manter, monitorar e administrar ambientes que fazem a integração entre diversas camadas do sistema e entre outros sistemas com diferentes protocolos de comunicação, plataformas e dependências do sistema operacional.	Compreendendo os componentes da arquitetura apoiam nas atividades através do entendimento das comunicações entre os sistemas.
Implantador	Responsável por implantar o sistema completo ou partes do sistema a partir do esforço de desenvolvimento, tornando-o operacional.	Cumprindo sua função comercial alocada, compreendendo os elementos arquiteturais que são entregues para serem instalados e sua responsabilidade geral em relação à função do sistema.
Administrador do Sistema	Responsável por manter o sistema implantado em execução, corrigindo erros e fornecendo aprimoramentos ao sistema ao longo de sua vida (incluindo a adaptação do sistema para usos não previstos originalmente).	Compreendendo as ramificações de uma mudança.
Equipe de Suporte	Responsável por fornecer suporte aos usuários do produto ou sistema quando estiver em execução.	Entendimento das documentações da arquitetura para possibilitar o conhecimento da estrutura do sistema.
Externo		
Representante de Sistemas Externos	Responsável pela gestão de um sistema com o qual este deve interoperar e sua interface com o sistema.	Definir o conjunto de acordos entre os sistemas.

Fornecedor	Responsável por criar e/ou fornecer o hardware, software ou infraestrutura para que o sistema funcione.	Através do conhecimento da arquitetura do sistema, um fornecedor pode realizar oportunidades de evolução do sistema com ofertas de software e hardware.
------------	---	---

4.7 Trabalhos Relacionados com Processos de Identificação de *Stakeholders*

O conhecimento sobre a identificação de *stakeholders* em sistemas de software se dá a partir do trabalho de Pacheco e Tovar [51], que faz uma revisão sistemática da literatura sobre a identificação de *stakeholders* e elabora um *overview* do estado da arte. Como resultados, os autores trazem os impactos da identificação dos *stakeholders* na qualidade dos requisitos de software e as práticas implementadas para a realização do trabalho de identificação, além de descrever os estudos e suas contribuições.

Outro trabalho relacionado com a identificação de *stakeholders* é o de Fassin [22], que faz uma análise da origem da teoria dos *stakeholders* e apresenta em perspectiva um *framework* gráfico. Este discute sobre a definição de *stakeholders*, identificação e categorização de *stakeholders*, para focar na origem estratégica do modelo de *stakeholder*. A ambiguidade e a imprecisão do conceito de *stakeholder* são discutidas, examinam-se os impactos de lacunas importantes do *framework* de *stakeholders* como, os limites e o nível do ambiente corporativo.

Ainda sobre a identificação de *stakeholders*, Power [54] desenvolve um modelo que ajuda organizações a entender a diversidade de *stakeholders* que influenciam seus esforços de desenvolvimento de produtos, fornecendo meios de quantificar a natureza de seus participantes e entendendo como os processos ágeis e enxutos facilitam a aplicação da teoria dos *stakeholders* no desenvolvimento de produtos de software.

Os processos de identificação de *stakeholders* são atividades presentes na ES. Neste sentido, abordagens teóricas e empíricas são empregadas com maior frequência para investigar o aumento da gama de fenômenos em ES, especificamente na elicitação de requisitos como parte da ER [50]. O estudo empírico de Pacheco e Garcia [50] apresentou como processos de identificação de *stakeholders* podem afetar a qualidade de requisitos e como consequência a qualidade do desenvolvimento do software. Neste estudo, evidenciou-se que engenheiros de software precisam identificar, caracterizar e gerenciar todos os pontos de vista de diferentes tipos de *stakeholders*. É, portanto, sempre necessário realizar uma avaliação de adaptação das contribuições dos *stakeholders* e seus interesses no projeto. Logo, segundo Sommerville [63] a identificação de *stakeholders*, incluindo a identificação de suas necessidades e expectativas, é mal alcançada em projetos de software, provavelmente porque este processo é visto erroneamente como uma tarefa automaticamente evidente onde usuários diretos, clientes e equipe de desenvolvimento são os únicos interessados.

Relacionando a ER com SoS, tem-se o trabalho de Barden e Evans [4], onde os autores salientam que a integração de várias equipes de desenvolvimento de requisitos e inúmeras equipes de projeto de sistema e subsistema dificultam o processo de elicitação de requisitos em SoS. Para isso, eles sugerem a composição de um Grupo de Avaliação de Requisitos capacitado, formado por alguns *stakeholders* como contratante, cliente e *stakeholders* usuários. Este grupo deve monitorar, tomar decisões e fornecer orientação sobre todas as atividades de desenvolvimento de requisitos durante o ciclo de vida conceitual e do ciclo de vida de desenvolvimento. Esta técnica foi utilizada pelo Programa de Gerenciamento do Exército americano e encontrou algumas dificuldades, como lidar com lacunas e problemas de requisitos ou lidar com mudanças de requisitos significativos para a especificação de alto nível e com documento de necessidades do usuário. Essas atividades foram as que mais tomaram tempo dos envolvidos no grupo.

Ainda relacionando a ER com SoS, são apresentados dois trabalhos recentes de Vierhauser, Rabiser e Grünbacher. No primeiro, de 2015, juntamente com Aumayr [66], os autores estudam o monitoramento de requisitos. Eles afirmam que as abordagens de monitoramento de requisitos são uma solução viável para verificar as propriedades do sistema em tempo de execução. No entanto, as abordagens existentes não consideram adequadamente as características dos SoS: existem diferentes tipos de requisitos em diferentes níveis e em diferentes sistemas; requisitos são mantidos por diferentes *stakeholders*; e os sistemas são implementados usando diversas tecnologias. Assim com essas premissas descrevem um modelo de monitoramento de requisitos tridimensionais (RMM) para SoS. Este modelo é avaliado usando um *framework* de monitoramento existente e um simulador das partes em execução deste SoS. Os resultados indicam que o modelo é suficiente para suportar os requisitos de monitoramento de SoS.

No segundo trabalho, de 2016, os autores [65], reforçam que os requisitos de SoS são de propriedade de diversos *stakeholders* e que existem em níveis diferentes, para diferentes sistemas e em diferentes artefatos. Muitas vezes se sobrepõem, se cruzam, ou até mesmo conflitam-se, e alocá-los a sistemas ou componentes específicos nem sempre é possível. Além disso, os requisitos são decorrentes de processos físicos, resultando na necessidade de considerar cuidadosamente as dependências entre software e hardware. Com base nestas dificuldades eles desenvolveram uma ferramenta (ReMinds) que pode ser utilizada para extrair eventos e dados em tempo real para definir requisitos.

Ncube [48] afirma que a SoSE requer conjuntos diferentes de competências e, em significativa extensão, envolve diferentes (múltiplos) *stakeholders*. Em SoS há *stakeholders* tanto para o SoS como para os próprios sistemas constituintes. Esses grupos de *stakeholders* têm seus próprios objetivos e contextos organizacionais que formam sua expectativa em relação ao SoS. Os *stakeholders* de um SoS podem ter conhecimento limitado das restrições e planos de desenvolvimento para os sistemas constituintes. Enquanto que os *stakeholders* de sistemas constituintes podem ter pequeno interesse no SoS, podendo dar baixa prioridade aos SoS ou podendo resistir as demandas do SoS para seus sistemas. Esta competição de interesses dos *stakeholders* estabelece o complexo ambiente para o engenheiro e arquiteto de sistemas do SoS.

Os ambientes inter-organizacionais são formados por um conjunto de organizações que possuem características diferentes e colaboram para alcançar metas comuns. Ambientes inter-organizacionais possuem características semelhantes aos SoS. Analisando sistemas inter-organizacionais, que assim como SoS envolvem muitos *stakeholders*, a abordagem proposta por Costa e Cunha [16] para esses sistemas sinaliza o uso de redes sociais como forma de identificação e caracterização dos *stakeholders*. Está dividida em 5 etapas, começando pelo entendimento da dinâmica e contexto das redes, passando a identificação de um conjunto inicial de *stakeholders*, depois o refinamento desse conjunto inicial, para então caracterizar os *stakeholders* e caracterizar as redes sociais. Para a identificação, os princípios são baseados na ideia de que a identificação de um *stakeholders* pode levar à identificação de outros *stakeholders*. Para a caracterização dos *stakeholders* é necessário preencher uma espécie de ficha para cada *stakeholder* ou grupo de *stakeholders* com 5 campos: relevância, interação organizacional, tipo de relação, perfil e objetivos. Estas informações podem ser utilizadas como base de conhecimento para o cenário de negócios caracterizar a rede social.

Ballejos e Montagna [3] propõem uma abordagem para identificação de *stakeholders* em sistemas inter-organizacionais. Eles propõem que uma equipe específica da interorganização comece com uma lista de *stakeholders* que estão dentro das organizações e são conhecidos. Define perfis para os *stakeholders* serem enquadrados, como operadores, desenvolvedores, clientes, entre outros. Associa os *stakeholders* com o perfil pré-definido. Então executa critérios de seleção, definindo qual *stakeholder* é interno na organização ou na interorganização e qual *stakeholder* é externo a ambas. Critérios identificam o *stakeholder* em diferentes ponto de vista, inicialmente foram considerados: funcional, localização geográfica, conhecimento/habilidades e nível na hierarquia. Assim, além de identificar os *stakeholders*, é possível elaborar uma matriz de interesses e influência dos *stakeholders* na organização.

Assim como sistemas inter-organizacionais, SoSs frequentemente envolvem um grande número de *stakeholders* e com interesses diversos. Consequentemente mais *stakeholders* geram mais preocupações e maiores dificuldades de envolver todos *stakeholders* necessários. O processo de identificação de *stakeholders* é uma tarefa crítica para o sucesso, no entanto muitos autores consideram uma tarefa padrão e não explicam o processo de identificação [3]. Conforme Demchak, Farcas e Kruger [19], um dos principais desafios para a construção de sistemas que atende às necessidades de um grande número de *stakeholders* está na integração e reutilização de sistemas existentes, ao mesmo tempo em que inclui-se novas funcionalidades. Essas características geram a compreensão e modelagem tanto dos sistemas existentes quanto dos requisitos emergentes, criando então sistemas robustos, performantes e sustentáveis. Dessa forma, a identificação de *stakeholders* é fator decisivo do desenvolvimento de todos os tipos de sistema, desde pequeno até ultra grande porte.

Hester, Bradley e Adams [28] sugerem uma abordagem para SoS que combina a tipologia de classificação de Mitchell [45] e a classificação de atitude de Savage [60]. Assim, montando uma matriz que define nas colunas a classificação de *stakeholders* conforme o número de atributos definidos por Mitchell (nenhum é Indefinido, um é Latente, dois são Expectantes e três são Definitivos)

e nas linhas a classificação de atitude de Savage (suporte, misto, sem suporte e marginal). Essa matriz é preenchida com os atributos N (nenhuma ação), M (monitorar), D (defender), C (colaborar) e I (envolver). Assim, em tempo de execução de um trabalho, os principais *stakeholders* são definidos entre as colunas Definitivos até Latentes, porém, precisa-se identificar todos os *stakeholders* que estão nas linhas da coluna Definitivos para depois buscarem os *stakeholders* Latentes e assim por diante até os Expectantes. Nesta abordagem não há limite para a identificação e classificação dos *stakeholders* em projetos de SoS. Há uma ordem de quais categorias e tipos de *stakeholders* são os mais importantes, porém não há menção de como os *stakeholders* serão abordados e propriamente identificados.

O trabalho de Lim e Ncube [41] propõem utilizar o poder de redes sociais e *crowdsourcing* para identificar e priorizar os *stakeholders* em projetos de SoS. O método proposto faz os *stakeholders* recomendarem outros *stakeholders*, construindo uma rede social de *stakeholders*, e prioriza os *stakeholders* utilizando métricas de redes sociais. A priorização pode, por exemplo, utilizar algoritmos para definir o grau de centralidade onde há a priorização dos *stakeholders* que mais forem recomendados por outros *stakeholders* ou a priorização dos *stakeholders* que são amplamente recomendados por grupos diferentes de *stakeholders*. O artigo descreve o método e discute os pontos fortes e limitações da aplicação do método em projetos de SoS.

Muller [46] traz uma série de exemplos em vários domínios (áreas de atuação) para ilustrar a abrangência de SoS. A partir de observações de muitas arquiteturas em execução, eles discutem a falta de conscientização dos *stakeholders* no sistema constituinte do SoS. Esta falta de consciência pode resultar em problemas durante a integração, comissionamento ou implantação de sistemas em seu contexto mais amplo.

Fang e DeLaurentis [21] analisam o problema de planejamento no desenvolvimento de SoS reforçando as restrições de recursos, incerteza e interesses conflitantes dos *stakeholders* como causadores dessa dificuldade. Diferentes *stakeholders* podem ter planos conflitantes que exigem comunicação, negociação e coordenação. Com base neste problema, os autores propõem um *framework* descentralizado de tomada de decisão *multi-stakeholder* onde os *stakeholders* locais conduzem o planejamento do desenvolvimento para benefícios individuais enquanto que os *stakeholders* no nível SoS desenham um mecanismo de coordenação para facilitar a comunicação entre os *stakeholders* dos sistemas constituintes para alcançar um resultado harmonioso.

Rajabalinejad [55] busca determinar significativamente os valores ou preferências dos *stakeholders* de um SoS. Sugere um método baseado na teoria da probabilidade para incorporar flexibilidade e tolerância, para então construir um *framework* básico para coletar, processar e combinar informações de incertezas. Este *framework* introduz o 'grau de consenso' como uma métrica para ranquear os valores identificados, esta métrica representa o consentimento dos *stakeholders* no SoS e pode ser utilizada na arquitetura, nas decisões de *design* entre outras situações.

Tanto o trabalho de Lim e Ncube [41], quanto os trabalhos de Ballejos e Montagna [3] e Salado e Nilchiani [59] são grandes contribuições para a identificação de *stakeholders* em sistemas grandes e complexos. Estes estudos foram utilizados como referência para o objetivo do trabalho

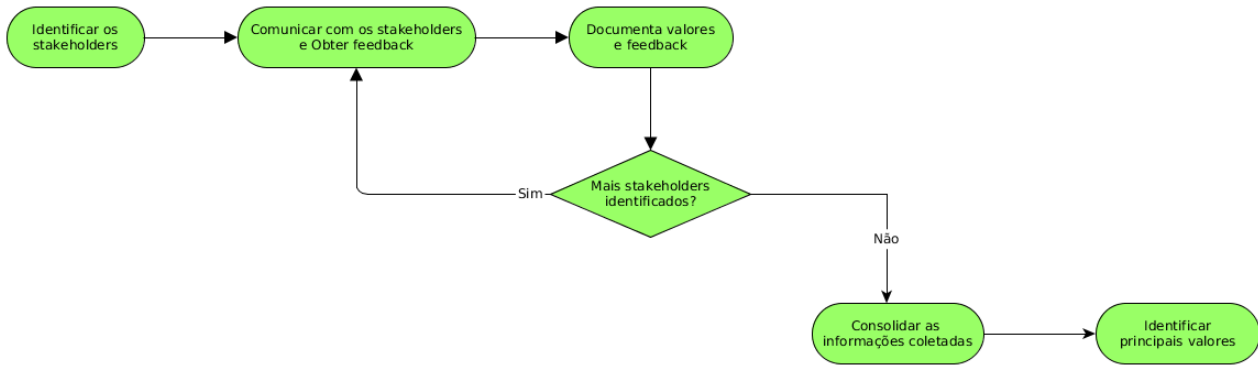


Figura 4.5 – Diagrama do processo de identificação de *stakeholders* e comunicação com eles. Fonte: [55] p.2

de definir os principais *stakeholders* ou os *stakeholders* relevantes para as definições de arquitetura em SoS que é o foco desta dissertação. Apesar disto, nem este nem os demais trabalhos anteriores tem o enfoque sobre a perspectiva de ASR em SoS, uma vez que esses requisitos são os que tem maior impacto na criação e manutenção do *design* de arquitetura de um SoS.

Bons métodos conhecidos de identificação de *stakeholders* podem funcionar em projetos pequenos ou monolíticos mas ficam aquém dos projetos de SoS e em projetos interorganizacionais devido a incapacidade de atender a complexidade e ao grande número de *stakeholders* envolvidos [41] [3]. SoS frequentemente envolvem um grande número de *stakeholders* e com interesses diversos. Conforme Rajabalinejad e Bonnema [55], é preciso tempo e esforço para alcançar um conjunto completo de *stakeholders*. A Figura 4.5 apresenta um diagrama funcional para a identificação dos *stakeholders* e a comunicação com eles. Neste diagrama em tempo de comunicação com os *stakeholders*, novos *stakeholders* podem ser identificados, ressaltando que a documentação dos valores e *feedbacks* tem alta importância no processo de identificação.

Para descobrir novos *stakeholders* e chegar a um conjunto completo deles, pode-se usar, por exemplo, o conjunto de perguntas sugeridas por Salado e Nilchiani [59]. Esses afirmam que as abordagens tradicionais para a identificação dos *stakeholders* centram-se em responder a pergunta quem, *who*, interage com o sistema. No método proposto, eles sugerem explorar as relações, descobrindo os *stakeholders* iterativamente avaliando as perguntas o porquê (*why*), o como (*how*), o onde (*where*) e o que (*what*) do sistema e assim cada *stakeholder* direto e indireto é identificado. Este conjunto de perguntas está representado no diagrama funcional na Figura 4.5.

5. EMBASAMENTO PARA CONCEPÇÃO DO PROCESSO E MODELO

Este capítulo apresenta as principais definições extraídas da Revisão de Literatura. Estas definições foram utilizadas para embasamento na construção do processo e modelo de identificação dos principais *stakeholders*. A Revisão da Literatura executada não foi uma Revisão Sistemática da Literatura como proposto por Kitchenham [38], porém a organização desta Revisão da Literatura seguiu um processo inspirado no processo proposto pela autora. Os detalhes da execução da Revisão de Literatura estão no Anexo A.

Com a Revisão da Literatura, buscou-se elencar os principais conceitos relacionados à identificação de *stakeholders* em sistemas tradicionais, complexos, distribuídos e inter-organizacionais. Estes precisam esclarecer quais são os tipos de *stakeholders*, de que maneira os *stakeholders* são identificados, tipos de requisitos, relevância, entre outros fatores que são necessários para a construção, manutenção e evolução de um SoS.

Cada trabalho extraído e selecionado na Revisão da Literatura faz afirmações importantes que destacam pontos que foram considerados na elaboração do processo e modelo propostos nesta dissertação. A seguir tem-se uma listagem de afirmações mais relevantes dos 18 trabalhos selecionados no contexto do trabalho proposto.

- **Muller (2016) [46]:** *"Observa-se que surpreendentemente poucos stakeholders de sistemas constituintes de um SoS estão cientes da natureza SoS de seu contexto."*

"Esta classificação (os tipos de SoS) sugere que há uma consciência razoável da natureza SoS em muitos níveis de SoSs. No entanto, em muitos casos, essa consciência é limitada a um pequeno grupo de diretores e engenheiros de sistemas."

"Com base nos casos analisados, a maioria dos stakeholders que trabalham em um sistema constituinte tem falta de conhecimento sobre o SoS. Falta de conhecimento do sistema como um todo."

"Os arquitetos precisam trabalhar com os stakeholders da sua própria empresa, de empresas cliente e de empresas fornecedoras."

"Para os SoSs, argumentamos que a maioria dos stakeholders dos sistemas constituintes estão insuficientemente conscientes do próximo nível de SoS que engloba o sistema de interesse."
- **McCaughin e White (2016) [44]:** *"Para uma primeira atividade, os oito tipos de stakeholders têm as seguintes tendências dominantes: Proprietários: controle (extroversão); Empregados: autonomia (introversão); Fornecedores: retorno (extroversão); Clientes: valor (introversão); Reguladores: controle e retorno; Público: autonomia e valor; Concorrentes: retorno e autonomia; Colaboradores: controle e valor."*

"A maioria dos papéis que um stakeholder assume são menos dominantes, apenas uma ou duas tendências se tornam dominantes."

"Identifique os principais stakeholders, em particular, os 20% dos principais stakeholders representam 80% das interações. Categorize os principais stakeholders de acordo com os oito tipos de stakeholders."

- **Yue e Barkley (2015) [69]:** *"Todos os profissionais de sistemas reconhecem a importância da análise e tratamento dos stakeholders e várias estratégias foram desenvolvidas."*

"Ilustra a necessidade de mudar os limites do sistema à medida que o SoS evolui e a importância dos stakeholders seniores - particularmente quando o papel do principal patrocinador muda entre as organizações."

- **Fang e DeLaurentis (2015) [21]:** *"Diferentes stakeholders podem ter planos mestres conflitantes que exigem comunicação, negociação e coordenação."*

"O problema torna-se como escolher os contratos de transferência adequados para induzir os stakeholders individuais a elaborarem planos de desenvolvimento que possam beneficiar o stakeholder global."

- **Safwat e Senousy (2015) [58]:** *"Sistemas de Grande Porte (ULS do inglês Ultra-Large-Scale), como SoS, tem ganhado consideráveis reflexões sobre as atividades de desenvolvimento de sistemas, já que a escala é incomparável para os sistemas tradicionais, uma vez que milhares de diferentes stakeholders estão envolvidos no desenvolvimento de software, cada um deles tem interesses diferentes."*

"A ER é essencialmente uma atividade de colaboração social, na qual os stakeholders envolvidos (por exemplo, clientes e desenvolvedores) devem trabalhar em conjunto para se comunicar, elicitar, negociar, definir, confirmar e, finalmente, criar os requisitos (incluindo funcional e requisitos não funcionais) para que o sistema seja implementado ou atualizado."

