

INSTITUTO TECNOLÓGICO DE AERONÁUTICA



**Fernando Oliveira Lisboa
Lorenzo Pellizzaro Lima**

**PROPOSTA DE FRAMEWORK SMAD-CAPELLA
UTILIZANDO COMO ESTUDO DE CASO O
CUBESAT RAINCUBE**

**Trabalho de Graduação
2020**

**Curso de Engenharia Aeronáutica e
Aeroespacial**

Fernando Oliveira Lisboa

Lorenzo Pellizzaro Lima

**PROPOSTA DE FRAMEWORK SMAD-CAPELLA
UTILIZANDO COMO ESTUDO DE CASO O
CUBESAT RAINCUBE**

Orientador

Prof. Dr Luis Eduardo Vergueiro Loures da Costa (ITA)

Co-orientador

Prof. Dr Christopher Shneider Cerqueira (ITA)

ENGENHARIA AERONÁUTICA E AEROESPACIAL

SÃO JOSÉ DOS CAMPOS
INSTITUTO TECNOLÓGICO DE AERONÁUTICA

2020

Dados Internacionais de Catalogação-na-Publicação (CIP)

Divisão de Informação e Documentação

Lisboa, Fernando Oliveira e Lima, Lorenzo Pellizzaro
Proposta de Framework SMAD-Capella Utilizando Como Estudo de Caso o Cubesat RaInCube /
Fernando Oliveira Lisboa e Lorenzo Pellizzaro Lima
São José dos Campos, 2020.
51f.

Trabalho de Graduação – Cursos de Engenharia Aeronáutica e Aeroespacial – Instituto Tecnológico de Aeronáutica, 2020. Orientador: Prof. Dr. Luis Eduardo Vergueiro Loures da Costa. Co-orientador: Prof. Dr. Christopher Shneider Cerqueira.

1. SMAD. 2. Arcadia. 3. RaInCube. I. Lorenzo Pellizzaro Lima II. Instituto Tecnológico de Aeronáutica. III. Título.

REFERÊNCIA BIBLIOGRÁFICA

LISBOA, Fernando Oliveira; LIMA, Lorenzo Pellizzaro. **Proposta de Framework SMAD-Capella Utilizando Como Estudo de Caso o CubeSat RaInCube**. 2020. 51f.. Trabalho de Conclusão de Curso. (Graduação em Engenharia Aeronáutica e Aeroespacial) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

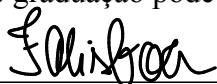
CESSÃO DE DIREITOS

NOME DOS AUTORES: Fernando Oliveira Lisboa e Lorenzo Pellizzaro Lima

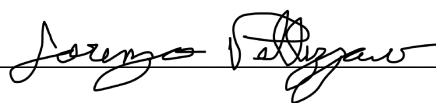
TÍTULO DO TRABALHO: Proposta de Framework SMAD-Capella Utilizando Como Estudo de Caso o CubeSat RaInCube

TIPO DO TRABALHO/ANO: Graduação / 2020

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias deste trabalho de graduação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte deste trabalho de graduação pode ser reproduzida sem a autorização dos autores.



Fernando Oliveira Lisboa
Rua H8B-215
12.228-461, São José Dos Campos - SP



Lorenzo Pellizzaro Lima
Rua H8A-121
12.228-460, São José Dos Campos - SP

PROPOSTA DE FRAMEWORK SMAD-CAPELLA UTILIZANDO COMO ESTUDO DE CASO O CUBESAT RAINCUBE

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação



Fernando Oliveira Lisboa

Autor



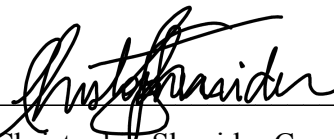
Lorenzo Pellizzaro Lima

Autor



Prof. Dr. Luis Eduardo Vergueiro Loures da Costa

Orientador



Prof. Dr. Christopher Shneider Cerqueira

Coorientador



Prof. Dr. Mauricio Andrés Varela Morales

Coordenador do Curso de Engenharia Aeronáutica

Cristiane Aparecida
Martins

Assinado digitalmente por Cristiane Aparecida Martins
DN: C=BR, OU=Instituto Tecnológico de Aeronáutica, O=Cristiane
Martins, CN=Cristiane Aparecida Martins, E=cmartins@ita.br
Razão: Eu sou o autor deste documento
Localização:
P
Data: 2020-11-26 14:35:52

Prof. Dra. Cristiane Aparecida Martins

Coordenadora do Curso de Engenharia Aeroespacial

São José dos Campos, 23 de Novembro de 2020

Dedicamos este trabalho aos familiares, por todo apoio durante esta jornada e aos amigos do Viradão Team, que estiveram presentes nas madrugadas que antecederam diversas provas ao decorrer dos longos anos do ITA.

Agradecimentos

Ao Prof. Dr. Christopher Shneider Cerqueira, por ter orientado a realização deste trabalho e ter estado presente, ainda que virtualmente, durante as segundas-feiras para desmistificar um pouco da grande complexidade que cerca a engenharia de sistemas. *Chris* foi mais que um brilhante orientador, foi nosso companheiro e amigo.

Ao Prof. Dr. Luis Eduardo Vergueiro Loures da Costa, por ter nos introduzido durante a graduação à engenharia de sistemas e tornado esta área tão instigante, o que culminou neste trabalho - cujo tema foi de sua sugestão.

*“Try not to become a man of success,
but rather try to become a man of value.”*

— ALBERT EINSTEIN

Resumo

O trabalho tem por finalidade o estudo de caso de um satélite CubeSat, o RaInCube, visando validar um modelo que conecte duas metodologias distintas de engenharia de sistemas, buscando, para isso, extrair suas principais informações técnicas, de forma que seja possível replicar parcial ou completamente sua estrutura funcional e operacional.

O RaInCube é um satélite experimental do tipo CubeSat, satélite miniatura aplicado em pesquisas espaciais, que foi desenvolvido pelo *Jet Propulsion Laboratory* (JPL), um laboratório de pesquisa da NASA, em 2018. Esse protótipo teve por finalidade a validação de novas tecnologias que pudessem expandir a aplicabilidade de radares em pequenos satélites, com aplicações meteorológicas.

A aplicação de radares meteorológicos em satélites do tipo CubeSat se mostrou potencialmente atraente porque esse tipo de satélite tem custo e complexidade consideravelmente menores em relação a modelos tradicionais.

A engenharia de sistemas do RaInCube foi desenvolvida com a teoria de Wertz, a metodologia Arcadia e depois aplicada com a ferramenta Capella. Foi realizada uma integração entre a teoria SMAD e a metodologia Arcadia, que permitiu a implementação na ferramenta Capella, pois esta é baseada em Arcadia.

Os resultados obtidos em Capella poderão ser utilizados como referência para modelagem de engenharia de sistemas de futuros projetos aeroespaciais desenvolvidos pelo CEI, Centro Espacial do ITA.

Abstract

The purpose of this work is the case study of a CubeSat satellite, RaInCube, aiming to validate a model that connects two different systems engineering methodologies, seeking, for this, to extract its main technical information, in a way that it is possible to completely or partially replicate its functional and operational structure.

RaInCube is an experimental satellite of the CubeSat type, a miniature satellite applied to space research, which was developed by the Jet Propulsion Laboratory (JPL), a NASA research laboratory, in 2018. This prototype was intended to validate new technologies that could expand the applicability of radars on small satellites, with meteorological applications.

The application of meteorological radars to CubeSat satellites has proved to be potentially attractive because this type of satellite has considerably lower cost and complexity compared to traditional models.

The systems engineering of RaInCube was developed with the theory of Wertz, the Arcadia methodology and later applied with the Capella tool. An integration between the SMAD theory and the Arcadia methodology was carried out, which allowed the implementation in the Capella tool, since it is based on Arcadia.

The purpose of this work is the case study of a CubeSat satellite, RaInCube, aiming to validate a model that connects two different systems engineering methodologies, seeking, for this, to extract its main technical information, in a way that it is possible to partially or partially replicate its functional and operational structure.

The results obtained in Capella can be used as a reference for modeling engineering systems for future aerospace projects developed by CEI, ITA Space Center.

Lista de Figuras

FIGURA 2.1 – Passos da metodologia SMAD para desenvolvimento de um projeto. (CERQUEIRA, 2020).	21
FIGURA 2.2 – Exemplo de diagrama operacional (ROGUES, 2017).	28
FIGURA 2.3 – Exemplo de arquitetura de sistema (ROGUES, 2017).	29
FIGURA 2.4 – Exemplo de arquitetura lógica (ROGUES, 2017).	30
FIGURA 2.5 – Exemplo de arquitetura lógica de uma porta (ROGUES, 2017).	30
FIGURA 2.6 – Exemplo de arquitetura física (ROGUES, 2017).	31
FIGURA 2.7 – Exemplo de arquitetura física de uma porta (ROGUES, 2017).	32
FIGURA 2.8 – Interface inicial de <i>workflow</i> no ambiente do programa Capella. É possível ver as etapas sequenciais do Arcadia, da análise operacional até a estratégia de produção de produto.	33
FIGURA 3.1 – Imagem do RaInCube em sua configuração operacional, com antena e painéis solares abertos (JET PROPULSION LABORATORY,).	35
FIGURA 3.2 – Esquema sequenciado de abertura da antena do radar do RaInCube. A antena, de diâmetro 0,5 m, precisa se conformar antes a um cilindro de diâmetro 0,1 m. (RAINCUBE...,).	36
FIGURA 3.3 – Radar e <i>spacecraft bus</i> como componentes do RaInCube (IMKEN, 2017).	36
FIGURA 4.1 – Metodologia proposta para obtenção do <i>framework</i>	37
FIGURA 4.2 – Planilha de correlação SMAD-Arcadia.	40
FIGURA 5.1 – Framework obtido.	43
FIGURA 5.2 – Missão RaInCube (CERQUEIRA, 2020), adaptado pelo autor.	44
FIGURA 5.3 – Diagrama de definição das entidades operacionais.	45

FIGURA 5.4 – Diagrama de definição das capacidades operacionais - 1.	46
FIGURA 5.5 – Diagrama de definição das capacidades operacionais - 2.	47
FIGURA 5.6 – Diagrama de atores do sistema.	48
FIGURA 5.7 – Diagrama de estrutura do RaInCube.	50

Lista de Tabelas

TABELA 4.1 – Passos do SMAD a ser correlacionado ao ARCADIA.	41
TABELA 4.2 – SMAD x CAPELLA - Análise Operacional	41
TABELA 4.3 – SMAD x CAPELLA - Análise de Sistema	41
TABELA 4.4 – SMAD x CAPELLA - Arquitetura Lógica	42
TABELA 4.5 – SMAD x CAPELLA - Arquitetura Física	42
TABELA 4.6 – SMAD x CAPELLA - Arquitetura Física	42

Lista de Abreviaturas e Siglas

CEI	Centro Espacial do ITA
ConOps	Concept of Operations
EPBS	End-Product Breakdown Structure
ESTO	Earth Science Technology Office
ISS	International Space Station
ITA	Instituto Tecnológico de Aeronáutica
JPL	Jet Propulsion Laboratory
LA	Logical Architecture
MoE	Measure of Effectiveness
NASA	National Aeronautics and Space Administration
OA	Operational Analysis
PA	Physical Architecture
SA	System Analysis
SMAD	Space Mission Analysis and Design