"Criação de novo paradigma de desenvolvimento que suporte as seguintes atividades:

- *O sistema de ultra grande porte inclui milhares ou mesmo centenas de milhares de stakeholders e é óbvio que o atendimento de todos os stakeholders é impossível;*

- *A diversidade de stakeholders com diferentes culturas, preocupações, políticas, processos de negócios, que precisam de maneiras de responder aos seus diferentes requisitos, conflitos e mudanças;*

- *Por um lado, os requisitos relevantes para todos os sistemas constituintes e a solução de integração devem ser entendidos, atualizados e transitados para a arquitetura. Além da flexibilidade da arquitetura, mais adaptabilidade do Sistema ULS é para as mudanças no ambiente operacionais;*

- *Os requisitos devem ser monitorados e gerenciados para garantir que nenhum indivíduo ou organização possa mudar sensivelmente o sistema sem entender e, talvez, obter aprovação dos outros participantes;*

- *Atividades de desenvolvimento distribuídas em muitas organizações que exigem novos métodos para compatibilidade de requisitos, verificações e detecções de problemas;*
 - *Contínua evolução em um ambiente operacional onde o número de mudanças é muito grande e o período entre o tempo de design e o tempo de execução não é claro;*
 - *Resumo das arquiteturas de sistemas, suas interfaces e o contexto para a evolução e adaptação dos Sistemas ULS;*
 - *Resposta de requisitos dinâmicos e rápidos para manter os recursos operacionais;"*
- **Rajabalinejad e Bonnema (2014) [55]:** *"É preciso tempo e esforço para alcançar um conjunto completo de stakeholders."*
 - "Alguns novos stakeholders podem ser encontrados durante a comunicação com stakeholders conhecidos."*
 - "Os valores e os feedbacks dos stakeholders são de grande importância e devem ser bem documentados."*
 - "Stakeholders geralmente têm competição ou conflitos de interesses e prioridades."*
 - "Pode acontecer de um stakeholder não ter interesse no SoS, mas apenas em um sistema constituinte dependente do SoS. Estes stakeholders exigem atenção na comunicação com os demais stakeholders, elicitação de informações, documentação e integração de informações."*
 - **Lim e Ncube (2013) [41]:** *"Um grande número de projetos falha porque eles omitem os stakeholders ou envolvem indivíduos que não têm conhecimento, tempo ou interesse para o envolvimento adequado no projeto."*
 - "A maioria, se não todos, projetos de SoS envolvem milhares ou mesmo centenas de milhares de stakeholders, com diferentes stakeholders tendo diferentes níveis de influência no projeto. Como resultado, além de identificar os stakeholders, um passo crucial na análise de stakeholders é a priorização de stakeholders com base na influência que o mesmo possui do projeto."*
 - "No trabalho, os autores narram que suas experiências como analistas de stakeholders e a revisão da literatura revelaram as seguintes deficiências com os métodos atuais de análise de stakeholders. Em primeiro lugar, alguns métodos são susceptíveis de perder stakeholders, e outros encontram stakeholders irrelevantes ou "non-stakeholders". Em segundo lugar, a maioria dos métodos trata todos os stakeholders como igualmente influentes e todos os representantes dos grupos são igualmente adequados. Esses métodos podem ser suficientes em projetos pequenos ou monolíticos, mas ficam aquém do sistema de projetos de sistemas devido à sua incapacidade de atender à complexidade e ao grande número de stakeholders envolvidos."*
 - "Como SoS tendem a ter limites pouco claros, é impossível identificar todos os stakeholders de um SoS."*
 - **Arrichiello (2012) [1]:** *"É preciso identificar os stakeholders envolvidos com o sistema ao longo de seu ciclo de vida, suas necessidades, expectativas e desejos."*

"Os principais grupos de stakeholders envolvidos no desenvolvimento de um SoS são os clientes, os usuários e os desenvolvedores."

- **Sommerville (2012) [64]:** *"Um problema duradouro na ER é em como identificar os stakeholders a serem consultadas e como ajudá-los a articular seus requisitos para um sistema."*

"A derivação dos requisitos envolve extensas discussões e consultas com stakeholders do sistema."

"Os problemas da identificação de stakeholders são agravados em situações em que o sistema a ser desenvolvido abrange várias organizações e esses stakeholders são distribuídas entre essas organizações."
- **Ireland; Rapaport e Omarova (2012) [30]:** *"Projetos ainda mais complexos do que os SoS "tradicionais", muitos dos quais militares, como o terrorismo, disputas internacionais e mudanças climáticas, exigem uma metodologia do sistema simples para identificar os stakeholders, os limites e as possíveis soluções, não são abordados em um Body of Knowledge."*

"Projetos de SoS exigem sistemas de pensamento para determinar os stakeholders, os limites dos projetos e os soft systems methods para desenvolver uma solução potencial."

"Sugere o uso da metodologia SAST-Strategic Assumptions Surfacing and Testing para identificação de stakeholders. Essa metodologia baseia-se em quatro princípios fundamentais: apoio participativo, adverso, integrador e gerencial."
- **Yang-Turner e Lau (2011) [68]:** *"No DICODE, existem dois grupos principais de stakeholders: (1) usuários de domínio que são parceiros de casos de uso. Eles se beneficiarão diretamente do resultado do projeto; (2) tecnólogos que são os parceiros técnicos e fornecem soluções técnicas para os parceiros de casos de uso. Eles se beneficiam da pesquisa científica e técnica realizada durante o processo."*

"Ambos os grupos de stakeholders têm responsabilidades iguais em relação ao objetivo do projeto. No entanto, não está claro se eles têm responsabilidades iguais na fase de elicitação do requisito. "
- **Liang; Avgeriou; He e Xu (2010) [39]:** *"Nas abordagens ER tradicionais, os stakeholders comunicam os requisitos através da documentação de requisitos, que funciona para um número pequeno e fixo de stakeholders, mas é insuficiente quando o alcance dos stakeholders de um sistema é imprevisível (por exemplo, usuários de sistemas baseados na Web)."*

"Os requisitos dos Sistemas ULS são inerentemente conflitantes, irreconhecíveis e diversos devido a uma grande variedade de stakeholders com necessidades diferentes, conflitantes, complexas e em mudança."

"Nas abordagens tradicionais de ER, os stakeholders comunicam os requisitos através da documentação de requisitos, que funciona para um número pequeno e fixo de stakeholders, mas é insuficiente quando o alcance dos stakeholders de um sistema é imprevisível."

"Autores propõe, usar o método de tagging colaborativa para marcar as declarações de requisitos pelos próprios stakeholders e, em seguida, essas marcações (tags) colaborativas (conhecimento coletivo) são transformadas em ontologias de requisitos com semântica formal e expressividade mais rica."

- **Costa e Cunha (2010) [16]:** *"Tem contribuições de quatro áreas principais: critérios para identificar os stakeholders nos sistemas inter-organizacionais, análise de redes sociais, teorias de modelos de negócios e teorias sociais."*

"A definição de stakeholder para os autores é semelhante a Freeman: cada indivíduo que pode influenciar ou ser influenciado pela rede social - direta ou indiretamente - deve ser considerado como um participante."

"A identificação dos stakeholders em redes sociais assume uma particular relevância devido aos aspectos culturais e sociais que podem influenciar seu comportamento."

"Os princípios baseiam-se na ideia de que cada participante identificado pode levar à identificação de outros. Por esse motivo, eles são aplicados em um processo iterativo. O analista interrompe seu estudo quando os novos stakeholders identificados têm influência residual no modelo de negócios."

"Uma ficha deve ser preenchida para cada stakeholder ou grupos de stakeholders."

"Deve ser dada especial atenção aos stakeholders importantes com grande influência na rede."

- **Boonstra e Vries (2008) [10]:** *"Os stakeholders geralmente podem ter percepções diferentes de um sistema inter-organizacional e interpretá-lo de maneira diferente. Em muitos casos, os interesses percebidos no sistema inter-organizacional diferem entre os stakeholders envolvidos."*

"Questionamentos relevantes que ajudam a identificar grupos e indivíduos relevantes, como: Quem são (iniciadores, patrocinadores); Quem adota...; Quem receberá...; Quem será impactado...; Quem ganha...; Quem perde..."

"Com base em entrevistas exploratórias, foi preparada uma lista contendo stakeholders relevantes (e representantes típicos de cada grupo de stakeholder que poderiam ser abordados para participar da pesquisa."

"Stakeholders não são entidades isoladas com interesses e interpretações fixas, mas eles negociam e influenciam mutuamente, o que pode levar a novas perspectivas e novas interpretações do sistema."

- **Boonstra; Boddy e Bell (2008) [9]:** *"Os resultados de um sistema inter-organizacional dependerão da interação dos stakeholders e dos aspectos dos contextos interno e externo que aqueles que promovem o sistema criam para implementá-lo."*

"As entrevistas continham três conjuntos de perguntas. O primeiro conjunto foi direcionado para identificar os stakeholders mais relevantes no momento do estudo e o segundo na avaliação

de seus interesses e poder em relação ao sistema. O terceiro conjunto foi direcionado para identificar suas atitudes em relação ao sistema."

- **Demchak; Farcas, C; Farcas, E e Kruger (2007) [19]:** *"Um dos principais desafios para a construção de sistemas que atendem às necessidades de um grande número de stakeholders é integrar e reutilizar sistemas existentes, ao mesmo tempo em que adicionar novas funcionalidades."*

"Arquiteturas SOAs podem criar SoS ao mapear sistemas existentes em serviços, orquestrando a comunicação entre estes serviços."

"O processo apresentado é orientado, especificamente, para a integração rápida de SoS."

"Processo dividido em 3 fases: Elicitação de Serviço; Arquitetura Rich Services; Definição de Arquitetura de Sistema."

- **Blair; Boardman e Sauser (2007) [6]:** *"Systemigrams são utilizados para diálogos entre stakeholders chave para apresentar/demonstrar assuntos chaves."*

"Uma técnica de story-boarding é usada para uma apresentação do systemigram para envolver a público de stakeholders em um processo iterativo de diálogo/debate. Esta atividade de interação de stakeholders através de storyboard é o cerne da técnica systemigram, refletindo seu poder como um sistema de aprendizagem conceitual e apreciativo."

- **Pouloudi e Whitley (1997) [53]:** *"O número de stakeholders envolvidos no desenvolvimento e uso do sistema inter-organizacional é muito maior que o da maioria dos sistemas organizacionais tradicionais. Além disso, uma vez que as inter-relações destes stakeholders são complexas e muitas vezes indiretas, elas estão em maior ou menor posição de influenciar e, ao mesmo tempo, ser afetadas pela função de um sistema de informação."*

"Embora seja reconhecida a importância de identificar os stakeholders, os stakeholders surgem como produto final de um processo de identificação de stakeholders que não é explicado na maioria dos trabalhos."

"Um dos principais problemas da falta de uma abordagem sistemática de identificação de stakeholders são as listas genéricas de stakeholders que não são apropriadas para todos os contextos."

"Os autores fornecem princípios para identificar os stakeholders como uma técnica flexível e dinâmica que permite modificações de acordo com o contexto particular e em diferentes momentos. Os princípios são:

P1: *os stakeholders dependem do contexto específico e do prazo. A implicação para a identificação dos stakeholders é que ele precisa ser um processo dinâmico, que possa suportar a instabilidade e a incerteza da realidade organizacional, pelo qual os novos stakeholders entram no domínio, enquanto outros escolhem ou são obrigados a sair.*

P2: *os stakeholders não podem ser vistos isoladamente. As inter-relações entre as diferentes stakeholders que compõem um dos componentes mais interessantes do estudo do comportamento dos stakeholders; eles revelam uma rede complexa de interações, interesses e poder.*

P3: *a posição de cada stakeholder pode mudar ao longo do tempo. Os stakeholders podem ser forçados a mudar sua posição em relação a outros stakeholders, "adaptar", ou, de fato, podem se beneficiar de uma oportunidade para fazê-lo, já que outros stakeholders reagem às mudanças no ambiente organizacional, imitando ou liderando a aplicação de novos planos, estruturas e programas.*

P4: *as opções possíveis podem diferir dos desejos dos stakeholders. stakeholders podem ser incapazes de planejar efetivamente a realização de seus desejos."*

"A identificação dos stakeholders deve ser um processo iterativo em que o mapa de stakeholders se torne cada vez mais abrangente para cobrir todos os stakeholders relevantes."

Além destes 18 trabalhos, durante a leitura dos trabalhos aceitos, percebeu-se que algumas referências desses eram relevantes para a pesquisa e que não estavam na listagem dos 767 trabalhos retornados pela pesquisa Scopus. Assim, novos trabalhos foram analisados e acrescentados a pesquisa, fazendo parte da fundamentação teórica, bem como da estrutura do processo e modelo proposto. Esses juntamente com estes 18 trabalhos foram previamente apresentados nos Capítulos 2, 3 e 4. Dentre os trabalhos adicionados, destacam-se os 9 trabalhos a seguir. Desses trabalhos foram extraídas as afirmações apresentadas junto a eles:

- **Vierhauser e Rabiser (2016) [65]:** *"Normalmente, SoS não são planejados e são projetados top-down, mas ocorrem gradualmente, muitas vezes como resultado de décadas de desenvolvimento por equipes independentes."*

"É essencial garantir que um SoS esté de acordo com seus requisitos. "

"Os requisitos de SoS são de propriedade de diversos stakeholders e existem em diferentes níveis, para diferentes sistemas e em diferentes artefatos. Eles geralmente se sobrepõem, cruzam ou se conflitam. Atribuí-los a sistemas específicos ou componentes nem sempre é possível."

"O comportamento completo de um SoS surge apenas em tempo de execução, quando os sistemas envolvidos interagem com hardware, sistemas de terceiros e/ou sistemas legados."

- **Vierhauser; Rabiser; Grünbacher e Aumayr (2015) [66]:** *"As arquiteturas de SoS consistem de diversos sistemas implementados por diferentes equipes ou empresas."*

"A independência gerencial e a diversidade de stakeholders em SoS significa que os sistemas são desenvolvidos, mantidos e operados por equipes independentes, muitas vezes por várias empresas diferentes. Essa situação reforça que a hipótese principal de engenharia de requisitos não tem mais validade: hipótese de que um sistema que será construído está sob o controle de um único stakeholder que determina um conjunto consistente de requisitos."

- **Hester; Bradley e Adams (2012) [28]:** *"Os stakeholders existem no centro de qualquer problema de sistemas."*

"Embora a tipologia de Mitchell [45] seja útil, alguns autores argumentam que é insuficiente."

"Uma estratégia para envolver os stakeholders baseada unicamente na classificação (conforme Mitchell [45]), é insuficiente, uma vez que não é responsável pelo suporte ou oposição dos stakeholders a um esforço específico. Para preencher essa lacuna, pode-se utilizar o trabalho de Savage [60] para categorizar os stakeholders."

"Na escala de envolvimento do stakeholder tem a seguinte ordem: envolvido, colaborativo, defende, monitora e sem ação."

"O stakeholder definitivo é aquele que possui poder, legitimidade e urgência."

"Os stakeholder definidos como Envolvidos e Colaborativos devem ser envolvidos no projeto."
- **Pacheco e Garcia (2012) [50]:** *"Uma das causas prováveis para que a identificação dos stakeholders, incluindo a identificação de suas necessidades e expectativas, seja mal alcançada em projetos de software é devido à este processo ser considerado erroneamente como uma tarefa óbvia, na qual usuários diretos, clientes e equipe de desenvolvimento são os únicos stakeholders. Isso também pode ser devido ao fato de que a área de identificação pode ser eliminada ou substituída por opiniões ou conhecimentos obtidos de outras fontes de informação mais acessíveis."*

"Os stakeholders podem variar de um projeto para outro. É, portanto, sempre necessário realizar uma avaliação de adaptação das contribuições dos stakeholders e seus interesses adquiridos em um projeto."

"Atualmente, os métodos de identificação de stakeholders são poucos e não são estruturados, pois cada autor descreve o processo a partir do seu ponto de vista, sem um framework comum de estudo e uma descrição uniforme. O alto nível de heterogeneidade dos estudos analisados não nos permite apresentar uma análise de dados quantitativos. "

"O status atual da identificação de stakeholder mostra diferentes interpretações do escopo do processo. Todas as iniciativas de software referidas contribuem para a melhoria do processo de software, implementando um conjunto de boas práticas da indústria para ER identificadas, reconhecidas e disseminadas. No entanto, eles não explicaram como realizar o identificação de stakeholder."

"Algumas iniciativas fornecem inúmeros exemplos de quem pode ser um stakeholder, estabelecendo categorias genéricas nas quais estes stakeholders podem ser agrupados."

"Nem todos os trabalhos cobrem os mesmos aspectos e, portanto, não são aplicáveis às mesmas situações. Isso torna difícil selecionar um método correto de identificação de stakeholders porque alguns métodos apenas caracterizam os stakeholders, sem atribuir o papel de um stakeholder em um projeto específico, nem analisar a interação dos stakeholders, nem cobrir os aspectos humanos da identificação dos stakeholders."

"Cada projeto de software pode ter diferentes tipos de stakeholders e selecioná-los adequadamente tem um forte impacto na qualidade dos requisitos de software e, conseqüentemente, no sucesso do próprio projeto de software."

"Além disso, são necessários mais trabalhos futuros para desenvolver um método ou modelo para realizar identificação de stakeholders na área ER."

- **Ncube (2011) [48]:** *"A escala e a complexidade dos SoS possuem desafios únicos para a comunidade de ER. As técnicas de ER atuais são inadequadas para enfrentar esses desafios e novos conceitos, métodos, técnicas, ferramentas e processos são necessários."*

"Os problemas são extremamente complexos e não limitados ou estáveis; eles não têm soluções únicas e corretas, mas sim soluções que são melhores ou piores do que outras, e não possuem uma formulação definitiva."

"O SoSE requer um conjunto diferente de perfis e envolve múltiplos stakeholders do que a engenharia de sistemas tradicionais está acostumada. Em SoS, há stakeholders tanto para os SoS quanto para os próprios sistemas constituintes. Esses grupos de stakeholders têm seus próprios objetivos e contexto organizacional que formam sua expectativa em relação ao SoS."

"Os stakeholders dos SoS podem ter um conhecimento limitado das restrições e planos de desenvolvimento para os sistemas constituintes."

"Os stakeholders de sistemas constituintes podem ter pouco interesse nos SoS, podem dar baixa prioridade ao SoS ou podem resistir às demandas de SoS em seus sistemas. Essas competições de interesses dos stakeholders estabelece ambiente complexo dos stakeholders para o engenheiro de sistemas SoS."

"Os requisitos de SoS estão em um nível superior e são definidos em termos de capacidades ao invés de requisitos atômicos, como é na engenharia de sistemas tradicional. Como os requisitos são frequentemente elencados em termos de capacidade mais ampla, a equipe de engenharia de sistemas SoS é obrigada a se envolver com gerentes de SoS, diversos stakeholders e usuários para derivar os requisitos SoS."

- **Power (2009) [54]:** *"O termo stakeholder é frequentemente usado no desenvolvimento de software, predominantemente limitado a clientes, usuários finais ou patrocinadores de projetos. As pessoas não se concentram nas equipes de desenvolvimento de produtos como um grupo diversificado de stakeholders com diversos interesses, onde há uma menção mais profunda dos stakeholders, eles tipicamente tratam a equipe de desenvolvimento ou 'o cliente' como um único stakeholder."*

"O modelo proposto inclui estratégias para identificação e envolvimento com os stakeholders e para classificação dos stakeholders, mapeando-os em seis grupos de stakeholders pré-definidos: Dono do Produto, de Entrega do Produto, Patrocinador do Produto, Clientes do Produto, Conselho do Produto e Equipe Principal."

- **Fassin (2009) [22]:** *"Uma visão geral da literatura sobre a teoria dos stakeholders cria a impressão de que os conceitos em torno do stakeholder são referidos de maneiras confusas."*

"A teoria dos stakeholders, como acontece com qualquer nova teoria, sofre de inúmeras falhas e imperfeições em parte devido à imprecisão, à ambiguidade e à amplitude da própria teoria dos stakeholders e, em parte, ao seu apelo intuitivo de grande alcance, muitas vezes as críticas são implícitas."

"A definição de Freeman, agora clássica, tornou-se a mais aceita das definições de um stakeholder e tem maior precisão do que a versão mais curta "aqueles que podem afetar ou podem ser afetados pela empresa"."

"Devido à globalização e à evolução tecnológica, com melhorias nas comunicações e sistemas de informação, praticamente todos, em todos os lugares, podem "afetar ou ser afetados" pelas decisões e ações de uma empresa comercial. Consequentemente, virtualmente todos devem ser considerados como stakeholders."

"Numerosos artigos acadêmicos concentraram-se na identificação e categorização de stakeholders. No entanto, muito poucos estudiosos continuam ligando sua análise a um esquema gráfico e, portanto, evitam analisar as inconsistências que podem existir entre suas definições e o modelo gráfico."
- **Ballejos e Montagna (2008) [3]:** *"Os ambientes inter-organizacionais são formados por um conjunto de organizações que possuem características diferentes e colaboram para alcançar objetivos comuns."*

"As decisões devem ser tomadas em termos de conjuntos de organizações cujos limites, muitas vezes, não são claros dentro da rede inter-organizacional ao qual se encontram, mas onde cada um ainda mantém sua autonomia."

"Portanto, não há apenas stakeholders dentro da empresa, mas também stakeholders dentro do rede inter-organizacional, que cuidarão dos objetivos comuns ao nível da rede. "

"As abordagens existentes para envolver os stakeholders em um projeto de software não fornecem ferramentas, modelos ou métodos concretos suficientes para identificá-los adequadamente, mesmo em ambientes tradicionais."

"Os tipos de stakeholders serão determinados considerando indivíduos dentro de organizações, em grupos ou na organização toda."

"No nível organizacional, os stakeholders selecionados devem ter algum conhecimento específico das tarefas internas de cada organização que serão afetadas pelo sistema inter-organizacional."