Sumário

1	INTRODUÇÃO	17
1.1	Motivação	17
1.2	Objetivo	18
1.3	Organização do trabalho	19
1.3.1	Capítulo 1	19
1.3.2	Capítulo 2	19
1.3.3	Capítulo 3	19
1.3.4	Capítulo 4	19
1.3.5	Capítulo 5	19
2	REVISÃO BIBLIOGRÁFICA	20
2.1	Projeto e Análise de Missões Espaciais - SMAD	20
2.1.1	Etapa 1: Definição dos Objetivos da Missão	21
2.1.2	Etapa 2: Definição dos Players Principais	22
2.1.3	Etapa 3: Definição da Linha do Tempo de Missão	22
2.1.4	Etapa 4: Estimativa de Necessidades, Requisitos e Limites Quantitativos	23
2.1.5	Etapa 5: Definição de Arquiteturas Alternativas de Missão	23
2.1.6	Etapa 6: Definição de Conceitos Alternativos de Missão	24
2.1.7	Etapa 7: Definição dos Requisitos-Chave e <i>System Drivers</i>	24
2.1.8	Etapa 8: Condução de Análises de Performance e Trocas de Sistema	25
2.1.9	Etapa 9: Avaliação da Utilidade de Missão	25
2.1.10	Etapa 10: Definição do Conceito Fundamental de Missão e Arquitetura	26

2.1.11	Etapa 11: Revisão dos Requisitos Quantitativos e Limites	26
2.1.12	Etapa 12: Iterar e Explorar Outras Alternativas	26
2.1.13	Etapa 13: Definição de Requisitos de Sistema	26
2.1.14	Etapa 14: Alocação dos Requisitos aos Elementos do Sistema	27
2.2	<i>Architecture Analysis & Design Integrated Approach - Arcadia</i> .	27
2.2.1	<i>Operational analysis</i> - Análise operacional	27
2.2.2	<i>System needs analysis</i> - Análise de sistema	27
2.2.3	<i>Logical architecture</i> - Arquitetura lógica	28
2.2.4	<i>Physical architecture</i> - Arquitetura física	30
2.2.5	<i>Product building strategy</i> - Estratégia de produção de produto	32
2.3	Capella	32
3	RAINCUBE	34
4	PROPOSTA DE FRAMEWORK ARCADIA SMAD	37
4.1	Metodologia	37
4.1.1	Análise Operacional	37
4.1.2	Análise de Sistema	38
4.1.3	Arquitetura Lógica	39
4.1.4	Arquitetura física	39
4.1.5	EPBS	39
5	APLICAÇÃO DO FRAMEWORK NO ESTUDO DE CASO RAINCUBE	43
5.1	Análise Operacional - RaInCube	44
5.1.1	Definir entidades operacionais e capacidades	44
5.1.2	Definir atividades operacionais e descrever as interações	45
5.1.3	Alocar atividades operacionais para os atores e entidades envolvidas .	46
5.2	Análise de Sistema - RaInCube	47
5.2.1	Definir atores, missões e capacidades	47
5.2.2	Refinar funções do sistema, descrever trocas funcionais	47
5.2.3	Alocar funções do sistema	49

5.2.4	Definir interfaces e descrever canários de interface	49
6	CONCLUSÃO	51
	REFERÊNCIAS	52

1 Introdução

1.1 Motivação

O estudo de missões espaciais é tradicionalmente feito seguindo a metodologia proposta por Wertz em seu livro (WERTZ *et al.*, 2011), na qual a missão deve ser desmembrada em algumas partes sequenciais. Com o avanço tecnológico, novos métodos para descrever missões de engenharia de sistemas foram surgindo, como o proposto pela linguagem Arcadia, de engenharia de sistemas baseada em modelos.

A substituição do modelo SMAD pelo do Arcadia ainda não ocorreu, uma vez que as modelagens são distintas e representam a missão de pontos de vista diferentes. No entanto, seria desejável haver uma conexão entre ambos os métodos para que pudesse ser possível aproveitar o detalhamento e anos de estudo do primeiro método aplicado em ferramentas mais modernas e que permitem uma observação diferenciada das missões. Para viabilizar isso, foi proposta uma forma de fazer a conexão entre ambas as metodologias e, para fazer a validação, foi utilizada uma missão como caso de estudo, a de um satélite do tipo CubeSat, chamado RaInCube.

Os CubeSats são satélites de dimensões reduzidas e de construção modular que foram desenvolvidos a partir de 1999 no meio acadêmico como um contraponto aos satélites tradicionais (HELVAJIAN; JANSON, 2008), em geral excessivamente caros e complexos para uma ampla gama de orçamentos. Desde então, os CubeSats se popularizaram ao redor do globo, democratizando a pesquisa experimental espacial.

Neste contexto, o *Jet Propulsion Laboratory* (JPL), subordinado à NASA, realizou em 2016 uma pesquisa espacial intitulada de RaInCube: um cubesat desenvolvido para monitoramento meteorológico.

Tradicionalmente, o monitoramento de condições meteorológicas terrestres é realizado por satélites maiores. Estes são equipados com uma antena que capta sinais eletromagnéticos refletidos das formações chuvosas, por exemplo. Os grandes radares destes satélites usam frequências que variam de 1 a 5 GHz, ideais para grandes distâncias de cobertura.