"A principal técnica utilizada foi a entrevista pessoal com as pessoas das entidades envolvidas."
- **Pacheco e Trovar (2007) [51]:** *"A identificação dos stakeholders, bem como as suas necessidades e expectativas, são mal realizadas em projetos de software, provavelmente porque*

este processo é considerado equivocadamente como uma tarefa evidente em que os usuários diretos, os clientes e a equipe de desenvolvimento são os únicos stakeholders. Também pode ser devido ao fato de que a área de identificação de stakeholders ser evitada ou substituída por opiniões ou conhecimentos obtidos de outras fontes de informação mais acessíveis."

"Qualquer processo de identificação de stakeholders que reconhece erroneamente alguém como um stakeholders, provavelmente incluirá requisitos que não correspondem a nenhuma necessidade real."

"Quando a tarefa de identificação de stakeholders não consegue detectar os participantes que são necessários para o projeto de software, as especificações de requisitos não são completas devido à omissão de determinados requisitos relevantes para o sucesso do projeto, o que pode dar origem a especificações inconsistentes."

"As implicações dos processos de identificação de stakeholders na qualidade dos requisitos são evidentes."

"Durante a elicitação de requisitos de software, decide-se o que exatamente deve ser produzido. Nesta fase, a identificação apropriada dos stakeholders é de vital importância como um meio para entender o ambiente em que o projeto de software será desenvolvido e operado e também identificar quais stakeholders participarão no projeto."

"Uma boa interação é vital durante o processo de coleta de requisitos e também entre todos os stakeholders e o sistema para evitar conflitos e problemas de comunicação decorrentes de diferentes pontos de vista."

"Processos de identificação de stakeholders devem desenvolver esquemas para caracterizar e avaliar as relações apropriadas entre todos os stakeholders."

6. PROCESSO E MODELO DE IDENTIFICAÇÃO DOS PRINCIPAIS *STAKEHOLDERS* EM *SoS Acknowledged*

Neste capítulo será apresentada a proposta de processo e de modelo para identificação dos principais *stakeholders* de *SoS Acknowledged*. Antes de apresentar a proposta, decidiu-se buscar na literatura definições dos termos que melhor se adequam ao objetivo do trabalho. Foram escolhidos os termos **processo** e **modelo**. Pois, de acordo com a ISO 15.288 [35] um **processo** é um conjunto de atividades inter-relacionadas ou interativas que transformam entradas em saídas. E conforme a ISO 1.233 [31], um **modelo** é uma representação de um processo, dispositivo ou conceito do mundo real.

Assim, o trabalho será constituído de um processo para a identificação dos principais *stakeholders* de *SoS Acknowledged* que produzirá um modelo que ilustra o resultado do processo. Para a construção deste modelo, organizou-se este capítulo em quatro seções. Na Seção 6.1, apresenta-se as premissas do Processo e Modelo. É nesta seção que as referências da literatura são utilizadas para embasamento para a construção dos processos que geram o modelo. Na Seção 6.2, tem-se a explicação de construção do processo. Na Seção 6.3, apresenta-se o processo de montagem do modelo, com o detalhamento de cada uma das etapas do processo. E, por fim, na Seção 6.4, o formato de representação utilizado para o modelo.

6.1 Premissas para construção do Processo e Modelo

O processo proposto neste trabalho para identificação dos principais *stakeholders* de *SoS Acknowledged* utiliza as referências extraídas da literatura listadas abaixo para a estruturação do processo, limitando escopo e elencando os pilares que embasam as ações implementadas no processo e no modelo.

- As mudanças nos sistemas são baseadas na colaboração entre o *SoS* e o sistema constituinte [27].

Nesta premissa percebe-se a existência de colaboração entre os sistemas. Este fator demonstra o engajamento dos sistemas constituintes nas solicitações do *SoS*. Aponta para um relacionamento entre a equipe do *SoS Acknowledged* com *stakeholders* dos sistemas constituintes.

- *SoS Acknowledged* possuem independência de atuação, orçamento, gestores e recursos próprios. Os engenheiros de *SoS* provem suporte técnico aos gestores de *SoS* [27].

Com essa afirmação entende-se que existe equipe no *SoS Acknowledged* que pode se envolver no desenvolvimento e manutenção do processo aqui proposto. Como o processo é de interesse do *SoS* e esse por sua vez possui recursos próprios para isso, entende-se que a equipe responsável por manter esse processo e modelo será composta dos recursos do *SoS*.

- Muitos projetos não criam ou não mantêm as documentações atualizadas [5].

Essa afirmação demonstra que construir o modelo baseado em documentações dos sistemas não é uma prática viável. Além de não estarem atualizadas, existe a diferença de documentações entre os sistemas constituintes. Alguns sistemas são legados que possivelmente nem documentação possuem.

- Alguns *stakeholders* dos sistemas constituintes podem ter interesses diferentes dos *stakeholders* do SoS [41].

As necessidades dos *stakeholders* do SoS podem ser diferentes das necessidades dos *stakeholders* do SoS, pois como cada sistema é independente, com gestão própria, a listagem das solicitações e prioridades pode ser, e em muitos casos é, diferente das prioridades do SoS. A negociação da execução de uma solicitação deve ser negociada e precisa envolver ambos, *stakeholders* do SoS e dos sistemas constituintes impactados.

- Quanto aos requisitos de SoS, há mudanças constantes nos requisitos, devido aos requisitos serem de diversas disciplinas, fragmentados, conflitantes, instáveis, irreconhecíveis e eventualmente não completamente definidos, o que os tornam difíceis de analisar [48].

Percebe-se a dificuldade de trabalhar com requisitos em SoS. As constantes mudanças reforçam a necessidade de monitoramento e colaboração entre os envolvidos.

- Devido a dificuldades, como a competição entre requisitos, métodos de design tradicionais não são adequados para SoS [49].

Com esta afirmação, Overton sugere que se considere outra abordagem para implementação de um SoS. A implementação *middle-out*, que está focada em partir da descrição do problema conforme um *Design Pattern* e desenhar a solução de acordo com os requisitos de um *Pattern*. Sabe-se também que além desses requisitos de origem de padrões, tem-se requisitos oriundos de necessidades dos *stakeholders* que também precisam ser considerados. Esses com origem numa outra abordagem de desenvolvimento, a *top-down*. Por isso, se deve considerar as duas abordagens de desenvolvimento de sistemas, *top-down* e *middle-out*, para a elicitação dos requisitos.

- Todos os níveis de um SoS possuem grande número de requisitos, em um exemplo do exército americano um SoS possui mais de 65 mil requisitos [4].

Percebe-se que analisar detalhadamente todos os requisitos é uma tarefa complexa e custosa que envolverá muitas pessoas, de acordo com a urgência da análise. Dependendo do tamanho da equipe e dos recursos do SoS essa atividade pode não ser viável. Por isso, analisar todos os requisitos detalhadamente pode não ser uma opção.

- A combinação do ciclo de vida de um sistema individual com o ciclo de vida de um SoS é um desafio para ES [13].

Um desafio para a equipe do SoS é lidar com sistemas constituintes que estão em construção, outros em manutenção, outros que são legados com limitações tecnológicas, entre outras dificuldades que surgem ao trabalhar com momentos diferentes dos sistemas constituintes. Bem diferente de trabalhar com todos os sistemas sendo construídos conjuntamente.

- A comunicação entre os sistemas é o que compõe um SoS [42].

A comunicação entre os sistemas constituintes e o SoS é o que mantém o SoS funcionando, uma vez que os sistemas constituintes são os responsáveis pelo fornecimento das capacidades do SoS. A maneira que o SoS entrega capacidades únicas aos *stakeholders* é através da comunicação, orquestração e enriquecimento das capacidades dos sistemas constituintes que os sistemas constituintes sozinhos não conseguiriam entregar.

Com base em cada um dos itens listados acima, pode-se extrair três pilares que estruturam o envolvimento dos arquitetos do SoS no momento da construção e manutenção do processo e do modelo de *stakeholders*: Colaboração, Abordagens de Desenvolvimento e os ASRs. É através destes pilares que o processo e modelo para identificação dos principais *stakeholders* de SoS *Acknowledged* foi construído.

- A Colaboração entre os arquitetos e os *stakeholders* é necessária para que se desenvolva algo que realmente seja necessário e atinja as expectativas dos *stakeholders*.
- A atenção às abordagens de desenvolvimento, fazendo com que além de atender as necessidades dos *stakeholders*, também se faça de acordo com o desenho de arquitetura ideal seguindo os *Design Patterns* de mercado.
- A correta identificação de ASRs que serão mais um insumo para que os arquitetos façam escolhas corretas de *Design Patterns* e atendam às necessidades dos *stakeholders*.

6.2 Apresentação do Processo

Partindo destes pilares, definiu-se que a construção, manutenção e evolução do processo de um SoS pode ficar sob responsabilidade da equipe do SoS no papel do arquiteto. Também que os sistemas constituintes devem considerar o SoS como mais um de seus *stakeholders*, uma vez que solicitações do SoS precisam ser implementadas pelos constituintes. E para a construção do processo deve-se considerar como insumos as necessidades de duas abordagens de desenvolvimento: *top-down* e *middle-out*. Na abordagem *top-down*, considera-se os documentos de requisitos e as solicitações dos *stakeholders*; enquanto que na abordagem *middle-out*, considera-se a os *Design Patterns* para construção do SoS. Acredita-se que a maior parte das solicitações serão oriundas da abordagem *middle-out* por ser a abordagem mais utilizada em SoS, porém solicitações oriundas da abordagem *top-down*, como necessidades de governos (mudanças e novas leis), não podem ser descartadas.

	Solicitação	ASR	Quando	Como	Onde	Quem Solicita	Quem Impacta
SoS				Descrição de como será implementado a nível de SoS	n/a	Stakeholder4	n/a
Sistema Constituinte	Detalhes da solicitação	S	DD/MM/YYYY	Descrição do como cada um dos SC precisa ser envolvido na implementação	SC1	n/a	Stakeholder1; Stakeholder2; Stakeholder5.... Stakeholder_n
SC2					Stakeholder1; Stakeholder3.... Stakeholder_n		
... SCn					... Stakeholder_n		
Módulo do Sistema Constituinte				Caso seja necessário, preencher como os módulos dos SC que serão envolvidos	SC1; SC2; ... SCn	n/a	Stakeholder1; Stakeholder2.... Stakeholder_n
...

Tabela 6.1 – Exemplo das respostas do modelo

O conjunto inicial de *stakeholders* pode ser definido de duas formas: no momento da constituição do SoS ou em SoS que já está operando. Quando um SoS ainda não está constituído, mas há a intenção de fazê-lo, o conjunto inicial de *stakeholders* pode ser definido pelos gestores dos sistemas constituintes. Quando um SoS já existe e está operando, o conjunto inicial dos *stakeholders* pode ser definido pelo gestor ou equipe do SoS. Como apoio a esta tomada de decisão, pode-se escolher representantes de grupos de *stakeholders*, como sugere Arrichiello [1], que afirma que os principais *stakeholders* envolvidos no desenvolvimento de um SoS são os clientes, os usuários e os desenvolvedores.

Uma vez que o conjunto inicial de *stakeholders* está definido, para cada nova solicitação oriunda das abordagens conhecidas para o SoS, o arquiteto do SoS deve selecionar qual das solicitações são impactantes na arquitetura, ou seja, definir qual das solicitações que estão no Repositório de Requisitos são ASRs. Este Repositório de Requisitos é um local comum onde cada solicitação identificada pelos *stakeholders*, incluindo os arquitetos do SoS, são armazenadas e posteriormente analisadas. Nesta análise, arquitetos definem quais solicitações são ASRs, para cada solicitação definida como ASR, deve-se realizar a execução do conjunto de perguntas ao solicitante. Esse conjunto de perguntas é uma adaptação do *Framework* de Zachman [70] e tem por objetivo responder "quando", "como", "onde", "quem solicita" e "quem impacta" está envolvido na solicitação em cada uma das camadas relevantes. Foram definidas três camadas de relevância, a camada correspondente ao SoS, ao Sistema Constituinte e uma camada opcional de Módulo do Constituinte, para os caso em que um Sistema Constituinte possua um subsistema ou esteja dividido em módulos. Cada ASR identificado e perguntas respondidas pelos solicitantes, distribui-se as respostas para cada uma das camadas, assim consegue-se separar e mensurar os impactos no SoS e nos sistemas constituintes. Um exemplo de estrutura dos ASRs e das respostas do processo pode ser visto na Tabela 6.1.

Cada uma das perguntas tem um objetivo e um papel na elaboração do modelo. A pergunta "quando" tem por objetivo definir a urgência de execução da solicitação, definindo uma data para realização da solicitação. A pergunta "como" busca determinar o esforço e as atividades necessárias para atendimento da solicitação. A pergunta "onde" lista todos os sistemas que serão envolvidos no atendimento da solicitação. E por fim as perguntas a pergunta "quem solicita" e "quem impacta" determina quem são os *stakeholders* envolvidos em cada uma das camadas em cada solicitação.

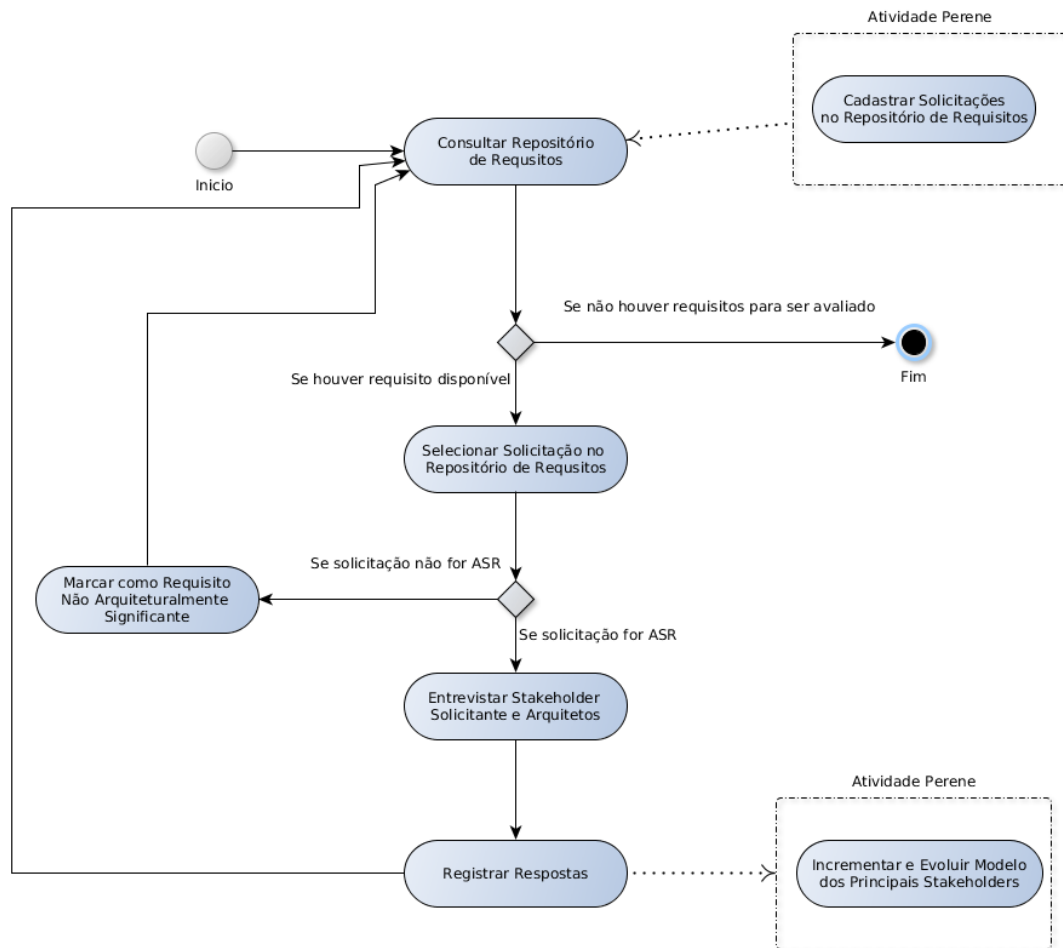


Figura 6.1 – Diagrama de Atividades para uma solicitação

Para responder as perguntas "quem" deve-se utilizar uma lista de perfis de *stakeholders* disponíveis, previamente definidos conforme o contexto do SoS em questão. Caso nenhum perfil disponível seja enquadrado, um novo perfil pode ser criado. A utilização de perfis de *stakeholders* tem o objetivo de diminuir a quantidade de *stakeholders* com a mesma atribuição no modelo. Dessa forma o modelo é incrementado a cada nova solicitação. Como sugestão de perfis de *stakeholders*, pode-se utilizar os perfis definidos na tabela de perfil de *stakeholders* no final do Capítulo 4.

Como utiliza-se abordagens de desenvolvimento de sistemas que são incrementais, o modelo também será incremental conforme as solicitações são recebidas e catalogadas. A evolução do modelo ocorre conforme o SoS aumenta sua abrangência de atuação. Por fim, pode-se representar várias visões do modelo, desde os *stakeholders* de cada módulo ou subsistema do sistema constituinte com suas relações até todos os *stakeholders* envolvidos em ASR no SoS. A representação do funcionamento de uma solicitação dentro do processo pode ser analisada no diagrama de atividades da Figura 6.1.

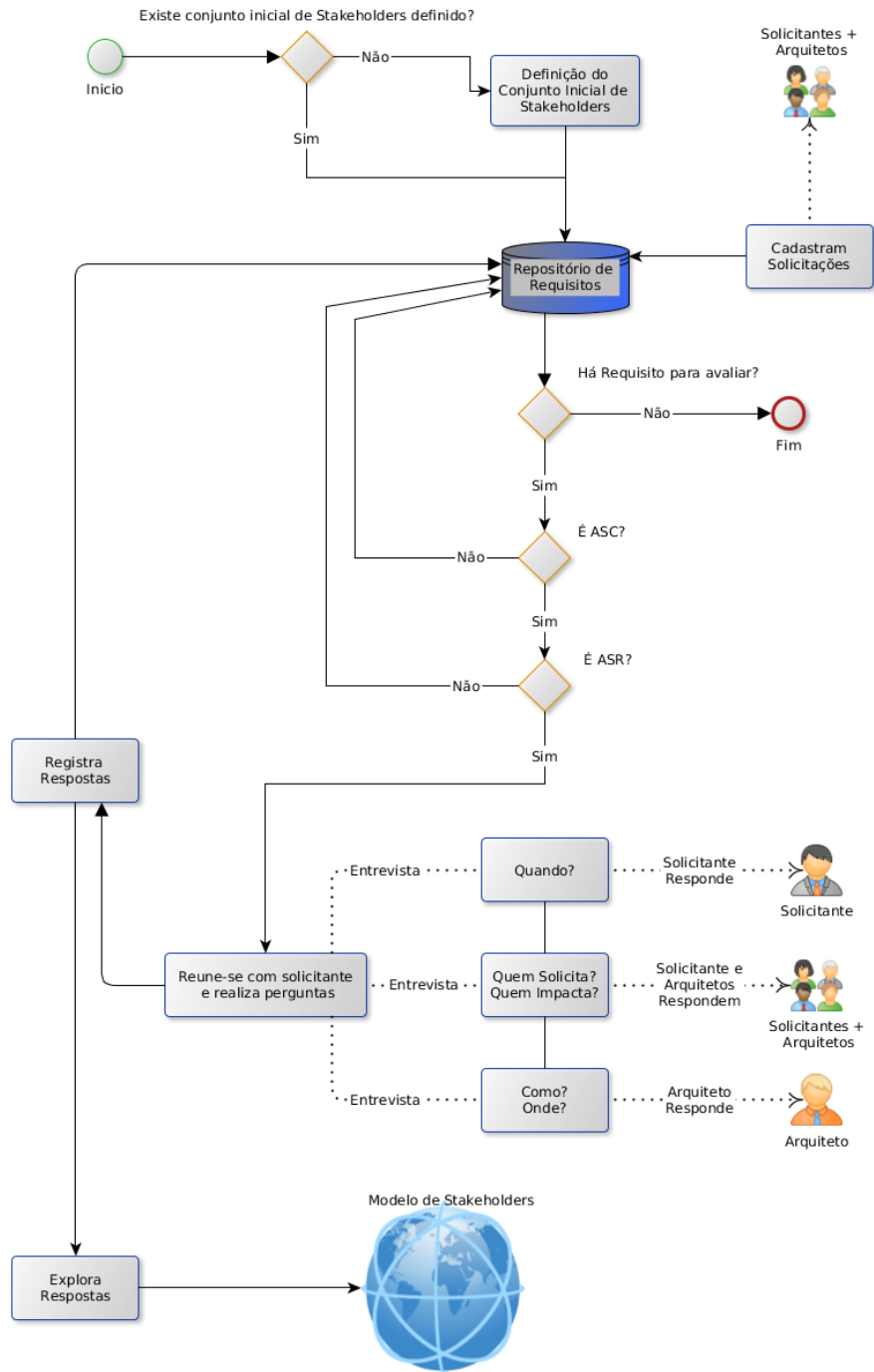


Figura 6.2 – Processo para construção do modelo

6.3 Utilização do Processo para criação do Modelo

Para construir um modelo de acordo com as definições, precisa-se seguir passo a passo o processo proposto, conforme Figura 6.2. Assim evita-se que haja distorções do entendimento dos processos para construção do modelo. Este processo funciona como um "algoritmo", onde uma sequência de etapas é executada para solucionar uma classe de problemas e está representado na Figura 6.2.

Inicia-se com a definição do conjunto inicial de *stakeholders*. Nesta etapa são selecionados os primeiros *stakeholders* que farão parte do modelo. Este conjunto define a estrutura base para o modelo. Após a definição dos primeiros *stakeholders*, tem início a atividade perene de cadastro de solicitações dos *stakeholders* interessados no SoS, essas ficam armazenadas num Repositório de Requisitos. O processo de evolução do modelo, tem início quando os arquitetos do SoS buscam solicitações neste repositório. Estas solicitações são expressas de forma textual passando o maior detalhamento possível para os arquitetos e identificando quem é o demandante. Quando a abordagem de desenvolvimento é *top-down*, o *stakeholder* solicitante cadastra sua necessidade, quando a abordagem for *middle-out* os arquitetos do SoS ou dos sistemas constituintes que fazem o cadastro no repositório de forma colaborativa.