O RaInCube se propôs a desafiar este paradigma tecnológico. Para que um CubeSat

faça monitoramento meteorológico, ele deve ser equipado de uma antena compatível com sua dimensão reduzida. Mas isso leva ao fato de que a antena será sensível a comprimentos menores de onda, especificamente 1 cm, da banda Ka (33-36 GHz). Foi necessária, então, nesta pesquisa conduzida pelo JPL, a validação dessa nova tecnologia. O conceito de operação final seria a presença de uma constelação de RaInCubes em órbita, que cobrisse em pequenos intervalos de tempo determinadas regiões de interesse.

Em 21 de maio de 2018, o RaInCube foi lançado pelo foguete Cygnus CRS OA-9E e encaminhado à Estação Espacial Internacional. A partir desta, ele foi lançado no espaço juntamente com outros CubeSats de diferentes pesquisas e finalidades. Em operação, o RaInCube pôde cumprir seu objetivo de detecção de nuvens, conforme divulgado em publicação da NASA (NASA, 2018).

Foi constatado que o desenvolvimento de pesquisas na área de CubeSats tornaria viável a diminuição dos custos para a confecção de satélites e a aplicação da engenharia de sistemas poderia auxiliar a antecipar problemas e erros enfrentados nos projetos futuros e já existentes.

A antecipação de erros e problemas no desenvolvimento, aliado a diminuição de custos proporcionado pelo modelo de satélite proposto, CubeSat, permitiria uma quantidade maior de instrumentos posicionados em órbita. Como primeiro impacto, espera-se haver maior quantidade de pontos de coleta de dados, auxiliando estudos posteriores.

1.2 Objetivo

O objetivo geral deste projeto é aplicar engenharia de sistemas baseada em modelos em um projeto de sistema aeroespacial - o RaInCube (*Radar In CubeSat*), cuja finalidade é o monitoramento meteorológico. Este objetivo geral pode ser fracionado nos seguintes objetivos específicos:

- Compreender e revisar o arcabouço teórico atual da engenharia de sistemas
- Interligar as metodologias do SMAD (*Space Mission Analysis and Design*) e do Arcadia (*Architecture Analysis & Design Integrated Approach*)
- Dominar o uso das funcionalidades pertinentes do *software* Capella
- Estabelecer uma ponte entre a teoria estudada e a prática em ambiente Capella
- Realizar o desdobramento sistêmico do RainCube no Capella utilizando os passos SMAD
- Criticar os resultados obtidos em Capella com a teoria já revisada

- Criar corpo de boas práticas para desdobrar missões espaciais utilizando o Capella
- Estabelecer conclusões e propor melhorias ao projeto

1.3 Organização do trabalho

O trabalho foi organizado em cinco capítulos e a conclusão, sendo eles:

1.3.1 Capítulo 1

O capítulo 1 contém a introdução do trabalho, onde são expostos a motivação e o objetivo do mesmo, a descrição do sistema e a formulação do problema com a nomenclatura utilizada, bem como a organização do trabalho.

1.3.2 Capítulo 2

O capítulo 2 consiste na revisão bibliográfica do trabalho. São expostos os fundamentos teóricos das metodologias utilizadas SMAD e Arcadia, além da ferramenta Capella.

1.3.3 Capítulo 3

O capítulo 3 expõe detalhes da missão do RaInCube, tal como foi desenvolvida e concluída com sucesso pelo JPL (NASA), que serão utilizados para validar o *framework* proposto.

1.3.4 Capítulo 4

No capítulo 4 é apresentada a metodologia utilizada para abordar o objetivo visado no trabalho, descrita através de métodos e procedimentos utilizados para alcançar a correlação desejada entre os métodos. Além disso, expõe detalhes da missão do RaInCube, que serão utilizados para validar o *framework* proposto. O capítulo 4 apresenta o framework obtido para correlacionar os métodos SMAD e Arcadia.

1.3.5 Capítulo 5

O capítulo 5 dispõe dos resultados obtidos aplicados ao estudo da missão do RaInCube, no qual foi utilizado o *software* Capella para a modelagem do sistema.

2 Revisão Bibliográfica

2.1 Projeto e Análise de Missões Espaciais - SMAD

O processo de desenvolvimento do SMAD, publicado por Wertz (WERTZ *et al.*, 2011), começa por definir o *Mission Statement*, enunciado da missão, que consiste em enunciar em termos amplos os objetivos e *constraints*, limites, principais da missão. É adequado ser o mais amplo e genérico possível neste enunciado, de modo a dar flexibilidade ao longo do desenvolvimento da missão. Sistemas espaciais são em geral custosos, e pode ser necessário sacrificar parâmetros e objetivos específicos ao longo do processo.

O SMAD é dividido sequencialmente em 14 etapas. Nota-se que o processo é iterativo, e a cada etapa concluída faz-se necessário checar concordância e coerência com etapas, definições e parâmetros anteriores. Frequentemente, o projeto depara-se com condições inesperadas que requerem adequação retroativa. Não apenas por consequência de situações inesperadas, a iteração deve ser feita também para otimizar constantemente os custos do projeto. A Fig. 2.1 representa um fluxograma-algoritmo desenvolvido por (CERQUEIRA, 2020), que esquematiza as 14 etapas do método SMAD.