Os arquitetos do SoS, conforme sua disponibilidade, avaliam as solicitações existentes no repositório de requisitos. Para cada solicitação, o arquiteto do SoS avalia se essa é um ASC. Se não for, a mesma é marcada como solicitação não impactante na arquitetura e conseqüentemente ignorada para o modelo resultante. Caso a solicitação seja um ASC, então o arquiteto precisa contactar o *stakeholder* solicitante para obter mais informações para verificar se o ASC torna-se um ASR respondendo a um conjunto de perguntas. No caso de um ASC não tornar-se um ASR, o mesmo é descartado da análise, porém se o ASC realmente for um ASR esta solicitação segue em avaliação.

Quando o *stakeholder* solicitante e o arquiteto se reúnem, esclarecem detalhes da solicitação e respondem as perguntas: "quando" a solicitação será implementada, a data limite para a implementação, "quem solicita" e "quem impacta" está solicitação, elencando os perfis de *stakeholders* solicitantes e impactantes em cada um dos níveis: SoS, sistemas constituintes e se necessário subsistemas dos sistemas constituintes. As perguntas "como" a solicitação será implementada e "onde" será implementada serão respondidas pelo próprio arquiteto do SoS durante ou após o encontro com o *stakeholder* solicitante.

As respostas de cada ASR são catalogadas pelo arquiteto do SoS em uma tabela que será utilizada como base para a evolução da estrutura do modelo. A partir dessa tabela o modelo é incrementado e a cada novo ASR analisado novas atualizações são geradas ao modelo. Caso o arquiteto entenda que o ASR analisado deve gerar outro requisito, o mesmo cadastra uma nova solicitação no Repositório de Requisitos para ser analisada em um novo ciclo do processo.

Quando um sistema constituinte deixar de fazer parte do SoS, automaticamente os *stakeholders* exclusivos destes sistemas serão removidos do modelo e os *stakeholders* que estão ligados a mais de um sistema constituinte tem seu vínculo atualizado no modelo. Esta remoção pode ser feita através de desativação ou remoção das referências deste sistema na tabela de base para o modelo.

6.4 Formatos de Representação do Modelo

Para representar o resultado do processo proposto nas seções anteriores, buscou-se na literatura opções que melhor o representariam. A representação através de modelos de arquitetura pode ser feita conforme a ISO 42.010 [34], utilizando a convenção de diagramas de classes *Unified Modeling Language* (UML). Conforme ISO 19.501 [32], o padrão UML tem o objetivo de prover à arquitetos de sistemas analisar objetos em uma forma consistente para especificar, visualizar, construir e documentar os artefatos de sistemas, bem como para modelagem de negócios.

Outra possibilidade é a representação através de *Systemigram* criado por Boardman e Sauser [7]. *Systemigram* é uma palavra amálgama composta por sistema e diagrama. Esta representação gráfica é utilizada para representar relacionamentos de comunicação. Esta ferramenta normalmente é utilizada como uma ferramenta de comunicação, oferecendo uma forma para representar graficamente um único sistema complexo, problema ou evento descrito [7]. Porém também podem ser usados para a representação visual da arquitetura geral de um sistema.

Com ambas abordagens consegue-se representar a relação dos diferentes *stakeholders* com os sistemas constituintes e com o SoS. Mas acredita-se que com a *Systemigram* consegue-se obter uma representação visualmente melhor e mais atual. Assim, optou-se por utilizar o *Systemigram* para representar estas relações entre os sistemas e os *stakeholders*, exibindo de forma gráfica os essas relações.

De acordo com Boardman e Sauser [8], *Systemigram* são compostos de pelo menos duas partes e uma relação entre elas. Estas partes, são conhecidas por nodos em formato de elipses e a relação é o *link* entre esses nodos. Os rótulos dos nodos são os substantivos da frase e o rótulo no *link* é o verbo. Aos nodos e *links* podem ser adicionadas fronteiras para simplificar o entendimento.

No caso do modelo de *stakeholders* proposto, nos nodos estarão os *stakeholders* e os sistemas e no *link* não há necessidade de rótulos. A construção visual do modelo deve seguir as instruções abaixo. O uso destas instruções é ilustrado no exemplo apresentado na Figura 6.3

- O SoS será representado por um nodo. Aconselha-se deixar este nodo em evidência com uma cor que possa diferencia-lo dos demais nodos. No exemplo da Figura 6.3, o SoS está no canto superior direito na elipse de cor verde.
- Os *stakeholders* exclusivos do SoS, ou seja, os *stakeholders* que não possuem nenhuma ligação direta com os sistema constituinte, serão ligados diretamente ao SoS. No exemplo da Figura 6.3, tem-se o Stakeholder4.
- Os *stakeholders* que estão relacionados com sistemas constituintes específicos devem estar dentro de um nodo que representa o sistema constituinte em questão. Este nodo delimita as fronteiras do sistema constituinte. No exemplo da Figura 6.3, tem-se os Stakeholders2, Stakeholders3 e Stakeholders5.

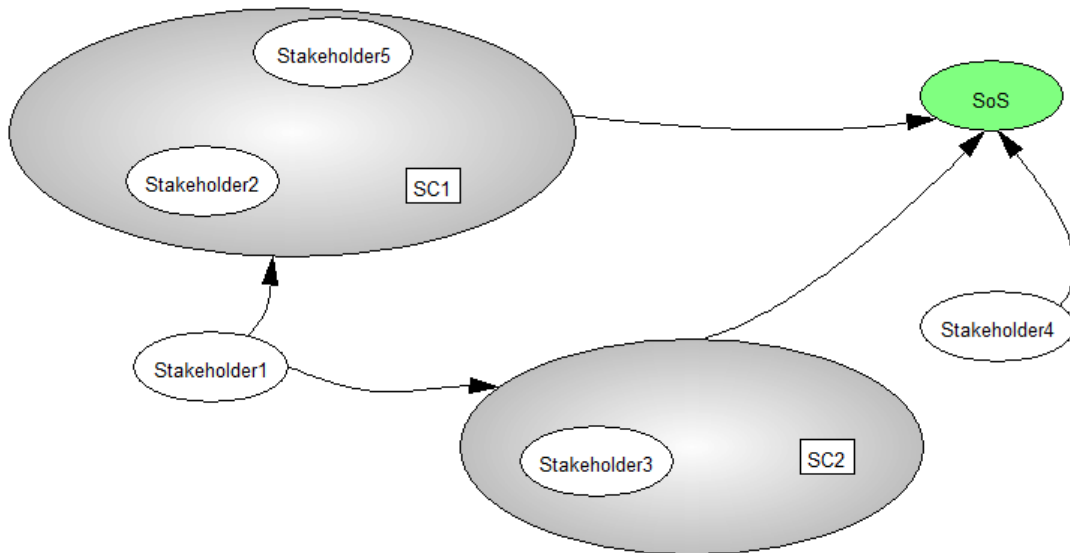


Figura 6.3 – Exemplo de representação gráfica do modelo de *stakeholders*

- Os *stakeholders* que estão relacionados com mais de um sistema constituinte podem ficar fora do nodo do sistema constituinte. No exemplo da Figura 6.3, tem-se o Stakeholder1.

Assim, *stakeholders* podem estar representados nestas três formas: conectados diretamente ao SoS, conectados a apenas um sistema constituintes ou conectados a dois ou mais sistemas constituintes. Pode-se dizer que os *stakeholders* que aparecem no modelo são os principais *stakeholders* de SoS *Acknowledged*.

7. AVALIAÇÃO DO PROCESSO E DO MODELO PROPOSTOS

A avaliação das propostas de processo e modelo para identificação dos principais *stakeholders* em SoS *Acknowledged* está detalhada neste capítulo, organizada em cinco seções. Na Seção 7.1, apresenta-se o planejamento da avaliação. Na Seção 7.2, tem-se a explicação da execução da avaliação planejada. Na Seção 7.3, tem-se a caracterização dos entrevistados. Na Seção 7.4, apresenta-se os resultados das entrevistas. E, por fim, na Seção 7.5, tem-se os comentários e observações gerais dos entrevistados.

7.1 Planejamento

Como trata-se de uma pesquisa qualitativa, para realizar a avaliação do processo e do modelo propostos optou-se por utilizar entrevistas estruturadas. Estas foram conduzidas com especialistas e arquitetos em SoS.

O planejamento para a execução da entrevista foi organizado em 2 etapas: contextualização e questionário. A etapa de contextualização tem por objetivo apresentar conceitos utilizados na pesquisa, objetivos e a questão de pesquisa. Na segunda etapa, do questionário, é apresentado um conjunto de perguntas estruturadas e ordenadas para o respondente com o objetivo de coletar informações sobre o entendimento e aplicabilidade do que está sendo proposto. Este questionário foi escrito em língua portuguesa e antes de ser apresentado aos entrevistados, passou por revisões e por três entrevistas piloto. Dessas três entrevistas piloto, a primeira foi com testador de software com experiência em engenharia de requisitos de software e as últimas duas foram com especialistas em SoS e engenharia de requisitos de software. Na Tabela 7.1 é apresentado na íntegra o perfil desses entrevistados nas entrevistas piloto. O questionário utilizado nas entrevistas finais é apresentado na íntegra no Anexo B.

Tabela 7.1 – Perfil dos Entrevistados na entrevista piloto

Idade	Gênero	Escolaridade	Tempo em anos de Experiência em SoS	Função/Cargo Atual	Tempo em anos de Experiência no Cargo Atual
28	Masculino	Pós-Graduação incompleto	0	Testador de Software	5
30	Masculino	Pós-graduação	5	Professor	5
39	Feminino	Pós-graduação incompleto	3	Estudante	5

7.2 Execução

A execução das entrevistas começou através de divulgação, via e-mail à contatos que trabalham com SoS da pesquisa que estava sendo conduzida e da necessidade de avaliação do que está sendo proposto. Para cada interessado em participar do processo foi acordado um horário, conforme a disponibilidade do entrevistado, para que a entrevista ocorresse através de uma chamada de voz. No horário marcado, ao iniciar a ligação, era explicado ao entrevistado como a entrevista estava organizada e como seria a condução da mesma. Também foram informados dos tempos médios estimados para cada etapa da entrevista. Em tempo de piloto, estipulou-se que a etapa de contextualização demoraria entre 5 e 7 minutos e a etapa de questionário cerca de 25 minutos, essa etapa podendo variar conforme cada entrevistado. Também foi solicitado ao entrevistado se o mesmo autorizava a gravação de áudio da entrevista a fim de garantir que nenhuma informação coletada seria perdida ou distorcida.

Ao iniciar a entrevista, com a etapa de contextualização, seguia-se sempre o mesmo roteiro. Inicialmente explicava-se o conceito de SoS do DoD [27], apresentava-se as cinco características de Maier [42] e as duas de Sommerville [63] para então apresentar-se as quatro categorias de SoS com exemplos práticos de cada uma delas. Neste momento, era explicado que a pesquisa atual teria foco em apenas uma das categorias de SoS: *Acknowledged*. Após esses conceitos, apresentava-se como chegou-se ao *gap* da literatura:

Estas características dos SoS dificultam a execução de processos da Engenharia de Software existentes.

Diversos pesquisadores como Lim, Ncube, [41], Ballejos e Montagna [3] afirmam que: bons métodos de identificação de *stakeholders* conhecidos podem funcionar em projetos pequenos ou monolíticos mas ficam aquém dos projetos de SoS e em projetos interorganizacionais devido a incapacidade de atender a complexidade e ao grande número de *stakeholders* envolvidos.

Ainda, conforme Nakagawa [47], para o sucesso de um SoS é fundamental dar atenção a arquitetura de software. Para a elaboração de uma arquitetura de software consistente e objetiva precisa-se conhecer os requisitos arquiteturalmente significantes. Neste ponto, faz-se a relação entre a arquitetura de software de SoS com a ER ao participantes e conclui-se a relação entre os conceitos e o *gap* que este trabalho busca resolver. A etapa de contextualização era finalizada apresentando o objetivo e a questão de pesquisa.

Então o entrevistado era avisado do fim da etapa de contextualização e início da etapa de questionário. Neste momento o *link* para o questionário era enviado e o entrevistado era orientado a respondê-lo. Também explicava-se ao mesmo que a qualquer momento o entrevistado poderia interromper a execução do questionário para esclarecer todo o tipo de dúvida que surgisse. O questionário enviado está na íntegra no Anexo B.

Este questionário foi organizado em 5 partes. Na primeira há o termo de confidencialidade dos dados. Na segunda estão informações gerais da pesquisa, basicamente as informações explicadas na contextualização para caso o entrevistado queira visitar algum dos conceitos explicados previamente. Na terceira tem-se perguntas para caracterizar o perfil do respondente. Na quarta, as perguntas sobre a opinião do entrevistado quanto à necessidade de identificação de *stakeholders* em SoS. E, por fim, na quinta parte, estão questionamentos sobre o processo e o modelo propostos.

Todas as questões das partes 4 e 5 possuíam campos para justificar a resposta, ou seja, campos descritivos abertos a comentários e opiniões do entrevistado. Também 7 das 10 questões poderiam ser respondidas através de múltipla escolha usando a item da escala Likert com 5 pontos [40]. Quais sejam: Discordo Completamente, Discordo Parcialmente, Não Discordo nem Concordo, Concordo Parcialmente e Concordo Completamente. Assim com essa escala consegue-se conhecer o grau de conformidade do entrevistado com os itens solicitados nas questões.

Para finalizar a entrevista, era solicitado ao respondente se o mesmo desejava fazer algum comentário além dos que foram feitos até então por escrito ou falado. Depois disso, a entrevista era concluída e a gravação finalizada.

7.3 Caracterização dos Entrevistados

As entrevistas foram divulgadas via *e-mails* para grupos de especialistas de SoS do Brasil. É difícil precisar o tamanho da comunidade de SoS no Brasil, porém tem-se experiência de que é um grupo reduzido de pessoas. Dos contatos, obteve-se 8 entrevistas. Essa amostra da população participante foi dimensionada de acordo com a disponibilidade dos mesmos. O perfil de cada entrevistado está apresentado na Tabela 7.2. As entrevistas ocorreram em Dezembro de 2017.

A idade dos entrevistados varia entre 23 e 52 anos, porém a maior parte está próxima dos 30 anos de idade. Todos os entrevistados tem pós-graduação concluída ou em andamento, e possuem alguma experiência em SoS. Com relação as experiência em SoS dos entrevistados, tem-se entrevistados entre 1 anos e 5 anos de experiência. A Figura 7.1 apresentada a seguir apresenta os dados focados no tempo de experiência em SoS.

Com relação as funções exercidas atualmente pelos entrevistados, percebe-se que há muitas funções em comum. Também há entrevistados que tem mais do que uma função, por exemplo o participante 4 que acumula 6 funções. A Figura 7.2 apresentada a seguir apresenta os dados com ênfase nas funções atuais.

7.4 Resultados

Após a etapa de caracterização dos entrevistados, iniciou-se a etapa que teve por objetivo coletar a opinião do entrevistado sobre necessidade e importância da pesquisa. A primeira pergunta

Tabela 7.2 – Perfil dos Entrevistados

ID	Idade	Gênero	Escolaridade	Tempo em anos de Experiência em SoS	Função/Cargo Atual	Tempo em anos de Experiência no Cargo Atual
1	24	Masculino	Pós-graduação incompleto	1	Engenheiro de software	1
2	29	Masculino	Pós-Graduação	1	Estudante	7
3	52	Feminino	Pós-Graduação	5	Instrutor/Professor de TI	14
4	30	Masculino	Pós-Graduação	4	Engenheiro de software; Estudante; Instrutor/Professor de TI	7
5	30	Masculino	Pós-Graduação	1,5	Desenvolvedor de software; Engenheiro de software; Pesquisador Técnico	8
6	23	Masculino	Pós-graduação incompleto	2	Arquiteto de software	2
7	29	Feminino	Pós-Graduação	5	Instrutor/Professor de TI; Pós-doc em Engenharia de Software de SoS	1
8	30	Feminino	Pós-Graduação	5	Analista de sistemas; Arquiteto de software; Engenheiro de software; Estudante	9

fazia referência a importância da identificação de *stakeholders* em SoS: Pergunta 1 - "A identificação dos *stakeholders* de um SoS é importante?" 100% dos entrevistados concorda que é a identificação dos *stakeholders* é importante para um SoS. Desses, 37,5% concordam parcialmente e 62,5% concordam completamente. A Figura 7.3 apresenta o conjunto dessas respostas de forma gráfica.

Dos entrevistados que Concordam Completamente com a necessidade de identificação dos *stakeholders*, foram feitos os seguintes comentários:

"...como em qualquer sistema, saber quem se beneficiará ou lidará com o sistema, desenvolvimento, manutenção, etc. é de extrema importância para o sucesso do projeto" Participante 3.

"...considerando que a engenharia de um SoS (assim como de qualquer tipo de software) depende diretamente dos requisitos eliciados, identificar os stakeholders auxilia diretamente neste processo, uma vez que as fontes de requisitos ficam explícitas" Participante 4.

"...é necessário que os stakeholders que realmente estão "interessados" no cumprimento das missões do SoS como um todo sejam identificados. Um exemplo interessante está na extração

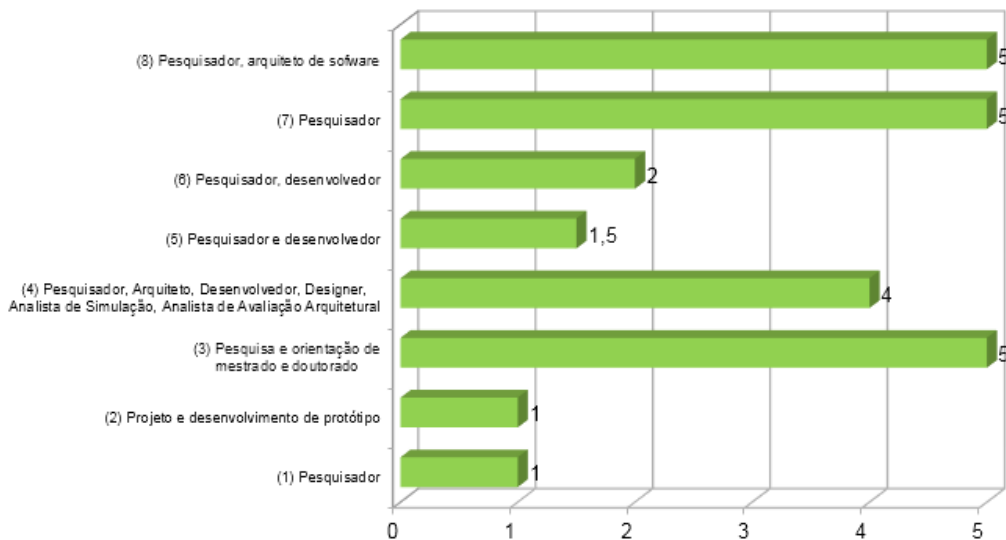


Figura 7.1 – Tempo de experiência dos entrevistados



Figura 7.2 – Cargos/Funções dos entrevistados

de conhecimento do SoS, cada constituinte possui sua própria base de dados, e ao unir toda elas, a intensidade de dados é imensa...Acredito que seria muito interessante se fossem identificados" Participante 6.

Os demais entrevistados também concordaram, mas trazem alguns pontos de atenção ao se fazer essa atividade.

"...para um SoS Acknowledged é importante, ...mas não deve ser para todos" Participante 2.

"...acredito sim ser muito importante, pois são as necessidades dos stakeholders frente aos seus sistemas (sistemas constituintes dentro do SoS) que irão montar/moldar o SoS como um todo. No entanto, não recebe nota máxima, pois a arquitetura do SoS é igualmente importante" Participante 5.

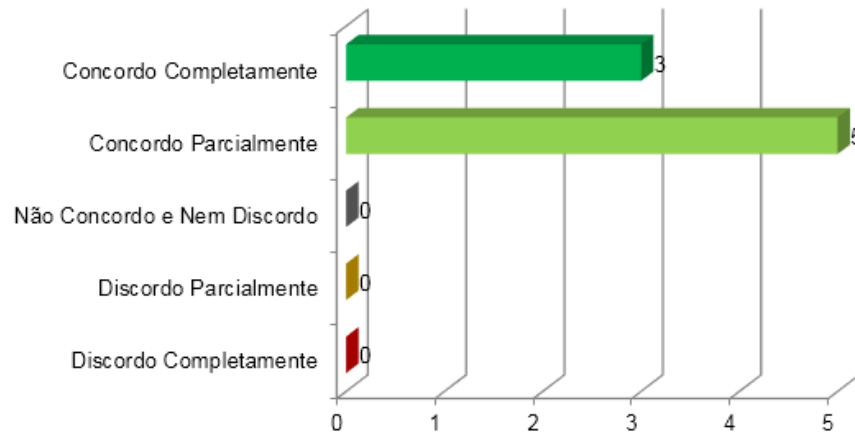


Figura 7.3 – Respostas da Pergunta 1 - A identificação dos *stakeholders* de um SoS é importante?

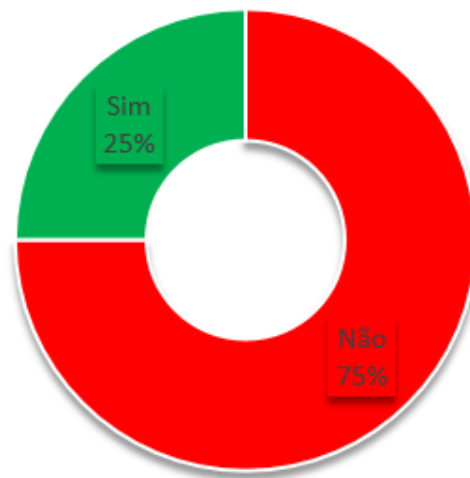


Figura 7.4 – Respostas da Pergunta 2 - Caso já tenha participado de um processo de identificação de *stakeholders* em SoS. Como esse processo ocorreu, quais técnicas foram utilizadas?

"...é importante e ao mesmo tempo muito difícil, uma vez que os constituintes do SoS podem mudar com o tempo e, conseqüentemente, o conjunto de interessados" Participante 7.

"...identificar e gerenciar os stakeholders é tão importante quanto documentar e interoperar sistemas constituintes únicos" Participante 1

"... é importante uma adequada caracterização dos stakeholders e das suas necessidades e responsabilidades" Participante 8.

Com o intuito de saber sobre as experiências dos entrevistados em processos de identificação de *stakeholders* em SoS, perguntou se já participaram diretamente de processo de identificação de *stakeholders* em SoS: Pergunta 2- "Caso já tenha participado de um processo de identificação de *stakeholders* em SoS. Como esse processo ocorreu, quais técnicas foram utilizadas?" Apenas 2 entrevistados participaram de processo de identificação de *stakeholders* em SoS, a Figura 7.4 apresenta o resultado sintético da pergunta.

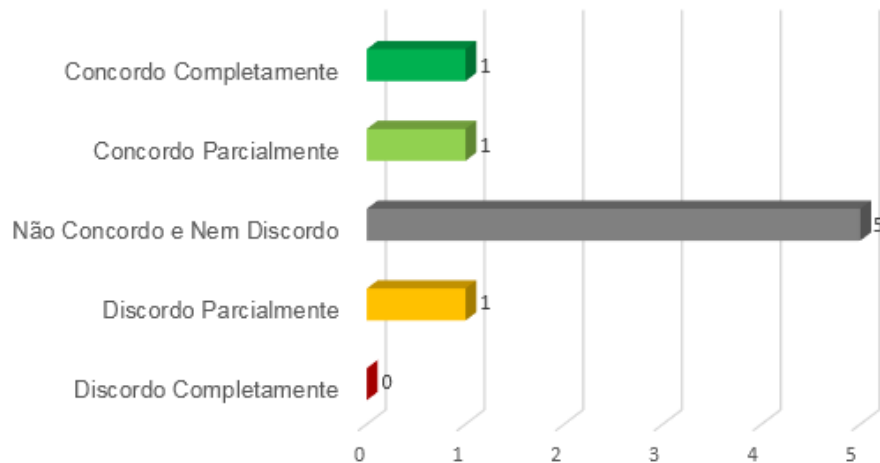


Figura 7.5 – Respostas da Pergunta 3 - SOMENTE as técnicas de identificação de requisitos existentes na engenharia de software tradicional, como as listadas pela ISO 1233, são suficientes para identificar os *stakeholders* de um SoS?

"...a identificação dos *stakeholders* ficou implícita no processo, pois o objetivo principal era identificar necessidade de sistemas de suporte a decisão. Empregamos entrevistas semi-estruturadas, grupos focais e questionários online" Participante 5.

"...a identificação dos *stakeholders* no domínio da saúde, tenho usado diretrizes, legislações, processos os quais definem os possíveis profissionais envolvidos ou responsáveis por sua correta execução. Também alguns *stakeholders* do domínio técnico envolvidos no ciclo de vida do SoS tem sido identificados, como por exemplo: engenheiro de requisitos, arquiteto de software, desenvolvedor, testador, engenheiro de interfaces, etc" Participante 8.

Antes da Pergunta 3 foram apresentadas as técnicas de identificação de requisitos em sistemas tradicionais de acordo com a ISO 1233 [31] de 1998. Então perguntou-se: Pergunta 3 - "SOMENTE as técnicas de identificação de requisitos existentes na engenharia de software tradicional, como as listadas pela ISO 1233, são suficientes para identificar os *stakeholders* de um SoS?" A maioria dos participantes, 62,5%, nem concordam e nem discordam dessa afirmação, enquanto que 12,5% discordam parcialmente, ou concordam parcialmente, ou ainda concordam completamente. Ainda, nenhum dos entrevistados discorda completamente. A Figura 7.5 apresenta os dados graficamente.

O participante que discorda parcialmente afirma que "*SoS são sistemas complexos e direcionados a problemas em larga-escala, principalmente no tipo Acknowledged que é fortemente ligado a sistemas de defesas. Existem muitos stakeholders e sistemas envolvidos, por isso esse processo precisa de uma atenção especial que o documento não aborda. Além da evolução de arquiteturas existentes*" Participante 1.

A grande maioria dos participantes não concorda e nem discorda e afirmam:

"...acredito que o foco em ter acordos de requisitos de integração dos constituintes e distribuição geográfica poderiam ter técnicas próprias, mas ao mesmo tempo parecem ser relacionados

com Observação de Padrões de Trabalho, Observação de Sistemas no Ambiente socio-organizacional e Engenharia Reversa” Participante 2.

“...elas são um bom começo, mas é preciso mais atenção para entender as colaborações entre os stakeholders e como isso impactar o SoS. Além disso, algumas vezes não é fácil identificar os constituintes, e sem isso não se sabe quem são os stakeholders envolvidos com cada constituinte” Participante 3.

“...fico um pouco neutro para responder esta pergunta, uma vez que o acervo de técnicas trazidos pela ISO é bem amplo...em se tratando de estruturas de SoS como Acknowledged, creio que muitas vezes o SoS é intencionalmente construído, de modo que os stakeholders estarão diretamente envolvidos no processo, e a sua "identificação" talvez seja mais associada à identificação dos constituintes necessários para oferecer uma ou mais funcionalidades do SoS. Assim, os stakeholders apareceriam associados aos constituintes que eles mantêm” Participante 4.

“...acredito que outras alternativas precisariam ser consideradas para um SoS de grande porte. Dependendo do escopo e da área de aplicação desse sistema, muito provavelmente vai ser necessário considerar stakeholders do governo (federal, estadual, municipal), algo que não foi necessariamente considerado pelos sistemas constituintes” Participante 7.

“...considero que a identificação dos stakeholders em SoS pode reusar algumas dessas técnicas” Participante 8.

O participante que concorda parcialmente afirma que *“...a ISO 1233 possui uma ampla diversidade de técnicas que podem servir bem para a identificação dos requisitos. Entretanto, seria interessante uma adaptação delas em nível de SoS, como por exemplo adaptar as estruturas para conversar com os stakeholders de cada um dos constituintes para unir as informações” Participante 6.*

E o participante que concorda completamente afirma que *“... não vejo uma forma adicional de identificarmos stakeholders. Talvez, uma alternativa fosse derivar os stakeholders a partir de uma arquitetura de SoS” Participante 5.*

Antes de fazer a próxima pergunta, apresentou-se o conceito de Processo conforme definido na ISO 15288 [35] de 2015. Então, perguntou-se sobre a importância de ter um processo realizar a identificação dos stakeholders de SoS: Pergunta 4- *“É importante se ter um processo para realizar a identificação dos stakeholders de SoS?”* 50% dos entrevistados concorda completamente que é importante ter um processo para identificação dos stakeholders de SoS, 25% nem concordam e nem discordam, 12,5% ou concordam parcialmente ou discordam parcialmente e nenhum entrevistado discorda completamente sobre a importância de se ter um processo para isso. A Figura 7.6 apresenta graficamente esses dados.

O Participante 4 que discorda parcialmente disse que não entende completamente a necessidade de um processo para isso.

O Participante 1 e 8 nem concordam e nem discordam. Conforme o Participante 8, *“nunca utilizei um processo para definição de stakeholders, então é difícil para eu mesurar sua importância”.*

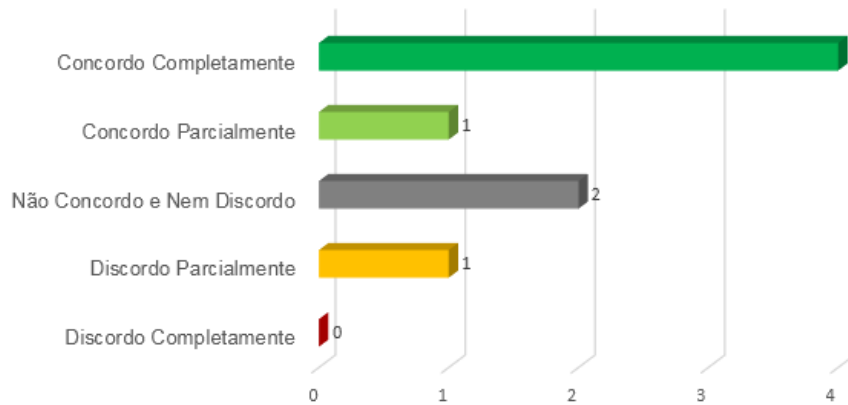


Figura 7.6 – Respostas da Pergunta 4 - É importante se ter um processo para realizar a identificação dos *stakeholders* de SoS?

O Participante 2 concorda parcialmente pois ele *"acredita que os processos definirão os papéis dos stakeholders"*.

Os participantes que concordam completamente, afirmam que:

"...ter um processo ajudaria na execução da tarefa de identificação de stakeholders" Participante 3.

"...saber quem são os stakeholders e suas necessidades é a base para o desenvolvimento de um sistema de qualidade" Participante 5.

"...a identificação não pode ser realizada de forma ad-hoc e sem uma avaliação profunda..." Participante 6.

"...ajuda muito ter uma direção sobre o que precisa ser feito para saber o que fazer e em que ponto estamos de uma determinada atividade" Participante 7.

A partir do próximo questionamento, inicia-se a avaliação da percepção do entrevistado com relação ao processo e modelo de identificação dos principais *stakeholders* de SoS *Acknowledged* propostos. Assim, foi apresentado o processo ao entrevistado no formato de diagrama de atividades, depois num formato explicativo descritivo e no final o *design* de cada uma das etapas do processo, conforme pode ser visto no apêndice . Para a avaliação do processo, fez-se 3 perguntas. A primeira pergunta é sobre viabilidade de construir um Repositório de Requisitos no contexto de SoS *Acknowledged*: Pergunta 5- "O Repositório de Requisitos, sendo construído de forma colaborativa, constitui em uma atividade viável de ser realizada no contexto de SoS *Acknowledged*?" Os entrevistados na sua grande maioria, 62,5%, concordam completamente que sim, enquanto que 12,5% ou concordam parcialmente ou nem concordam e nem discordam ou discordam parcialmente. A Figura 7.7 apresenta esses resultados.

Analisando as respostas dos entrevistados começando pelo Participante 8 que discorda parcialmente pois *"depende do estado do SoS. Acredito que a execução de essa atividade antes da existência de um SoS, não é viável"*. De acordo com este participante, se o SoS estiver em fase de negociação dos primeiros requisitos, essa atividade não será bem desenvolvida. O Participante 3 que nem concorda e nem discorda ficou em dúvida quanto a ordem do processo, comentou que *"se*

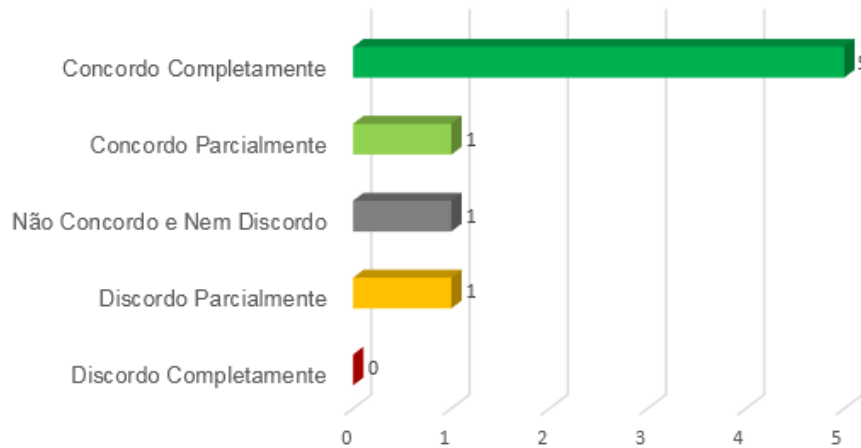


Figura 7.7 – Respostas da Pergunta 5 - O Repositório de Requisitos, sendo construído de forma colaborativa, constitui em uma atividade viável de ser realizada no contexto de SoS *Acknowledged*?

eu tenho interesse em registrar um requisito e ainda não existe o SoS. Quem olharia ou monitoraria o repositório para atender minha solicitação, se ainda não existe um SoS?”. Neste caso, percebe-se que ambos tem dúvidas do funcionamento quando um SoS ainda não existe. Como resposta para essa dúvida, quando um SoS ainda não existe, deve-se utilizar o processo inicial para levantamento dos principais *stakeholders* conforme está no Capítulo 7.

O Participante 5 concorda parcialmente ficando receoso quanto ao *“...empenho e dedicação dos stakeholders em contribuir. Talvez, isso desse ser garantido antes”* de alguma forma. Os demais participantes concordam completamente com o Repositório de Requisitos e afirmam:

“...parece ser viável porque um sistema de ticket é viável”. Porém fica com dúvida quanto a carga de trabalho sobre o arquiteto, *“só achei que dependendo do projeto pode sobrecarregar colocar tudo na mão do arquiteto para avaliar”* Participante 2.

“...após entender que esta proposta tem uma vertente mais social, percebo a importância e o quão interessante é identificar stakeholders em SoS. Trata-se de um esforço que pode ser aproveitado em processos de aquisição de constituintes, governança, e durante engenharia de requisitos e design arquitetural. É uma atividade bem viável e poderia até mesmo dar origem a uma nova forma de desenvolver software para SoS, inerentemente colaborativa e até mesmo próxima a um crowdsupporting” Participante 4.

“...faz o total sentido, visto que o sistema cresce e se adapta pela colaboração dos constituintes em busca de alcançar as missões do SoS” Participante 6.

“...registrar eletronicamente os requisitos em um formato que facilite o processamento, manutenção e recuperação é essencial para manter um documento “vivo”na organização” Participante 7.

Agora pensando na seleção de ASCs e ASRs, perguntou-se sobre a viabilidade da execução ser realizada pelo arquiteto do SoS: Pergunta 6- *“O arquiteto do SoS ser o responsável pela seleção de ASC e ASR, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?”*

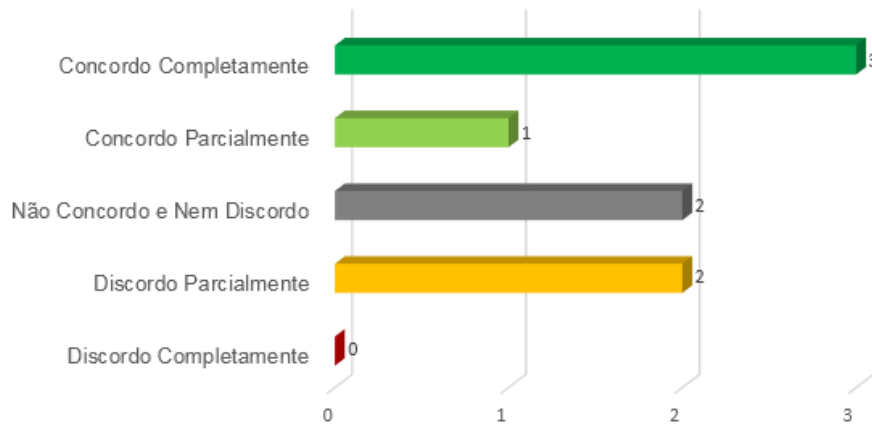


Figura 7.8 – Respostas da Pergunta 6 - O arquiteto do SoS ser o responsável pela seleção de ASC e ASR, constitui em uma atividade viável de ser realizada no contexto de SoS *Acknowledged*?

Esta pergunta dividiu bastantes as opiniões dos entrevistados, conforme a Figura 7.8, onde a única opção não escolhida foi a discordo completamente. Ainda sim a maioria dos entrevistados, 37,5%, concorda completamente com isso, enquanto que 12,5% concordam parcialmente indicando que metade dos entrevistados concorda de alguma forma com o arquiteto fazendo este trabalho. No entanto, 25% ou nem concordam e nem discordam ou discordam parcialmente.

Os participantes que discordam parcialmente que o arquiteto do SoS deva ser o responsável pela seleção de ASC e ASR comentaram:

"...o arquiteto pode não saber tudo o que é necessário, ele pode se focar em viabilidade técnica e esquecer de necessidades reais" Participante 2.

"...sim, se o foco for a arquitetura, mas acredito que esse não é o caso" Participante 8.

Os participantes que nem concordam e nem discordam questionam se apenas o arquiteto deve ser considerado:

"...um arquiteto de SoS é o suficiente ou devem ser considerados outros atores?" Participante 1.

"...parece inviável ter alguém só pra fazer isso..." Participante 3.

O participante que concorda parcialmente sugere que alguns *stakeholders* também ajudem no processo:

"... talvez seria interessante que durante a criação do modelo, os stakeholders que estão diretamente conectados ao SoS possam também auxiliar nessa tomada de decisão, visto que podem analisar a proposta em perspectivas diferentes" Participante 6.

Os participantes que concordam completamente reforçam a ideia de o arquiteto de SoS ser o responsável pela visão abrangente que esse profissional possui.

"...arquiteto juntamente com outros envolvidos no processo podem trabalhar em conjunto para definir os ASR" Participante 4

"...esse arquiteto é o que conhece mais do SoS..." Participante 5.

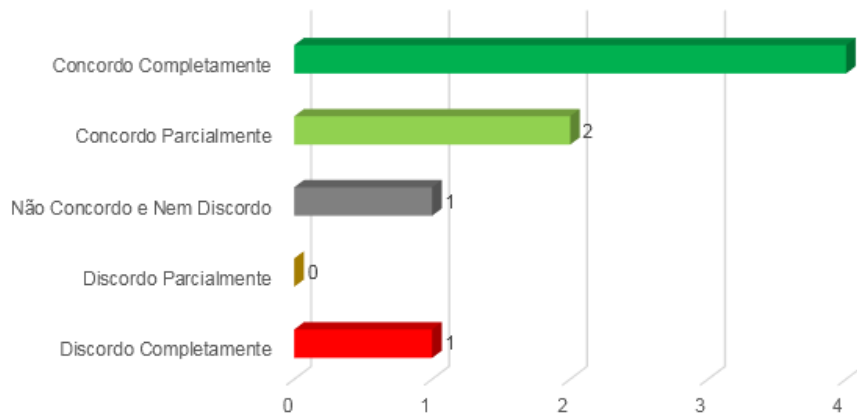


Figura 7.9 – Respostas da Pergunta 7 - O arquiteto do SoS ser o responsável pela entrevista com os *stakeholders* que definem ASRs, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?

"... logo é viável que ele seja o escolhido para centralizar essa atividade" Participante 7.

A respeito da viabilidade das entrevistas serem realizadas pelos arquitetos de SoS juntamente com os *stakeholders* que definem um ASR, perguntou-se: Pergunta 7- *"O arquiteto do SoS ser o responsável pela entrevista com os stakeholders que definem ASRs, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?"* Grande parte dos entrevistados concorda que é viável, 50% concordam completamente e 25% concordam parcialmente. E 12,5% ou nem concordam e nem discordam ou discordam completamente. A Figura 7.9 apresenta este resultado.

O participante que discorda completamente entende que *"... o arquiteto do SoS é quem define os ASRs baseado nos requisitos dos outros stakeholders."* Participante 8. Parece que houve um problema de entendimento com relação a pergunta

O participante que nem concorda e nem discorda afirma que *"...realizar entrevistas e arquitetar um sistema requerem habilidades e experiências diferentes...acho que deveria haver um analista de requisitos"* Participante 5.

Os participantes que concordam parcialmente comentaram:

"...arquiteto deveria saber projetar como fazer o que é pedido" Participante 2.

"...talvez o próprio repositório poderia automatizar parte dessa atividade, uma vez que o tempo do arquiteto é limitado e o número de interessados pode ser muito alto" Participante 7.

Dos participantes que concordam completamente, 2 concordam sem questionamentos e outros 2 reforçaram:

"...pensando que já temos um arquiteto, essa seria uma de suas funções principais" Participante 3

"...o arquiteto tem o conhecimento suficiente para realizar essa entrevista e tomar as decisões corretas" Participante 6

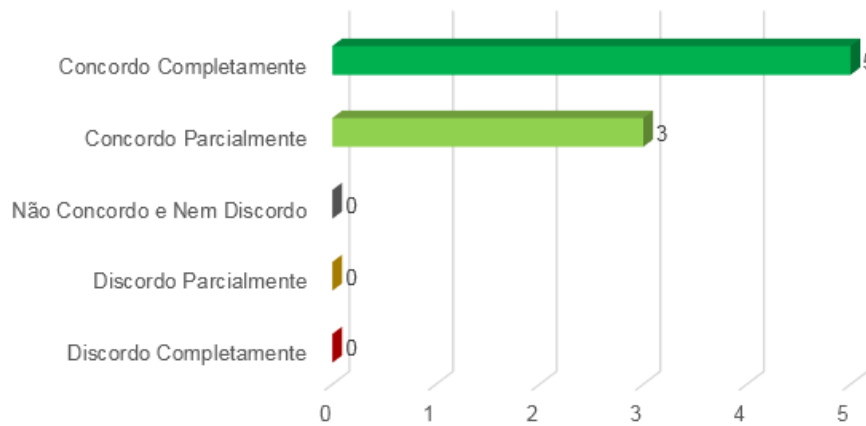


Figura 7.10 – Respostas da Pergunta 8 - O resultado do processo é um modelo. Este modelo que representa o resultado da execução do processo é importante?

A próxima pergunta faz referência ao modelo resultante do processo. Antes de fazê-la foi apresentado o conceito de modelo, segundo a ISO 1233 [31] e apresentado o modelo para identificação dos principais *stakeholders* de SoS *Acknowledged* de forma textual e gráfica. Sobre este modelo proposto foi perguntado: Pergunta 8- "O resultado do processo é um modelo. Este modelo que representa o resultado da execução do processo é importante?" Todos os participantes concordam completamente ou parcialmente com a importância do modelo. Conforme a Figura 7.10, 62,5% dos respondentes concordam completamente e 37,5% dos respondentes concordam parcialmente.