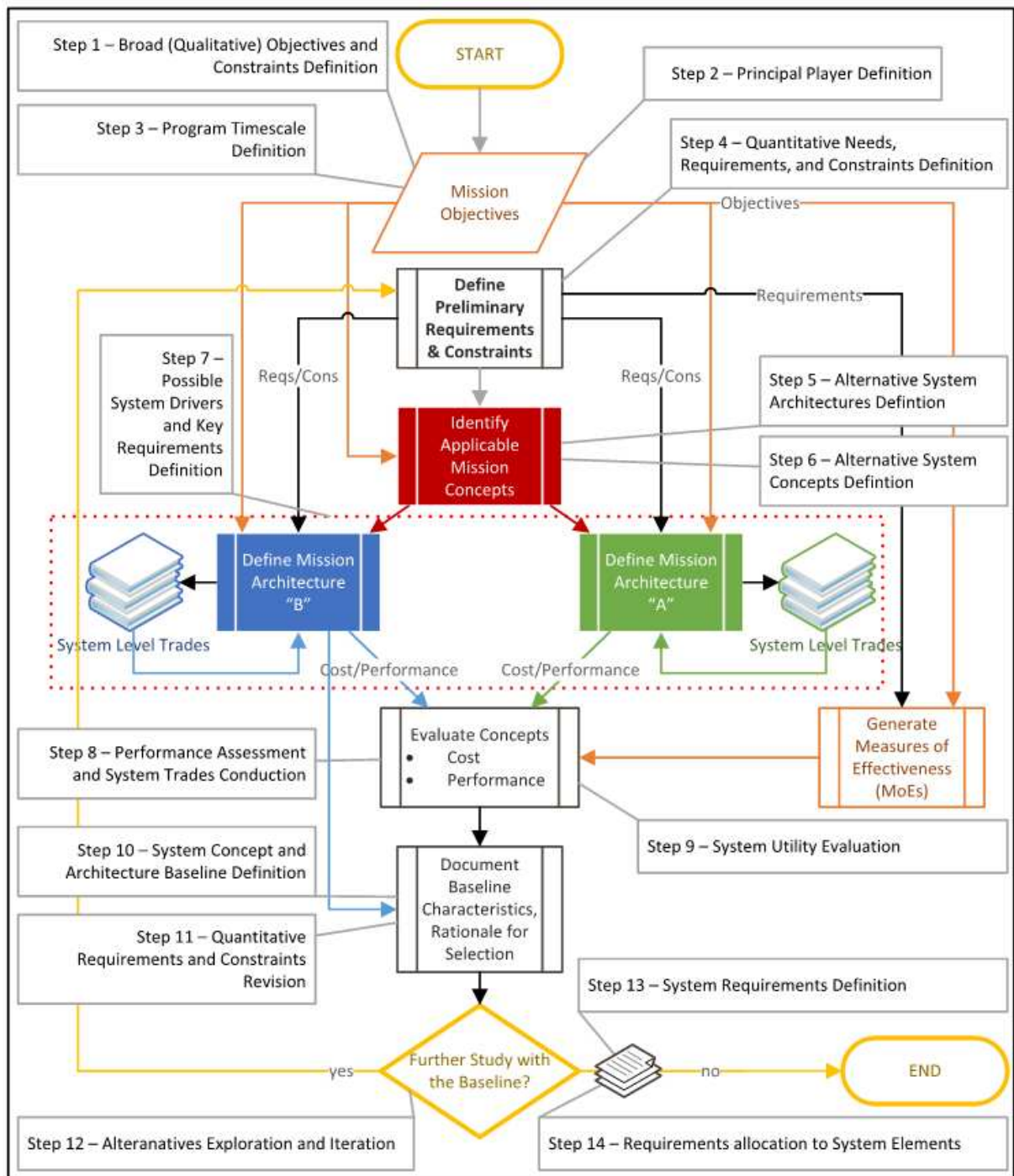


FIGURA 2.1 – Passos da metodologia SMAD para desenvolvimento de um projeto. (CERQUEIRA, 2020).

A seguir estão apresentadas resumidamente as 14 etapas do SMAD:

2.1.1 Etapa 1: Definição dos Objetivos da Missão

A primeira etapa do SMAD consiste em definir os objetivos da missão. Para que o sistema seja satisfatório, é preciso que os objetivos sejam atingidos e, nessa etapa do

processo, eles serão genéricos e qualitativos, para que não se limite a poucas possibilidades o projeto.

Dentro dos objetivos, existem os primários, que são inerentes ao funcionamento e operação do equipamento espacial, e são fundamentais, e também os secundários. Os secundários apesar de não guardarem relação com a operação do sistema, são objetivos da ordem política, social e cultural, que podem ser alcançados através da execução do equipamento principal da missão.

2.1.2 Etapa 2: Definição dos Players Principais

Os players principais são também chamados de *stakeholders*, e são impactados direta ou indiretamente pelo sistema desenvolvido. Tipicamente, há quatro principais *stakeholders*:

- Usuário final: é o *player* que irá utilizar e se beneficiar dos dados ou resultados finais gerados pelo sistema
- Operador: é o *player* que efetivamente irá operar o sistema desenvolvido
- Cliente primário: é o indivíduo, grupo ou entidade que de fato adquire o sistema
- Cliente secundário: o *player* ao qual o cliente primário tem que prestar contas. Por exemplo, a câmara legislativa (cliente secundário) que aprova um orçamento a uma agência espacial (cliente primário)

2.1.3 Etapa 3: Definição da Linha do Tempo de Missão

O ciclo de vida de uma missão espacial geralmente passa por quatro fases:

- Exploração de conceito: é a fase inicial de estudo, resultante da definição geral da missão espacial
- Desenvolvimento detalhado: é a fase de design propriamente dito, do que resultam definições detalhadas e validação de *hardware* e *software*
- Produção e lançamento: nesta fase se constrói o *hardware* e a ele integra-se o *software*
- Operação e suporte: é a fase final, em que se opera o sistema até o final de sua vida útil, usualmente caracterizado pela reentrada atmosférica

2.1.4 Etapa 4: Estimativa de Necessidades, Requisitos e Limites Quantitativos

Nesta etapa, transforma-se os objetivos gerais já definidos em necessidades, requisitos e limites preliminares mas tangíveis, ou seja, associados quantitativamente à missão. Este processo passa por três áreas gerais:

- Requisitos funcionais: como o sistema deve performar para atingir seus objetivos
- Requisitos operacionais: como o sistema opera e como seus usuários interagem com ele
- *Constraints* (Limites): estabelecem limites de custo e prazo, entre outros

Esta etapa é altamente iterativa e seus requisitos evoluem constantemente ao longo do projeto, tornando-se mais específicos e obedecendo a *trade-offs*.

2.1.5 Etapa 5: Definição de Arquiteturas Alternativas de Missão

Esta etapa consiste em unir ao conceito de missão um conjunto de elementos de missão. Toda arquitetura de missão é composta por oito elementos gerais, além do conceito de missão e o usuário final:

1. Sujeito: o que interage com o sistema; aquele que o sistema observa ou coleta dados
 2. *Payload* (Carga-paga): *hardware* e *software* que observa ou interage com o sujeito
 3. *Spacecraft bus*: os demais subsistemas da espaço-nave necessários ao suporte da carga-paga
 4. Segmento de solo: os equipamentos que se comunicam e controlam a espaço-nave
 5. Operações de missão: as pessoas e *software* que operam o sistema recorrentemente
 6. Arquitetura de comando, controle e comunicação: o modo como as diversas partes integrantes da missão comunicam-se entre si
 7. Órbita: o lugar geométrico espacial da espaço-nave durante sua operação
 8. Segmento de lançamento: o modo como a espaço-nave é lançada em órbita
- Conceito de missão: é a definição da forma como os elementos do sistema atingem juntos as necessidades do usuário final

Visando padronizar a construção da arquitetura de missão, podem ser seguidos passos pré-definidos, que facilitam o processo:

- Identificação de elementos da missão suscetíveis a *trade-offs*
- Identificação das principais opções de para cada elemento suscetível a *trade-offs*
- Procurar opções que funcionam bem conjuntamente ou não
- Procurar alternativas que podem influenciar a execução da operação

2.1.6 Etapa 6: Definição de Conceitos Alternativos de Missão

Esta etapa consiste em trabalhar com diferentes alternativas de elementos de missão se combinando de formas diversas e atingindo os objetivos da missão. É um processo que inerentemente envolve *trade-offs*.

O conceito de missão pode ser representado por quatro elementos gerais:

- Entrega de dados
- Controle e planejamento
- Arquitetura de comunicações
- Linha do tempo do programa

2.1.7 Etapa 7: Definição dos Requisitos-Chave e *System Drivers*

Os *system drivers* são os parâmetros principais da missão e influenciam a performance, o custo e risco do sistema - e daí a importância de identificá-los corretamente. Um passo-a-passo simplificado proposto por Wertz é:

1. Identificar a área de interesse
2. Definir parâmetros para mensurar a área de interesse
3. Confeccionar algoritmos para estimar valores para os parâmetros
4. Examinar os fatores do algoritmo, os ajustáveis e com maior impacto nos resultados serão os *system drivers*

Eventualmente o sistema conterà *hidden-drivers*, que consistem em variáveis implícitas no algoritmo que afetam mais de uma característica do sistema.

Concomitantemente, deve-se determinar os requisitos de sistema, que dominarão o design do sistema, seus custos e performance. Ao passo que não há um mecanismo definido para sua identificação, eles usualmente emergem por inspeção do conceito de missão. Alguns dos mais comuns são:

- Tempo de resposta
- Resolução
- Sensibilidade
- Potência de sinal

2.1.8 Etapa 8: Condução de Análises de Performance e Trocas de Sistema

Esta etapa passa por analisar e avaliar diversos *system trades*, com a finalidade de atingir os objetivos da missão, naturalmente atendendo a seus requisitos. Algumas trocas de sistema são simples e ordinárias, já outras podem ter consequências e implicações de sistema múltiplas, e outras ainda são abordagens únicas que exigem a reavaliação de todos os paradigmas assumidos até então.

2.1.9 Etapa 9: Avaliação da Utilidade de Missão

As definições de utilidade da missão podem ser tomadas pela quantificação da performance do sistema em termos de *design*, custo, risco e prazos. Essas definições quantitativas auxiliarão a tomar as decisões sobre dar prosseguimento ou não ao projeto.

Por meio de parâmetros definidos para cada missão, a utilidade deve ser definida de forma que seja possível estimar a viabilidade de performance e de efetividade. Enquanto o ponto de vista de performance busca estimar quão bem o sistema se adequa àquilo que ele foi proposto a cumprir, do ponto de vista qualitativo, o de efetividade utilizará probabilidade e estatística para mensurar, por exemplo, se o retorno esperado para a missão não seria inferior ao de outras, com tecnologia já existente e economicamente mais viáveis.

Os parâmetros de performance podem ser obtidos por testes experimentais, simulações computacionais ou mesmo análise teórica. As medidas de efetividade quantificam se o

o sistema cumpre a missão e os objetivos e devem ser descritas sucintamente, de modo a permitir a compreensão dos *stakeholders*.

2.1.10 Etapa 10: Definição do Conceito Fundamental de Missão e Arquitetura

Esta etapa pode ser realizada seguindo-se os seguintes passos:

1. Listar os objetivos e limites da missão
2. Identificar a comunidade envolvida, os usuários finais e seus amplos objetivos e abordagens
3. Determinar qual o critério mais importante da missão, dentre os seguintes: performance, custo, risco e prazo
4. Selecionar abordagens consistentes com o passo anterior para cada elemento da arquitetura de missão
5. Listar abordagens alternativas
6. Identificar os requisitos que comandam a performance, custo, risco e prazo
7. Para cada alternativa de arquitetura, realizar os *trade-offs* relevantes
8. Selecionar um conceito fundamental de missão e principais alternativas para aprofundar a análise destes
9. Documentar todo o processo e iterar

2.1.11 Etapa 11: Revisão dos Requisitos Quantitativos e Limites

Esta etapa está contida nos passos 6 e 7 da etapa anterior.