Dos que concordam parcialmente, um entende que o modelo poderia também mostrar os impactos sobre os sistemas constituintes (Participante 2) e outro entende que poderia usar para analisar e priorizar ASR (Participante 8).

"...acho ruim a seta ser a mesma entre o Sistema Constituinte e o SOS e a seta do stakeholder e o Sistema Constituinte. Parece bom para reconhecer responsabilidade mas acho que o mais interessante seria uma forma para mostrar impactos em possíveis modificações de constituintes" Participante 2

"...é interessante para analisar ou priorizar ASRs durante a construção do SoS" Participante 4.

"...sim. Com ele, o arquiteto conseguiria entender quem deve ser ouvido, avaliado, quando uma nova funcionalidade chega; ou, até mesmo, se um requisito conflitante chega, quais sistemas são impactados e com quem devemos conversar" Participante 5

Por sua vez os que concordam completamente ou não fizeram comentários ou deram fins distintos ao modelo, como:

"...acho importante ter um modelo como forma de documentação do resultado" Participante 3.

"...de posse deste modelo, o arquiteto sabe com quem deve se comunicar à medida que conduzir o processo de design" Participante 4

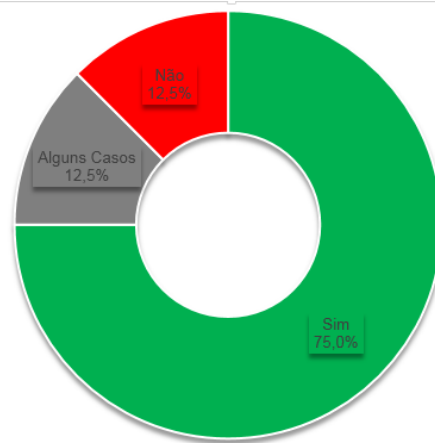


Figura 7.11 – Respostas da Pergunta 9 - Você acredita que este processo e o modelo resultante funcionariam num SoS real?

"... é importante pois gera uma visualização rápida de quem são os envolvidos quando decisões precisam ser tomadas" Participante 6.

Após, avaliar as partes do processo e o modelo separadamente, perguntou-se a opinião do entrevistado quando ao funcionamento do processo e modelo juntos, o modelo sendo o resultado do processo, na prática em um SoS real. Perguntou-se: Pergunta 9- *"Você acredita que este processo e o modelo resultante funcionariam num SoS real?"* A grande maioria dos entrevistados acredita que o processo pode funcionar. Assim, tentando agrupar as respostas para gerar um racional, foram criadas três categorias em tempo de análise das respostas e exibidas na Figura 7.11.

"...o processo é promissor em SoS gerenciados por uma autoridade central" Participante 1 como SoS Acknowledged.

"...teoricamente alguns SoS não precisam ser gigantescos. A proposta deve funcionar em alguns casos mas pode não escalar. Talvez o mais importante seria mostrar quando ela funciona e não se ela funciona" Participante 2.

"...sim, mas é claro que precisaria validar com um estudo concreto" Participante 3.

"...completamente. Na verdade, ao compreender a abordagem, fiquei bem empolgado com a ideia de usar o modelo" Participante 4.

"...fico apenas em dúvida quanto ao empenho dos stakeholders de sub-sistemas, e.g., "por que eu não mexo logo aqui (no constituinte)?" ou "não lembrei de falar sobre essa alteração". De alguma forma, eles devem entender que os constituintes fazem parte de algo maior. Contudo, acredito que funcionaria sim!" Participante 5.

"...totalmente, o processo é dinâmico e possui uma arquitetura bem estabelecida, definindo a hierarquia correta das tomadas de decisões, registrando as etapas de forma padronizada e se adaptando ao contexto. Não vejo motivos para não funcionar" Participante 6

"...se tiver contar com apoio ferramental, acredito que sim" Participante 7.

"...considero que falta esclarecer alguns conceitos em relação ao papel do arquiteto do SoS. Baseado no fato de que o processo está somente focado nos ASRs do SoS, acredito que vai ser

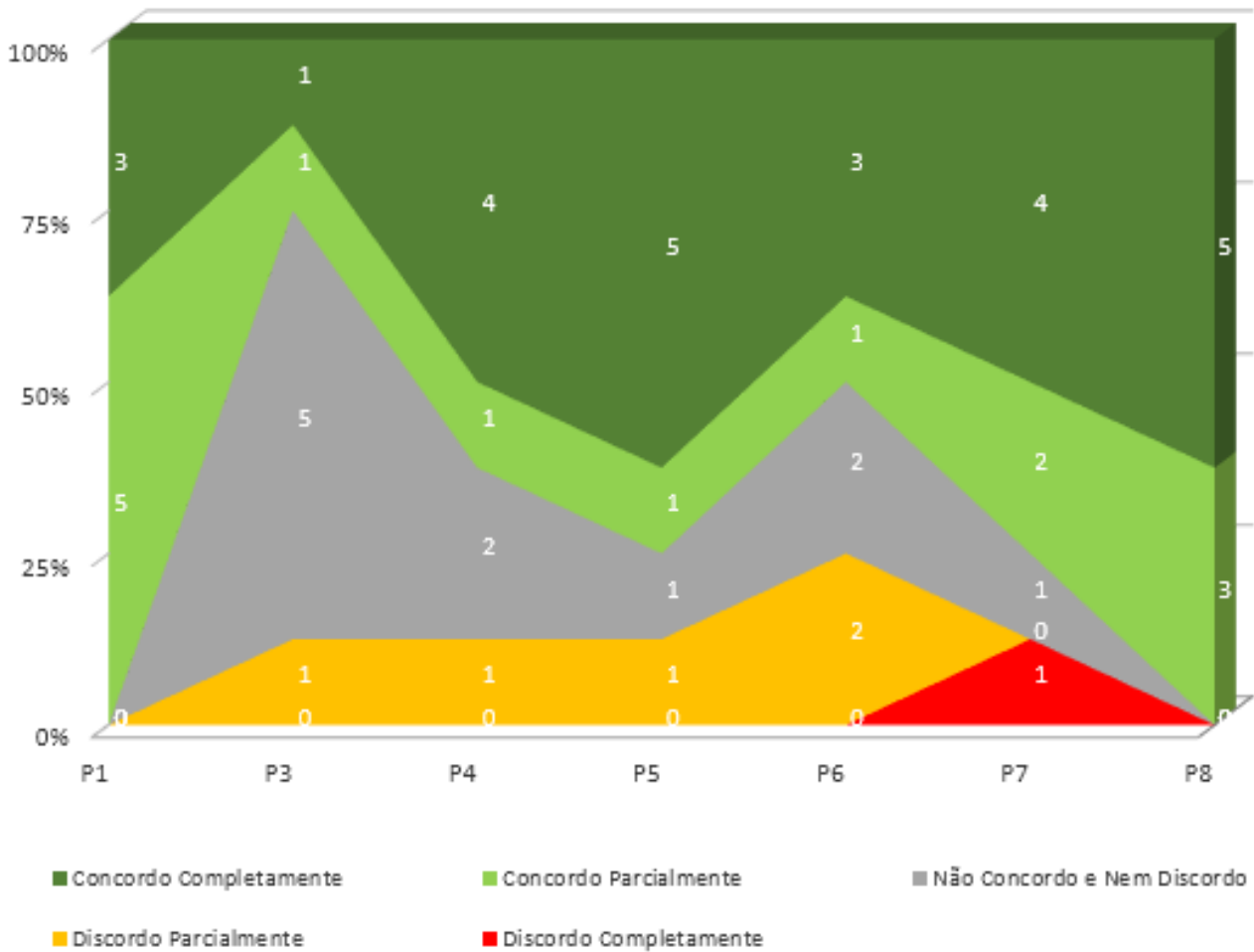


Figura 7.12 – Consolidação de respostas em escala Likert

inviável ser utilizado num projeto real, devido que stakeholders envolvidos com requisitos funcionais, ou interessados no cumprimento dos objetivos do SoS, estão sendo ignorados” Participante 8.

7.5 Comentários e Observações Gerais

Para as perguntas estruturadas em escala Likert, pode-se agrupar as respostas para possibilitar análise conjunta de todos os respondentes para as diversas respostas. Na Figura 7.12 a seguir, cada pergunta foi representada com uma sigla que é apresentada na sequência.

- **P1** = A identificação dos *stakeholders* de um SoS é importante?
- **P3** = SOMENTE as técnicas de identificação de requisitos existentes na engenharia de software tradicional, como as listadas pela ISO 1233, são suficientes para identificar os *stakeholders* de um SoS?
- **P4** = É importante se ter um processo para realizar a identificação dos *stakeholders* de SoS?

- **P5** = O Repositório de Requisitos, sendo construído de forma colaborativa, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?
- **P6** = O arquiteto do SoS ser o responsável pela seleção de ASC e ASR, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?
- **P7** = O arquiteto do SoS ser o responsável pela entrevista com os stakeholders que definem ASRs, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?
- **P8** = O resultado do processo é um modelo. Este modelo que representa o resultado da execução do processo é importante?

Ao analisar as respostas, percebe-se que a grande maioria dos participantes entende que é importante existir um processo para a identificação dos principais *stakeholders* de SoS *Acknowledged* (P1). Também, há uma grande incerteza quanto a aplicação de somente as técnicas da Engenharia de Software existente para identificação de *stakeholders* em SoS (P3). A grande maioria dos entrevistados concorda que o arquiteto de SoS tem que ser o responsável por entrevistar os *stakeholders* do SoS (P7). E que há uma certeza entre todos os entrevistados, os quais concordam de alguma forma que o modelo proposto representa o resultado da execução do processo (P8).

No final da entrevista havia uma pergunta sobre sugestões de melhoria para o processo ou modelo. O objetivo dessa pergunta era, além de coletar sugestões dos participantes, esclarecer alguma necessidade ao trabalho que o participante não conseguiu expressar nas perguntas anteriores.

O Participante 1 ficou com uma dúvida sobre os envolvidos: *"...foi considerado a participação de mais atores além do arquiteto de SoS?"*. Este participante questionou se não seria interessante ter mais um perfil para compartilhar a responsabilidade com o arquiteto de SoS.

O Participante 2 fez sugestões ao modelo: *"...diferenciar setas e representação de elipses. Muitos não gostam de elipses porque não encaixam bem em papel. Eu recomendaria algo mais estilo modelo de goals que mostrem o ASR/ASC/Requisito com arestas para os stakeholders e arestas para os Sistemas Constituintes mostrando se elas auxiliam/contribuem ou atrapalham. Assim pode até mostrar para os stakeholders o que atrapalharia implementar algum pedido ruim"*.

O Participante 3 sugere que o repositório seja algo compartilhado entre SoSs: *"...considerar que o repositório serviria à vários SoS diferentes e ver o que isso impacta no processo"*.

O Participante 4 sugere atualização de alguns conceitos, como *"...atualmente, não se associa mais distribuição geográfica a SoS, justamente porque os constituintes podem estar em uma mesma sala. Tem-se utilizado a terminologia "distribuição" apenas". Também sugere para tomar cuidado com exemplos de SoS, como "...sistemas bancários não podem ser considerados SoS porque seus constituintes não possuem independência gerencial..."*.

O Participante 5 sugere inclusão de novos atores/papeis ao processo, com a inclusão de *"...atividade para análise dos stakeholder (comprometimento, características, etc) e uma analista de requisitos de SoS, diferente do analista de constituinte"*.

O Participante 6 sugere que os principais *stakeholders* sejam envolvidos na tomada de decisão do arquiteto: *"...talvez, conforme o modelo de stakeholders fosse gerado, eles também auxiliassem nas tomadas de decisões, que as vezes poderiam ser impactadas pelo conhecimento apenas do arquiteto"*.

O Participante 7 reforça as sugestões que fez ao longo da entrevista como: *"...o repositório pode automatizar parte dessa atividade"(definição dos ASRs)", uma vez que o tempo do arquiteto é limitado e o número de interessados pode ser muito alto"*.

O Participante 8 salienta quem precisa realmente ser envolvido em sua opinião: *"...considerar os stakeholders interessados nos objetivos ou missões do SoS, que no final, para isso que o SoS é construído"*.

As sugestões dos entrevistados e comentários durante a entrevista foram avaliados. Algumas dessas sugestões, entende-se que seriam boas melhorias ao trabalho, como a inclusão de novos participantes para dividir a responsabilidade com o arquiteto de SoS sugerida pelos Participantes 1 e 5 e tentar envolver os principais *stakeholders* identificados em determinadas tomadas de decisões que o arquiteto do SoS precisa fazer sugerida pelo Participante 6. Estas sugestões são excelentes contribuições que podem ser tratadas num trabalho futuro. As considerações feitas pelo Participante 4 como atualização de conceitos e cuidados com exemplos de SoS citados foram corrigidos no trabalho.

8. CONSIDERAÇÕES FINAIS

Este capítulo apresenta as resoluções finais e conclusões do estudo. Na Seção 8.1, apresenta-se a síntese do trabalho e na Seção 8.2, apresenta-se os possíveis trabalhos futuros que podem, surgir a partir desse trabalho.

8.1 Conclusões

A presente pesquisa apresenta um estudo qualitativo constituído por uma proposta de processo e modelo para identificação dos principais *stakeholders* de SoS *Acknowledged*. A pesquisa utiliza-se de entrevistas com especialistas em arquitetura de software ou em SoS para avaliar a eficácia da proposta.

Neste trabalho foi necessário pesquisar em diferentes áreas da computação, o que fez com que muitas definições e conceitos fossem envolvidos no trabalho. Além de analisar como pode ser a relação entre estas diferentes áreas. Assim, com base nos conceitos, definições, características e de acordo com uma visão emergente de arquitetura, onde o arquiteto se aproxima do solicitante, construiu-se o processo de identificação dos principais *stakeholders* de SoS *Acknowledged* e o modelo de exibição dos resultados das iterações desse processo.

O processo proposto é liderado pelo Arquiteto de SoS, que no final é o responsável pelo modelo resultante. A execução do processo é uma atividade colaborativa entre os *stakeholders* interessados e a equipe do SoS, no papel do Arquiteto de SoS. Apenas os *stakeholders* que definem os requisitos que impactam a arquitetura do SoS são entrevistados pelo arquiteto do SoS, e suas respostas são catalogadas para a montagem do modelo. Assim, a cada entrevista realizada, o modelo pode ir se modificando e evoluindo de acordo com as evoluções do SoS.

A opção pelo tema de pesquisa foi por se tratar de um campo de pesquisa emergente que tem uma demanda crescente por soluções. Porém essa característica traz consigo a dificuldade de encontrar pessoas especialistas para entrevistar que pudessem avaliar o estudo, bem como SoS disponíveis para que o processo e modelo fossem colocados em prática. Mesmo assim, com base nos resultados da avaliação, o processo e modelo são factíveis de serem executados e trazerem resultados significativos ao SoS. Apesar da dificuldade em conseguir participantes para a avaliação do trabalho, esta foi muito importante para refletir sobre sugestões de melhorias. Todas as sugestões foram analisadas e algumas foram definidas como trabalhos futuros na evolução do processo e modelo propostos.

Espera-se que com esse trabalho mais pesquisadores possam se interessar pelo tema, dando continuidade à pesquisa, aumentando a abrangência da área de SoS e evoluindo os trabalhos realizados até o momento. Além de auxiliar os SoS a identificar de forma mais fácil seus principais *stakeholders*, mantendo essa informação clara e objetiva a todos os envolvidos no SoS.

8.2 Trabalhos Futuros

Foram levantadas algumas sugestões de trabalhos futuros. A primeira tem relação com a limitação inicial do trabalho, com a expansão do processo e consequentemente modelo para demais categorias de SoS como *Directed*, *Collaborative* e até ao *Virtual*.

Outra sugestão de trabalho futuro tem origem nas entrevistas, onde mais de um participante comenta que seria interessante dividir a responsabilidade com o arquiteto de SoS. Assim algum analista de sistemas ou analista de requisitos ou até arquiteto de sistemas constituintes possam ser envolvidos no processo.

Houve uma sugestão nas entrevistas piloto para que haja um processo para construção do Repositório de Requisitos. Um processo para diminuir a quantidade de requisições que o arquiteto do SoS precisará analisar e também para eliminar solicitações que não estão estruturadas devidamente ou que não tem possibilidade de tornar um ASC.

Outra sugestão que surgiu em discussões após as avaliações foi em ter uma etapa do processo para registrar as decisões arquiteturais tomadas pelos arquitetos para inclusão ou remoção de *stakeholders* do modelo.

E por fim, e talvez a mais importante, a aplicação do processo e modelo num SoS real. Assim possíveis novos "gaps" que ainda não foram identificados podem surgir, além de tornar possível validar o que esta sendo proposto neste trabalho. Com os resultados desta aplicação num SoS real, pode-se refinar o processo e modelo, e na sequencia prototipar uma ferramenta dos mesmos em grande escala ou para uso da comunidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Arrichiello, V. "The originating concept: A foundation for system of systems architecting decision making". In: Proceedings of the International Conference on System of Systems Engineering (SoSE), 2012, pp. 519–524.
- [2] Axelsson, J. "A systematic mapping of the research literature on system-of-systems engineering". In: Proceedings of the International Conference on System of Systems Engineering (SoSE), 2015, pp. 18–23.
- [3] Ballejos, L. C.; Montagna, J. M. "Method for stakeholder identification in interorganizational environments", *Requirements Engineering*, vol. 13–4, 2008, pp. 281–297.
- [4] Barden, R. R.; Evans, D. D. "Managing system of systems requirements with a requirements screening group". In: Proceedings of the International Symposium on International Council on Systems Engineering (INCOSE), 2012, pp. 2031–2039.
- [5] Bass, L.; Clements, P.; Kazman, R. "Software Architecture in Practice". Westford, USA: Addison-Wesley, 2012, 3rd ed., 662p.
- [6] Blair, C. D.; Boardman, J. T.; Sauser, B. J. "Communicating strategic intent with systemigrams: Application to the network-enabled challenge", *Systems Engineering*, vol. 10–4, 2007, pp. 309–322.
- [7] Boardman, J.; Sauser, B. "Systems thinking: Coping with 21st century problems". Boca Raton, USA: CRC Press, 2008, 1st ed., 240p.
- [8] Boardman, J.; Sauser, B. "Systemic thinking: building maps for worlds of systems". New Jersey, USA: John Wiley & Sons, 2013, 1st ed., 262p.
- [9] Boonstra, A.; Boddy, D.; Bell, S. "Stakeholder management in ios projects: analysis of an attempt to implement an electronic patient file", *European Journal of Information Systems*, vol. 17–2, 2008, pp. 100–111.
- [10] Boonstra, A.; de Vries, J. "Managing stakeholders around inter-organizational systems: A diagnostic approach", *The Journal of Strategic Information Systems*, vol. 17–3, 2008, pp. 190–201.
- [11] Butterfield, M. L.; Pearlman, J. S.; Vickroy, S. C. "A system-of-systems engineering geoss: Architectural approach", *IEEE Systems Journal*, vol. 2–3, 2008, pp. 321–332.
- [12] Chen, L.; Babar, M. A.; Nuseibeh, B. "Characterizing architecturally significant requirements", *IEEE Software*, vol. 30–2, 2013, pp. 38–45.

- [13] Chen, P.; Clothier, J. “Advancing systems engineering for systems-of-systems challenges”, *Systems Engineering*, vol. 6–3, 2003, pp. 170–183.
- [14] Cheng, B. H.; Atlee, J. M. “Research directions in requirements engineering”. In: Proceedings of the Future of Software Engineering (FoSE), 2007, pp. 285–303.
- [15] Clarkson, M. “A risk based model of stakeholder theory”. In: Proceedings of the Toronto Conference on Stakeholder Theory (TCOST), 1994, pp. 18–19.
- [16] Costa, C. C.; da Cunha, P. R. “Who are the players? finding and characterizing stakeholders in social networks”. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS), 2010, pp. 1–10.
- [17] Dahmann, J. S.; Baldwin, K. J. “Understanding the current state of us defense systems of systems and the implications for systems engineering”. In: Proceedings of the Annual IEEE Systems Conference (SysCon), 2008, pp. 1–7.
- [18] De Boer, R. C.; Van Vliet, H. “On the similarity between requirements and architecture”, *Journal of Systems and Software*, vol. 82–3, 2009, pp. 544–550.
- [19] Demchak, B.; Farcas, C.; Farcas, E.; Kruger, I. H. “The treasure map for rich services”. In: Proceedings of the International Conference on Information Reuse and Integration (IRI), 2007, pp. 400–405.
- [20] Dvorak, D. L.; Indictor, M. B.; Ingham, M. D.; Rasmussen, R. D.; Stringfellow, M. V. “A unifying framework for systems modeling, control systems design, and system operation”. In: Proceedings of the International Conference on Systems, Man and Cybernetics (SMC), 2005, pp. 3648–3653.
- [21] Fang, Z.; DeLaurentis, D. “Multi-stakeholder dynamic planning of system of systems development and evolution”, *Procedia Computer Science*, vol. 44, 2015, pp. 95–104.
- [22] Fassin, Y. “The stakeholder model refined”, *Journal of Business Ethics*, vol. 84–1, 2009, pp. 113–135.
- [23] Freeman, R. E. “Strategic management: A stakeholder approach”. Cambridge, UK: Cambridge University Press, 2010, 1st ed., 292p.
- [24] GEOSS. “Geoss: About geoss”. Capturado em: <https://www.earthobservations.org/geoss.php>, Jun 2017.
- [25] Gonçalves, M. B.; Cavalcante, E.; Batista, T.; Oquendo, F.; Nakagawa, E. Y. “Towards a conceptual model for software-intensive system-of-systems”. In: Proceedings of the International Conference on Systems, Man and Cybernetics (SMC), 2014, pp. 1605–1610.