2.1.12 Etapa 12: Iterar e Explorar Outras Alternativas

Analogamente, esta etapa está contida no último passo, 9, da etapa 10.

2.1.13 Etapa 13: Definição de Requisitos de Sistema

Os requisitos subdividem-se em três tipos:

- Requisitos funcionais. Exemplos: de performance, cobertura e responsividade
- Requisitos operacionais. Exemplos: duração, sobrevivência (resistência a elementos naturais) e distribuição de dados (arquitetura de comunicação)
- *Constraints*. Exemplos: Custo, prazo e regulações políticas e ambientais

2.1.14 Etapa 14: Alocação dos Requisitos aos Elementos do Sistema

O último passo consiste em alocar os requisitos definidos a seus respectivos elementos do sistema. Por exemplo, o requisito de duração deve ser alocado pelo menos (mas não só necessariamente) ao elemento bateria, ao passo que o requisito de performance deve ser alocado pelo menos ao elemento carga-paga.

2.2 *Architecture Analysis & Design Integrated Approach - Arcadia*

O Arcadia é um método proposto para desenvolver a arquitetura de uma missão, utilizando como base sistemas baseados em modelos (ROQUES, 2017).

A implementação do método Arcadia consiste no desmembramento do sistema em algumas perspectivas, de sequência definida porém não deixando de ser iterativas:

2.2.1 *Operational analysis - Análise operacional*

Motivação: O que os usuários do sistema precisam realizar, o que os usuários do sistema utilizarão para cumprir o necessário.

Essa perspectiva enfoca os usuários operacionais. Esses são usuários que interagem diretamente com o sistema e suas interfaces. É preciso então identificar e definir seus objetivos, metas, limitações e condições. Em suma, sua missão e capacidades operacionais requeridas.

Não espera-se que se aprofunde no sistema em si, apenas que seja abordado superficialmente sobre o que se espera que ele deva cumprir. O sistema, seu *design* e lógica são abordados por outras perspectivas.

2.2.2 *System needs analysis - Análise de sistema*

Motivação: O que o sistema deve realizar aos seus usuários.

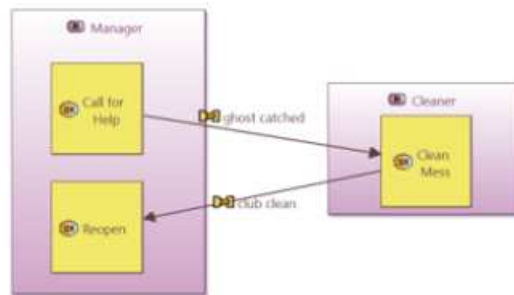


FIGURA 2.2 – Exemplo de diagrama operacional (ROGUES, 2017).

Nesta fase faz-se uma análise funcional externa, isto é, construída com base nos requisitos operacionais. Busca-se identificar funções de resposta ao usuários e demais sistemas interligados. Ou seja, descreve-se essas funções necessárias para que se desempenhe os serviços requeridos.

Alguns diagramas são usados nessa análise:

- *Contextual System Actors* - Diagrama das partes envolvidas com o sistema
- *Mission Capabilities* - Diagrama das capacidades que devem ser desempenhadas por cada parte
- *System Architecture* - Diagrama que relaciona atores, requisitos e funções
- *System Dataflow* - Esquematização do fluxo de informações no sistema
- *System Functional Breakdown* - Identificação de cada função, separadamente
- *Function Scenario* - Contextualização das funções com seu cenário de atuação
- *Exchange Scenario* - Esquematização das relações entre partes do sistema
- *Contextual External Interface* - Contextualização da atuação do sistema com agentes externos
- *Contextual Detailed Interface* - Detalhamento dos elementos do diagrama anterior
- *System Architecture with Requirements* - Arquitetura do sistema esquematizada juntamente com seus requisitos de funções

2.2.3 *Logical architecture* - Arquitetura lógica

A partir das necessidades e requisitos dispostos nas perspectivas anteriores, busca-se um *design* funcional de sistema. É necessário definir os fatores que impactam a infraestrut-

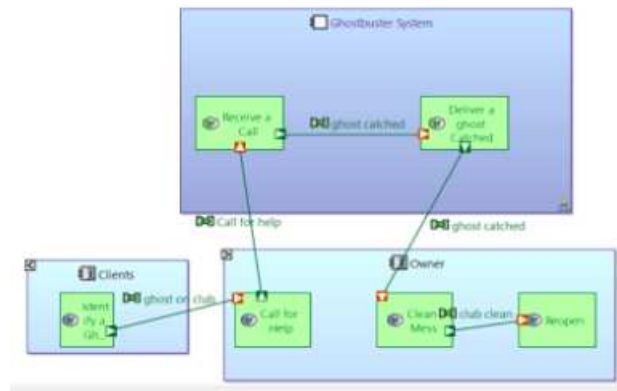


FIGURA 2.3 – Exemplo de arquitetura de sistema (ROGUES, 2017).

tura do sistema, os princípios que guiam seu funcionamento, alternativas de construção e finalmente selecionar a alternativa que oferece o melhor custo-benefício.