- [26] Gonçalves, M. B.; Oquendo, F.; Nakagawa, E. Y. "A meta-process to construct software architectures for system of systems". In: Proceedings of the Annual ACM Symposium on Applied Computing (SAC), 2015, pp. 1411–1416.
- [27] Guide, U. D. S. "Systems engineering guide for systems of systems", *Pentagon*, vol. 1, 2008, pp. 1–148.
- [28] Hester, P. T.; Bradley, J. M.; Adams, K. M. "Stakeholders in systems problems", *International Journal of System of Systems Engineering*, vol. 3–4, 2012, pp. 225–232.
- [29] Hofmeister, C.; Kruchten, P.; Nord, R. L.; Obbink, H.; Ran, A.; America, P. "A general model of software architecture design derived from five industrial approaches", *Journal of Systems and Software*, vol. 80–1, 2007, pp. 106–126.
- [30] Ireland, V.; Rapaport, B.; Omarova, A. "Addressing wicked problems in a range of project types", *Procedia Computer Science*, vol. 12, 2012, pp. 49–55.
- [31] ISO/IEEE. "1233: Guide for developing system requirements specifications", Relatório Técnico, Geneva, Switzerland, 1998, 36p.
- [32] ISO/IEEE. "19501: Information technology - open distributed processing - unified modeling language (uml)", Relatório Técnico, Geneva, Switzerland, 2005, 454p.
- [33] ISO/IEEE. "12207: Information technology - software lifecycle processes", Relatório Técnico, Geneva, Switzerland, 2008, 138p.
- [34] ISO/IEEE. "43010: Systems and software engineering - architecture description", Relatório Técnico, Geneva, Switzerland, 2011, 46p.
- [35] ISO/IEEE. "15288: Systems and software engineering - system life cycle process", Relatório Técnico, Geneva, Switzerland, 2015, 118p.
- [36] Keating, C.; Rogers, R.; Unal, R.; Dryer, D.; Sousa-Poza, A.; Safford, R.; Peterson, W.; Rabadi, G. "System of systems engineering", *Engineering Management Journal*, vol. 15–3, 2003, pp. 36–45.
- [37] Keating, C. B.; Padilla, J. J.; Adams, K. "System of systems engineering requirements: challenges and guidelines", *Engineering Management Journal*, vol. 20–4, 2008, pp. 24–31.
- [38] Kitchenham, B. "Procedures for performing systematic reviews", *Keele University*, vol. 33, 2004, pp. 1–26.
- [39] Liang, P.; Avgeriou, P.; He, K.; Xu, L. "From collective knowledge to intelligence: pre-requirements analysis of large and complex systems". In: Proceedings of the Workshop on Web 2.0 for Software Engineering (Web2SE), 2010, pp. 26–30.

- [40] Likert, R. "A method of constructing an attitude scale", *Scaling: A sourcebook for behavioral scientists*, vol. 1, 1974, pp. 233–243.
- [41] Lim, S. L.; Ncube, C. "Social networks and crowdsourcing for stakeholder analysis in system of systems projects". In: *Proceedings of the International Conference on System of Systems Engineering (SoSE)*, 2013, pp. 13–18.
- [42] Maier, M. W. "Architecting principles for systems-of-systems". In: *Proceedings of the International Symposium on International Council on Systems Engineering (INCOSE)*, 1996, pp. 565–573.
- [43] Maier, M. W. "Architecting principles for systems-of-systems", *Systems Engineering*, vol. 1–4, 1998, pp. 267–284.
- [44] McCaughin, L. K.; White, B. E. "An architecture for stewarding enterprises". In: *Proceedings of the System of Systems Engineering Conference (SoSE)*, 2016, pp. 1–6.
- [45] Mitchell, R. K.; Agle, B. R.; Wood, D. J. "Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts", *Academy of Management Review*, vol. 22–4, 1997, pp. 853–886.
- [46] Muller, G. "Are stakeholders in the constituent systems sos aware? reflecting on the current status in multiple domains". In: *Proceedings of the System of Systems Engineering Conference (SoSE)*, 2016, pp. 1–5.
- [47] Nakagawa, E. Y.; Gonçalves, M.; Guessi, M.; Oliveira, L. B.; Oquendo, F. "The state of the art and future perspectives in systems of systems software architectures". In: *Proceedings of the International Workshop on Software Engineering for Systems-of-Systems (SESoS)*, 2013, pp. 13–20.
- [48] Ncube, C. "On the engineering of systems of systems: Key challenges for the requirements engineering community". In: *Proceedings of the Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RESS)*, 2011, pp. 70–73.
- [49] Overton, J.; Hall, J. G.; Rapanotti, L. "Middle-out design: A proposed best-practice for geoss design", *Relatório Técnico*, Milton Keynes, UK, 2010, 6p.
- [50] Pacheco, C.; Garcia, I. "A systematic literature review of stakeholder identification methods in requirements elicitation", *Journal of Systems and Software*, vol. 85–9, 2012, pp. 2171–2181.
- [51] Pacheco, C.; Tovar, E. "Stakeholder identification as an issue in the improvement of software requirements quality". In: *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE)*, 2007, pp. 370–380.
- [52] Petcu, V.; Petrescu, A. "Systems of systems applications for telemedicine". In: *Proceedings of the Roedunet International Conference (RoEduNet)*, 2010, pp. 208–211.

- [53] Pouloudi, A.; Whitley, E. A. "Stakeholder identification in inter-organizational systems: gaining insights for drug use management systems", *European journal of information systems*, vol. 6–1, 1997, pp. 1–14.
- [54] Power, K. "Stakeholder identification in agile software product development organizations: A model for understanding who and what really counts". In: Proceedings of the Agile Conference (AGILE), 2010, pp. 87–94.
- [55] Rajabalinejad, M.; Bonnema, G. M. "Determination of stakeholders' consensus over values of system of systems". In: Proceedings of the International Conference on System of Systems Engineering (SoSE), 2014, pp. 25–30.
- [56] Rechtin, E. "Systems architecting: Creating and building complex systems". Los Angeles, USA: Prentice Hall, 1991, 1st ed., 333p.
- [57] Rozanski, N.; Woods, E. "Software systems architecture: working with stakeholders using viewpoints and perspectives". Indiana, USA: Addison-Wesley, 2005, 1st ed., 567p.
- [58] Safwat, A.; Senousy, M. "Addressing challenges of ultra large scale system on requirements engineering", *Procedia Computer Science*, vol. 65, 2015, pp. 442–449.
- [59] Salado, A.; Nilchiani, R. "Contextual and behavioral-centric stakeholder identification", *Procedia Computer Science*, vol. 16, 2013, pp. 908–917.
- [60] Savage, G. T.; Nix, T. W.; Whitehead, C. J.; Blair, J. D. "Strategies for assessing and managing organizational stakeholders", *The executive*, vol. 5–2, 1991, pp. 61–75.
- [61] SEBoK, A. "Guide to the systems engineering body of knowledge (sebok) version 1.8". Capturado em: [http://sebokwiki.org/w/index.php?title=Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)&oldid=52784](http://sebokwiki.org/w/index.php?title=Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)&oldid=52784), Ago 2017.
- [62] Sindi, O. V.; DeLaurentis, D. A.; Akaydin, K.; Smith, D. "Exploration of a solar system mobility network via a system-of-systems engineering framework". In: Proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Space Conference & Exposition, 2007, pp. 18–20.
- [63] Sommerville, I. "Software Engineering". London, UK: Pearson Education, 2015, 10th ed., 816p.
- [64] Sommerville, I.; Lock, R.; Storer, T. "Information requirements for enterprise systems". In: Proceedings of the Large-Scale Complex IT Systems. Development, Operation and Management (Monterey Workshop), 2012, pp. 266–282.
- [65] Vierhauser, M.; Rabiser, R. "Monitoring requirements in systems of systems", *IEEE Software*, vol. 33–5, 2016, pp. 22–24.

- [66] Vierhauser, M.; Rabiser, R.; Grünbacher, P.; Aumayr, B. "A requirements monitoring model for systems of systems". In: Proceedings of the International Requirements Engineering Conference (RE), 2015, pp. 96–105.
- [67] Wiegers, K.; Beatty, J. "Software Requirements". Washington, USA: Pearson Education, 2013, 3rd ed., 673p.
- [68] Yang-Turner, F.; Lau, L. "A pragmatic strategy for creative requirements elicitation: from current work practice to future work practice". In: Proceedings of the Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), 2011, pp. 28–31.
- [69] Yue, Y.; Barkley, H. "A system of systems representation of solving a complex problem: the establishment of a maritime safety radar", *International Journal of System of Systems Engineering*, vol. 6–3, 2015, pp. 237–252.
- [70] Zachman, J. A. "Enterprise architecture: The issue of the century", *Database Programming and Design*, vol. 10–3, 1997, pp. 44–53.

ANEXO A – Execução da Revisão da Literatura

O processo de Revisão da Literatura iniciou-se com base no motor de busca Scopus, que é um projeto da editora Elsevier que indexa os conteúdos das bibliotecas digitais mais utilizadas na Ciência da Computação como ACM, IEEE e Science Direct, além de documentos de índices de outras áreas como Engenharias. A *string* de busca que é o meio pelo qual os trabalhos são procurados, foi dividida em dois grupos: variações do termo *stakeholder* e as diversidades dos sistemas. Foram pesquisados trabalhos nos campos Título, Palavras Chaves e Abstract. Inicialmente a *string* de busca ficou definida conforme a seguir:

String de Busca:

```
TITLE-ABS-KEY (
    ("Stakeholder" OR "Stakeholder Identification" OR "identification of stakeholder" OR
    "Stakeholder Model" OR "management of stakeholder" OR "Stakeholder Analysis")
    AND
    ("system-of-system" OR "system of systems" OR "SoS" OR "complex system" OR
    "distributed system" OR "large system" OR "inter-organizational system")
)
```

Essa *string* retornou 1.192 trabalhos, então aplicou-se alguns filtros para refinar a busca. Primeiramente, limitou-se à artigos escritos nas linguas inglesa e portuguesa. A nova *string* ficou como segue:

String de Busca Refinada:

```
TITLE-ABS-KEY (
    ("Stakeholder" OR "Stakeholder Identification" OR "identification of stakeholder" OR
    "Stakeholder Model" OR "management of stakeholder" OR "Stakeholder Analysis")
    AND
    ("system-of-system" OR "system of systems" OR "SoS" OR "complex system" OR
    "distributed system" OR "large system" OR "inter-organizational system")
)
AND
(LIMIT-TO( LANGUAGE,"English" ) OR LIMIT-TO(LANGUAGE,"Portuguese" ))
```

Com essas restrições, o número de trabalhos reduziu pouco, para 1.178. Então, decidiu-se aplicar mais um filtro na área de publicação dos trabalhos para Computação e Engenharia. A nova *string* de busca ficou conforme a seguir:

String de Busca Definitiva:

```

TITLE-ABS-KEY (
  ("Stakeholder" OR "Stakeholder Identification" OR "identification of stakeholder" OR
  "Stakeholder Model" OR "management of stakeholder" OR "Stakeholder Analysis")
  AND
  ("system-of-system" OR "system of systems" OR "SoS" OR "complex system" OR
  "distributed system" OR "large system" OR "inter-organizational system")
)
AND
(LIMIT-TO( LANGUAGE,"English" ) OR LIMIT-TO(LANGUAGE,"Portuguese" ))
AND
(LIMIT-TO(SUBJAREA,"COMP ") OR LIMIT-TO(SUBJAREA,"ENGI "))

```

Com isso, conseguiu-se limitar os resultados a 767 trabalhos. Esses trabalhos foram exportados com suas informações para arquivo no formato "bib" e importados na ferramenta Start. Essa ferramenta foi utilizada para apoiar na seleção, classificação e extração dos trabalhos relevantes para a pesquisa. A primeira seleção executada tinha o objetivo de definir quais trabalhos estavam realmente relacionados com a identificação, classificação ou alguma forma de definição dos *stakeholders* envolvidos em SoS ou algum sistema tão complexo quanto. Nesta primeira classificação, encontrou-se 14 (2%) trabalhos duplicados, 96 (12%) trabalhos aceitos e 657 (86%) trabalhos rejeitados.

Os 96 trabalhos aceitos foram procurados para leitura por completo para extrair as informações relevantes. Destes, 78 (81%) trabalhos foram rejeitados por não auxiliarem a responder a questão de pesquisa ou por não ter nenhuma relação direta com algum dos tipos de sistemas relacionados na busca ou com *stakeholders*. Enquanto que os 18 (19%) trabalhos restantes foram aceitos justamente por esclarecerem dúvidas ou ajudarem a responder a questão de pesquisa. A Tabela A.1 apresenta as principais contribuições destes 18 trabalhos selecionados ordenados pelos anos de publicação.

Tabela A.1 – Trabalhos selecionados

Trabalho	Principal Contribuição
Muller (2016) [46]	Trabalho com diversos estudos de casos em vários variados para ilustrar um SoS.
McCaughin e White (2016) [44]	Ressalta a importância da identificação de <i>stakeholders</i> , apresenta um passo a passo do que deve ser feito, porém não diz como envolver o <i>stakeholder</i> . Apresenta uma arquitetura de <i>stakeholders</i> da empresa usada para orientar esse desenvolvimento de administração. Finalmente, apresenta uma abordagem para gerenciar qualquer empresa dentro do seu ambiente para criar valor para todos os <i>stakeholders</i> .

Yue e Barkley (2015) [69]	Este trabalho apresenta um estudo de caso que mostra como os <i>stakeholders</i> mudam de perfil ao longo do tempo. Além de apresentar os problemas e dificuldades enfrentadas dentro do estudo de caso.
Fang e De-Laurentis (2015) [21]	Propõe um <i>framework</i> descentralizado de multi- <i>stakeholders</i> para tomada de decisão onde os <i>stakeholders</i> locais (dos sistemas constituintes) conduzem o planejamento de desenvolvimento para benefícios individuais, enquanto o participante do nível SoS projeta um mecanismo de coordenação para facilitar a comunicação entre os <i>stakeholders</i> e alcançar um resultado harmonioso. Apresenta o <i>framework</i> , mas não define como identificar os <i>stakeholders</i> .
Safwat e Senousy (2015) [58]	Trabalho aponta as limitações das práticas ER atuais para os desafios forçados pela natureza grandes sistemas, como SoS, e se concentra nas contribuições de várias abordagens para superar essas dificuldades para abordar as áreas não resolvidas dessas soluções. Para identificar <i>stakeholders</i> , os autores citam o trabalho de Lim e Ncube [41] com a ferramenta Stakesource 2.0. Ferramenta baseada em crowdsourcing.
Rajabalinejad e Bonnema (2014) [55]	Sugere um método baseado na flexibilidade e tolerância. A partir de um <i>framework</i> , coleta, processa e combina informações de incertezas. Este <i>framework</i> introduz o 'grau de consenso' como uma métrica para ranquear os valores identificados, esta métrica representa o consentimento dos <i>stakeholders</i> no SoS e pode ser utilizada na arquitetura, nas decisões de design entre outras situações. Este <i>framework</i> usa um ferramenta gráfica para se comunicar com os <i>stakeholders</i> , coletar as informações e combina-las para gerar o grau de consenso. Utiliza a comunicação para identificar os <i>stakeholders</i> . Conjunto de questões propostas por [59].
Lim e Ncube (2013) [41]	Propõem utilizar redes sociais e <i>crowdsourcing</i> para identificar e priorizar os <i>stakeholders</i> em projetos de SoS. Através de uma ferramenta web (StakeSource), o método proposto faz os <i>stakeholders</i> recomendarem outros <i>stakeholders</i> , construindo uma rede social de <i>stakeholders</i> e priorizando os <i>stakeholders</i> utilizando métricas de redes sociais.
Arrichiello (2012) [1]	Define o conjunto inicial de <i>stakeholders</i> para SoS. Descreve uma abordagem para realizar a primeira das tarefas: entender a necessidade e o problema, traduzindo claramente para requisitos. Com o objetivo de fornecer uma base sólida para o processo de decisão e, ao mesmo tempo, para todo o processo de desenvolvimento do SoS.

Sommerville (2012) [64]	Um estudo de caso Gerenciamento de Emergências Civis. Este estudo de caso apresenta uma abordagem de ER baseada em modelos de responsabilidade. Os modelos de responsabilidade servem como base para a identificação dos <i>stakeholders</i> que devem ser considerados no estabelecimento dos requisitos. No processo identificam-se explicitamente as responsabilidades dos <i>stakeholders</i> da organização em uma configuração de problema e um modelo mostra essas responsabilidades e sua atribuição aos agentes. Isso serve como base para a identificação de <i>stakeholders</i> e para identificar se existem ou não inconsistências na percepção de responsabilidade nas diferentes organizações envolvidas. A identificação dos <i>stakeholders</i> está baseada num conjunto de questões através de entrevistas, estruturando assim a conversa do engenheiro de requisitos e do <i>stakeholder</i> .
Ireland; Raport e Omarova (2012) [30]	Objetivo do trabalho é reconhecer a necessidade de problemas ruins (<i>wicked problems</i>) serem vistos como um subconjunto de projetos complexos, buscando contribuir para identificar métodos de solução para estes e tirar algumas conclusões sobre como esses métodos são diferentes dos outros sistemas complexos. <i>Wicked Problems</i> podem ser o terrorismo, as mudanças climáticas, as controvérsias internacionais e as drogas ilícitas.
Yang-Turner e Lau (2011) [68]	O trabalho apresenta uma estratégia pragmática desenvolvida em um projeto de SoS que define o objetivo, o escopo e quatro princípios sobre como reunir todos os <i>stakeholders</i> para trabalhar junto dos usuários finais. Apresenta a experiência dos autores ao explorar as questões estratégicas em Dicode (<i>Data-intensive collaboration and decision making</i>).
Liang; Avgeriou; He e Xu (2010) [39]	Os autores propõe um <i>framework</i> para tratar pré-requisitos em sistemas complexos, através do uso de técnicas de inteligência coletiva baseadas em ferramentas e tecnologias populares como <i>wikis</i> , <i>tags</i> , e <i>semantic web</i> .
Costa e Cunha (2010) [16]	Os autores apresentam uma abordagem para a realização da identificação dos <i>stakeholders</i> considerando as demandas específicas de redes sociais.
Boonstra e de Vries (2008) [10]	O trabalho apresenta uma abordagem diagnóstica para a identificação de <i>stakeholders</i> em sistemas inter-organizacionais e a avaliação na perspectiva de seu poder e interesses em relação a esses sistemas. Esta abordagem é ilustrada através de um estudo de caso.
Boonstra; Boddy e Bell (2008) [9]	O trabalho desenvolve e testa um modelo de gerenciamento de <i>stakeholders</i> baseado no poder, na influência e nas atitudes. Ressaltando, em um estudo de caso, que aqueles com alto interesse no sistema não tinham o poder de implementá-lo, enquanto aqueles com baixo interesse tinham o poder de bloqueá-lo.

Demchak; Farcas, C; Farcas, E e Kruger (2007) [19]	O trabalho descreve um processo de desenvolvimento que influencia um tipo de arquitetura SOA. Esse processo permite a decomposição de uma arquitetura de sistema em preocupações separadas, capturando diferentes aspectos do sistema e suas interações. Essa decomposição se faz necessária pois arquitetura SOA é empregada em grandes sistemas e há um grande número de <i>stakeholders</i> envolvidos.
Blair; Board- man e Sauser (2007) [6]	O trabalho apresenta uma visão geral de um estudo de caso do Ministério da Defesa do Reino Unido, representando-o como uma atividade humana complexa de SoS cuja análise não pode depender de abordagens de ES puramente tradicionais. Traz uma revisão da literatura, abrangendo metodologias de sistemas não tradicionais e apresenta o diagrama sistêmico (<i>systemigram</i>). Apesar de não abordar o tema identificação de <i>stakeholders</i> , apresenta uma técnica que pode ser utilizada para construção de modelos, na qual utiliza <i>systemigram</i> . Neste caso o <i>systemigram</i> é decomposto em vários temas com o objetivo de serem utilizados num diálogo entre <i>stakeholders</i> para esclarecer o debate de questões-chave, resultando num modelo de sistema simplificado que pode ser usado na comunicação da estratégia.
Pouloudi e Whitley (1997) [53]	O trabalho descreve como identificar os <i>stakeholders</i> em um estudo de caso e conforme o autor, um processo que foi negligenciado na análise de <i>stakeholders</i> e literatura de sistemas inter-organizacionais. Examina as percepções de uma série de <i>stakeholders</i> no campo de gerenciamento de uso de drogas no Reino Unido.

ANEXO B – Questionário aos Especialistas e Arquitetos de SoS

B.1 Termo de Consentimento

Processo e Modelo de Identificação de Stakeholders em SoS Acknowledged

Somos pesquisadores da PUCRS avaliando um processo e um modelo para identificação de Stakeholders em Sistema-de-Sistemas (SoS do inglês System-of-Systems) Acknowledged.

Para isto, os participantes são convidados a avaliar o processo e modelo proposto através desse questionário. Lembramos que o objetivo deste estudo não é avaliar o participante, mas sim **avaliar o modelo proposto**.

O uso que se faz dos registros efetuados durante o questionário é estritamente limitado a atividades de pesquisa, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências e periódicos) ou pedagógicos (tais como materiais de aula).
2. Todo participante terá acesso a cópias destes documentos após a publicação dos mesmos, caso tenha interesse.
3. Todo participante que se sentir constrangido ou incomodado pode interromper sua participação e estará fazendo um favor à equipe se registrar por escrito as razões ou sensações que o levaram a esta atitude. A equipe fica obrigada a descartar os testes para fins da avaliação a que se destinaria.
4. A equipe tem direito de utilizar os dados de todos, mantidas as condições acima mencionadas, para quaisquer fins acadêmicos ou pedagógicos contemplados por seus membros.

O tempo estimado de participação é de 25 minutos.

Agradecemos a sua contribuição,

Mateus Balen (PUCRS, Brasil) - mateus.balen@acad.pucrs.br – aluno de mestrado

Sabrina Marczak (PUCRS, Brasil) - sabrina.marczak@pucrs.br – professora orientadora

Sim, concordo com o termo de compromisso

Informações Gerais

Esta pesquisa faz parte de um projeto de mestrado que visa definir um processo teórico e um modelo para identificar os principais stakeholders de SoS Acknowledged

Com o objetivo de facilitar o entendimento do objetivo do trabalho, alguns documentos foram anexados e podem ser consultados. Mais Detalhes sobre Conceitos, Características, Taxonomia e Categorias de SoS: <https://goo.gl/VQixsw>

Diversos pesquisadores como (Lim, Ncube, Ballejos, Montagna) afirmam que: Bons métodos de identificação de stakeholders convencionais podem funcionar em projetos pequenos ou monolíticos. Mas ficam aquém dos projetos de SoS devido a incapacidade de atender a complexidade e à grande número de stakeholders envolvidos de um SoS.

Partindo deste problema que elaboramos nossa questão de pesquisa:

Questão de Pesquisa (QP):

QP: Como definir os principais *stakeholders* de um *SoS Acknowledged*?

B.2 Dados do Perfil do Participante

1) Nome Completo:

2) Gênero:

- Feminino
- Masculino
- Outro

3) Idade:

4) Escolaridade:

- Ensino Médio Incompleto
- Ensino Médio

- Ensino Superior Incompleto
- Ensino Superior
- Pós-graduação incompleto
- Pós-Graduação

5) Função atual: *Marque todas que se aplicam.*

- Administrador de banco de dados
- Analista de negócios
- Analista de segurança
- Analista de sistemas
- Analista de telecomunicações
- Arquiteto de software
- Auditor de TI
- Desenvolvedor de software
- Designer de interfaces
- Diretor / gestor de TI
- Engenheiro de software
- Engenheiro de testes
- Estudante
- Gerente
- Instrutor / professor de TI
- Testador de software
- Suporte técnico
- Outro:

6) Tempo de experiência, em anos, na função atual:

7) Com relação a sua experiência, quais funções (ex. pesquisador, desenvolvedor, etc) já exerceu no tema Sistema-de-Sistemas (SoS):

8) Tempo de experiência total, em anos, com SoS:

B.3 Questionário parte 1 - Sobre a necessidade

De acordo com o Departamento de Defesa americano (DoD), SoS é um conjunto ou arranjo de sistemas que resultam, quando independentes e úteis, em sistemas integrados que entregam capacidades únicas. A categoria de SoS Acknowledged abrange sistemas que reconhecem os objetivos do SoS, possuem um gerente/gestor e recursos próprios para o SoS. Porém os sistemas que fazem parte do SoS tem independência de propriedade, objetivos, financiamento, desenvolvimento e sustentação. As mudanças nos sistemas são baseadas na colaboração entre o SoS e os sistemas constituintes.

1) A identificação dos stakeholders de um SoS é importante?

Discordo Completamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo Completamente
---------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	---------------------------

Justifique:

2) Caso já tenha participado de um processo de identificação de stakeholders em SoS. Como esse processo ocorreu, quais técnicas foram utilizadas?

Segundo a ISO 1233, de 1998 que constitui-se de um guia para o desenvolvimento de especificações de requisitos de sistemas, as técnicas para identificação de requisitos em sistemas tradicionais são:

- **Workshops estruturados:** reuniões estruturadas de um grupo de pessoas interessadas em um determinado assunto;
- **Brainstorming ou sessões para resolução de problemas:** reuniões de um grupo de pessoas interessadas em explorar a potencialidade criativa de um indivíduo ou de um grupo para resolução de problemas;
- **Entrevistas:** conversas entre pessoas interessadas para obter informações necessárias;
- **Surveys ou questionários:** questionários estruturados com o objetivo de coletar informações;
- **Observação de padrões de trabalho;**

- **Observação de sistemas ambiente socio-organizacional;**
- **Revisão de documentação técnica;**
- **Análise de mercado:** análise do que seus clientes querem e do que seus concorrentes oferecem;
- **Avaliação de sistemas concorrentes;**
- **Engenharia Reversa:** processo de descobrir os princípios e o funcionamento de um sistema, através da análise de sua estrutura, função e operação. Desconstruir o que existe para entender como funciona;
- **Simulações:** processo de simular a interação do usuário no sistema;
- **Prototipações:** visão inicial de parte do sistema que possibilita demonstrar conceitos e experimentar opções de implementação;
- **Processos e sistemas de Benchmarking:** comparação de processos e de sistemas;

3) SOMENTE as técnicas de identificação de requisitos existentes na engenharia de software tradicional, como as listadas pela ISO 1233, são suficientes para identificar os stakeholders de um SoS?

As técnicas tradicionais buscam todos os indivíduos ou grupos de indivíduos que tem legítimo interesse no sistema ao longo do seu ciclo de vida.

Discordo Completamente Concordo Completamente

Justifique:

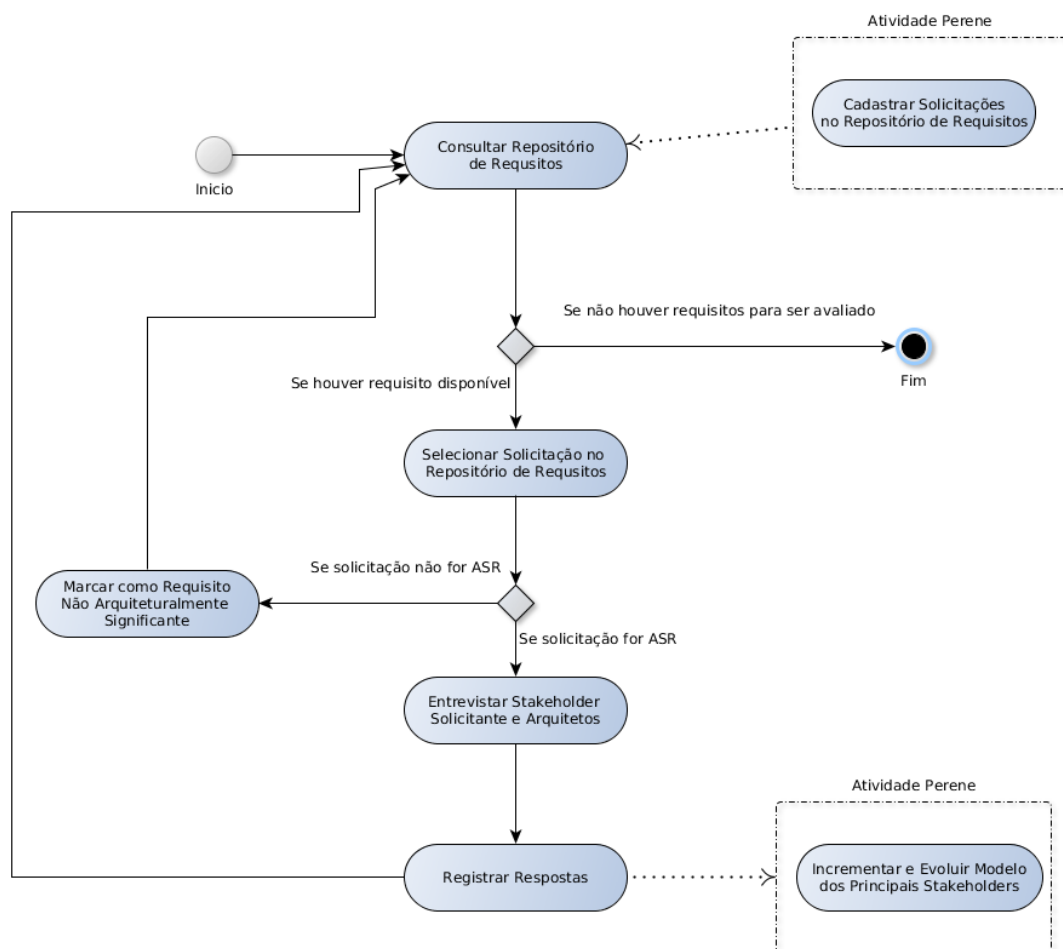
Segundo a ISO 15288 de 2015 que apresenta processos que compõe o ciclo de vida de um software, processo é conjunto de atividades inter-relacionadas ou interativas que transformam entradas em resultados.

4) É importante se ter um processo para realizar a identificação dos stakeholders de SoS?

Justifique:

B.4 Questionário parte 2 - Sobre o Processo e Modelo Propostos

Abaixo apresenta-se o diagrama de atividades e a descrição das etapas do processo colaborativo para identificação dos principais stakeholders em SoS Acknowledged. Neste processo, cada requisito é analisado e os arquiteturalmente significantes são submetidos a questionamentos para identificação dos stakeholders envolvidos.



Processo para identificação dos principais **stakeholders** de SoS **Acknowledged**:

1) O Processo de identificação dos principais *stakeholders* começa de forma colaborativa. Todos os interessados no SoS podem submeter solicitações em um local comum, por exemplo através de um portal. Cada solicitação recebida fica armazenada num **Repositório de Requisitos**.

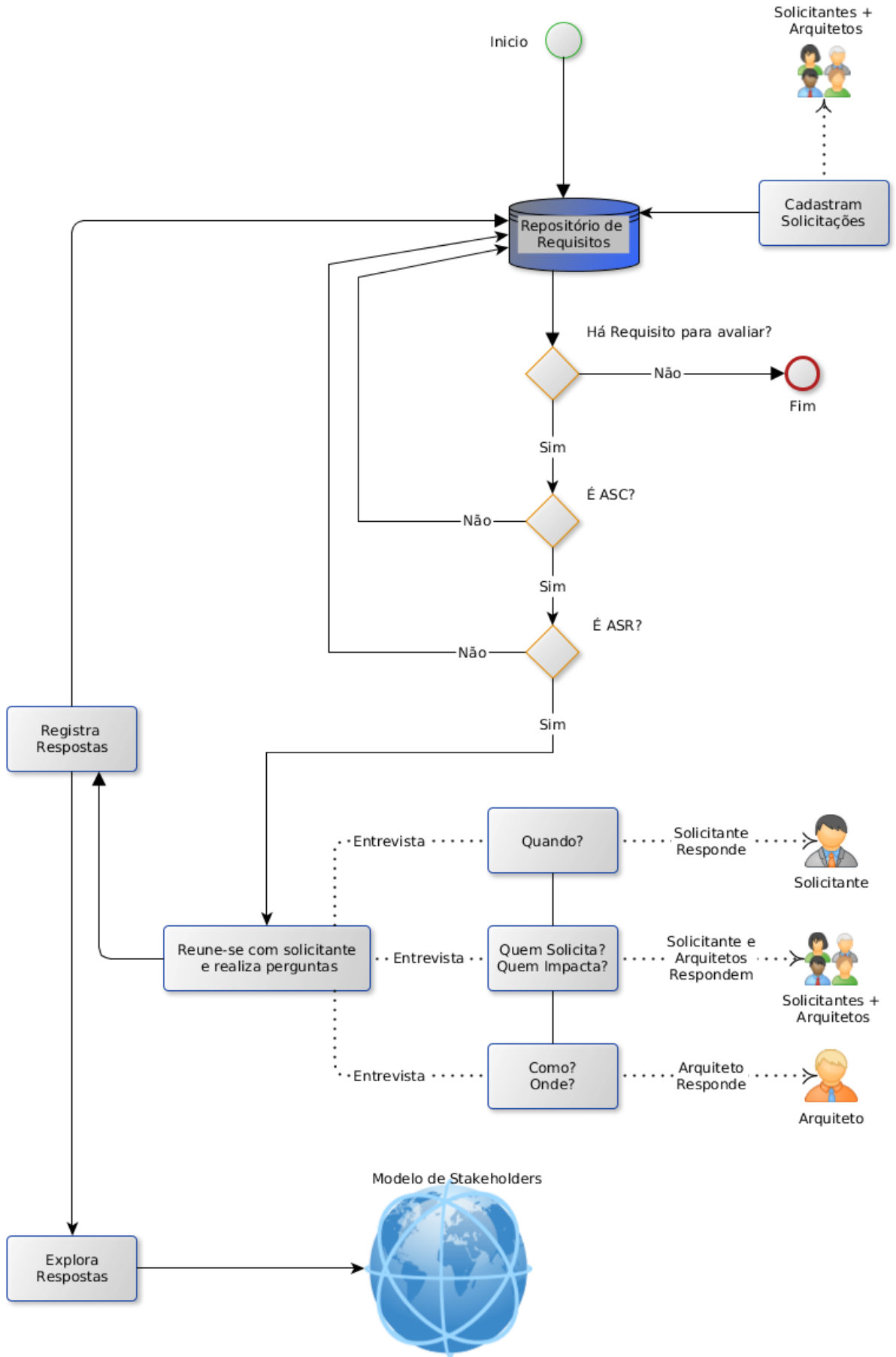
2) Os arquitetos do SoS, avaliam as solicitações existentes no Repositório de Requisitos. Para cada solicitação, o arquiteto do SoS avalia se essa é realmente uma **preocupação arquiteturalmente significativa** ou ASC. Se não for a mesma é marcada como solicitação não impactante na arquitetura e conseqüentemente ignorada no processo. Caso a solicitação seja possivelmente significativa para arquitetura, então o arquiteto precisa contactar o stakeholder solicitante para obter

mais informações para verificar se o ASC torna-se um **requisito arquiteturalmente significativo** ou ASR respondendo a um conjunto de perguntas. No caso de um ASC não tornar-se um ASR o mesmo é descartado da análise, porém se o ASC realmente for um ASR esta solicitação segue em avaliação.

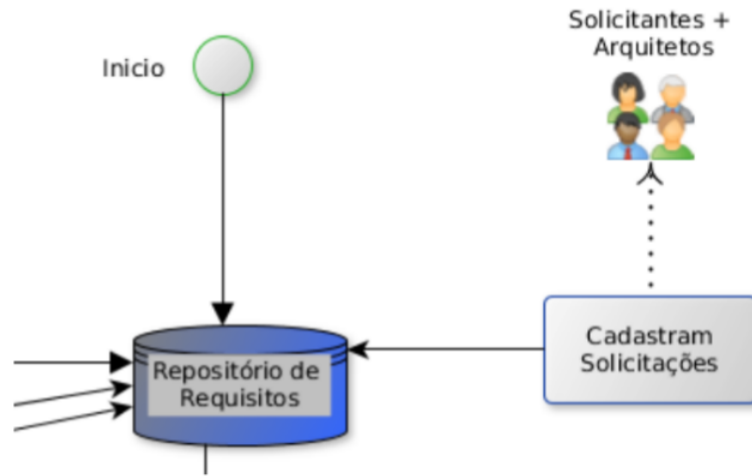
3) Quando *stakeholder solicitante e arquiteto se reúnem*, esclarecem detalhes da solicitação e respondem as perguntas: 'quando' a solicitação será implementada, a data limite para a implementação, "quem solicita" e "quem impacta" está solicitação, elencando os perfis de *stakeholders* solicitantes e impactantes em cada um dos níveis: SoS, Sistemas Constituintes e se necessário Subsistemas dos Sistemas Constituintes. As perguntas "como" a solicitação será implementada e "onde" será implementada serão respondidas pelo próprio arquiteto durante ou após o encontro com o *stakeholder* solicitante.

4) As **respostas** para cada requisito com impacto na arquitetura são **catalogadas** pelo arquiteto seguindo um padrão e uma ordem com o objetivo de todos os trabalhos gerarem os dados na mesma estrutura. A partir desse catálogo o modelo é incrementado, evoluindo a cada novo requisito arquiteturalmente significativo analisado e novas atualizações são geradas ao modelo.

Processo para identificação dos principais stakeholders em SoS Acknowledged



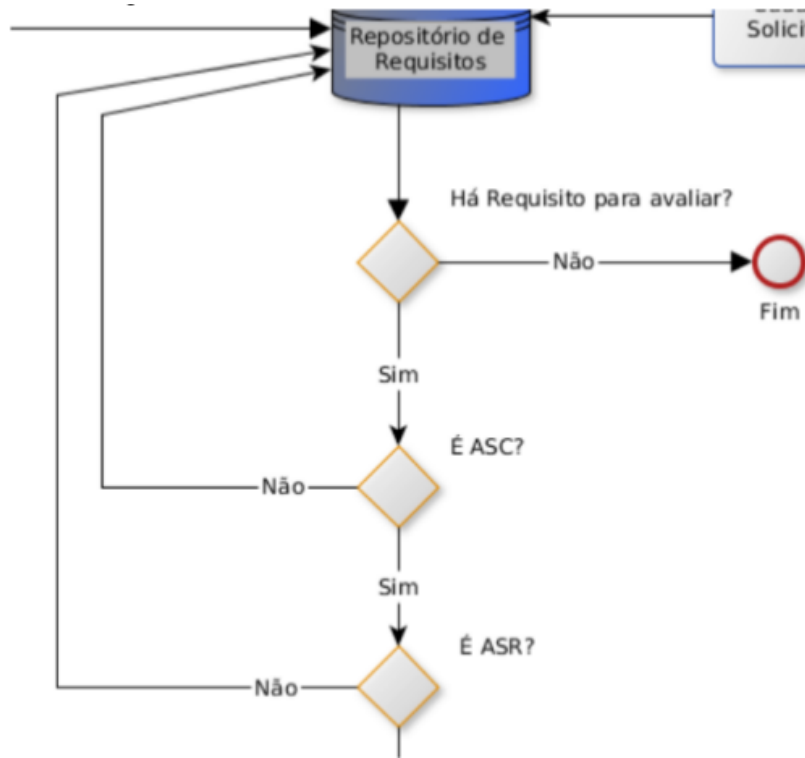
5) O Repositório de Requisitos, sendo construído de forma colaborativa, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?



Discordo Completamente Concordo Completamente

Justifique:

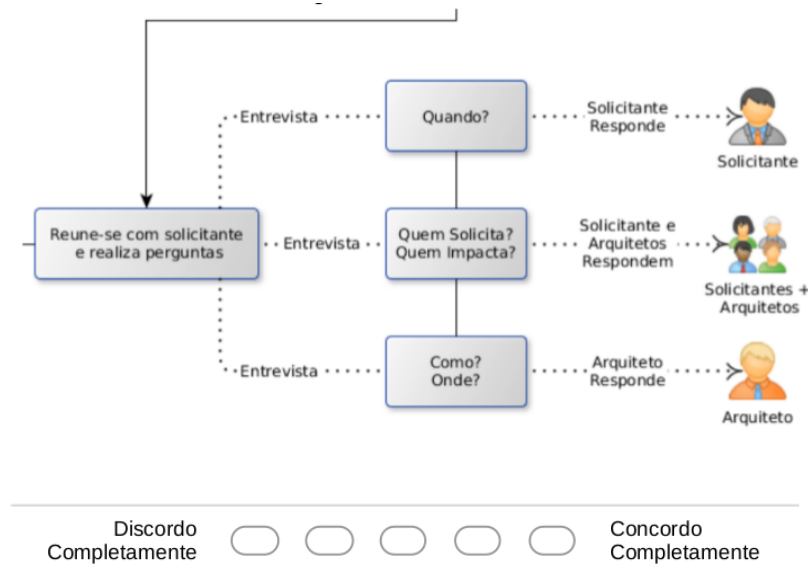
6) O arquiteto do SoS ser o responsável pela seleção de ASC e ASR, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?



Discordo Completamente Concordo Completamente

Justifique:

7) O arquiteto do SoS ser o responsável pela entrevista com os stakeholders que definem ASRs, constitui em uma atividade viável de ser realizada no contexto de SoS Acknowledged?



Justifique:

Segundo a ISO 1233, um modelo é uma representação de um processo, dispositivo ou conceito do mundo real.

Para a construção do modelo, segue-se os passos:

Cada requisito com impacto na arquitetura que foi catalogado é analisado para avaliar evolução da representação dos principais *stakeholders* no modelo. Este modelo é construído da seguinte forma:

1º) O SoS será representado por um nodo. Aconselha-se deixar este nodo em evidência com uma cor que possa diferenciá-lo dos demais nodos.

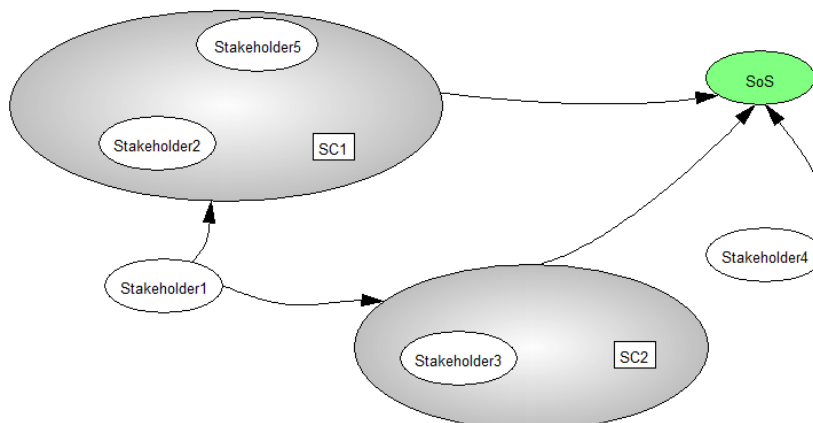
2º) Os *stakeholders* exclusivos do SoS, ou seja, os *stakeholders* que não possuem nenhuma ligação direta com os sistemas constituintes, serão ligados diretamente ao SoS.

3º) Os *stakeholders* que estão relacionados com sistemas constituintes específicos devem estar dentro de um nodo que representa o sistema constituinte em questão. Este nodo delimita as fronteiras do sistema constituinte.

4º) Os *stakeholders* que estão relacionados com mais de um sistema constituinte podem ficar fora do nodo do sistema constituinte.

SC1 e SC2 são dois Sistemas Constituintes do SoS Stakeholders1 à Stakeholders5 são grupos de *stakeholders*

Exemplo hipotético de Modelo de Identificação dos Principais Stakeholders



8) O resultado do processo é um modelo. Este modelo que representa o resultado da execução do processo é importante?

Discordo Completamente Concordo Completamente

Justifique:

9) Você acredita que este processo e o modelo resultante funcionariam num SoS real? Justifiquei sua opinião

10) Sugestão de melhoria no processo ou no modelo?



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br