Devido à necessidade imposta pelas outras perspectivas, é possível a escolha inicial de *design* da solução, primeiramente pela análise funcional interna do sistema. A arquitetura lógica irá descrever as funções a serem executadas e montadas para a implementação das funções de serviços identificadas previamente. Mantém-se nessa etapa a identificação dos componentes que desempenham essas funções da solução, integrando as restrições não funcionais. O nível LA tem por objetivo:

- Identificar componentes lógicos dentro do sistema
- Identificar as subfunções necessárias para executar as funções do sistema escolhidas durante a fase anterior
- Determinar uma divisão em componentes lógicos para alocar essas subfunções internas

Para a definição da arquitetura do princípio lógico, deve-se:

- Definir fatores que impactam a arquitetura e a análise
- Definir os princípios que delimitam o comportamento do sistema
- Criar alternativas de estruturação de sistemas baseados em componentes
- Selecionar a alternativa que oferece o melhor compromisso

Os elementos presentes nesta arquitetura são:

- *Logical Component* - Elemento estrutural dentro do sistema, com portas para integrar com os outros componentes lógicos/atores externos

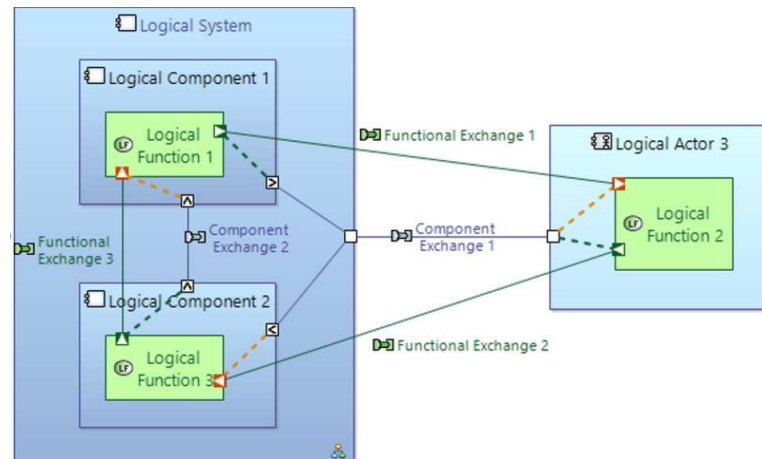


FIGURA 2.4 – Exemplo de arquitetura lógica (ROGUES, 2017).

- *Logical Actor* - Qualquer elemento externo ao sistema e que interage com ele
- *Logical Function* - Comportamento ou serviço fornecido por um componente lógico ou por um ator lógico
- *Functional Exchange* - Uma troca unidirecional de informações ou matéria entre duas funções lógicas, vinculando duas portas de função
- *Component Exchange* - Conexão entre os componentes lógicos/atores lógicos, permitindo a circulação das trocas funcionais

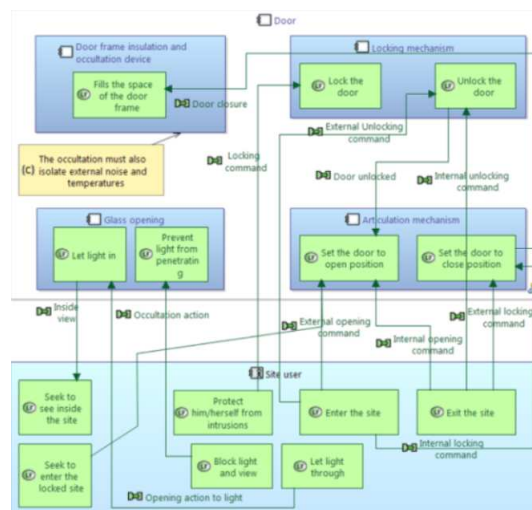


FIGURA 2.5 – Exemplo de arquitetura lógica de uma porta (ROGUES, 2017).

2.2.4 *Physical architecture* - Arquitetura física

Nessa etapa, define-se a arquitetura final do sistema, como ela deve ser completa e integrada. Adicionam-se as funções exigidas pela implementação e escolhas técnicas,

mostrando os componentes que desempenham essas funções.

Essa perspectiva apresenta objetivos semelhantes ao da arquitetura lógica, mas ela define a arquitetura finalizada do sistema, concluída e integrada, adicionando funções exigidas pela implementação e pelas escolhas técnicas e revela os componentes comportamentais que desempenham essas funções. Dessa forma, podem-se destacar como objetivos da PA como:

- Detalhar e finalizar o comportamento esperado do sistema
- Construir e racionalizar arquiteturas de sistemas possíveis
- Selecionar, terminar e justificar a arquitetura do sistema mantida

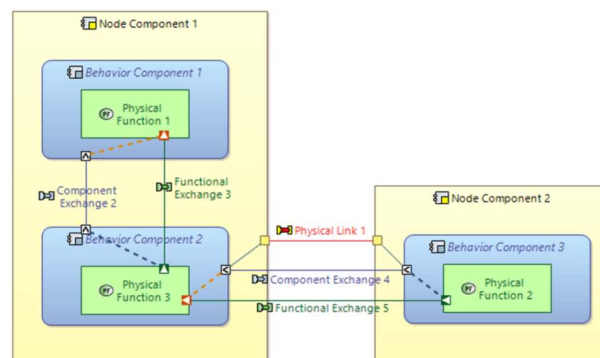


FIGURA 2.6 – Exemplo de arquitetura física (ROGUES, 2017).

Elementos presentes na arquitetura física incluem:

- *Node Physical Component* - Componente físico que fornece os recursos materiais necessários para componentes de comportamento
- *Physical Function* - Função desempenhada por componente ou ator físico
- *Physical Link* - Conexão de material não orientada entre os componentes de implementação. A troca de componentes permanece uma conexão entre os componentes de comportamento
- *Physical Port* - Porta não orientada que pertence a um componente de implementação
- *Behavior Physical Component* - Componente físico encarregado de funções físicas, executando parte do comportamento do sistema

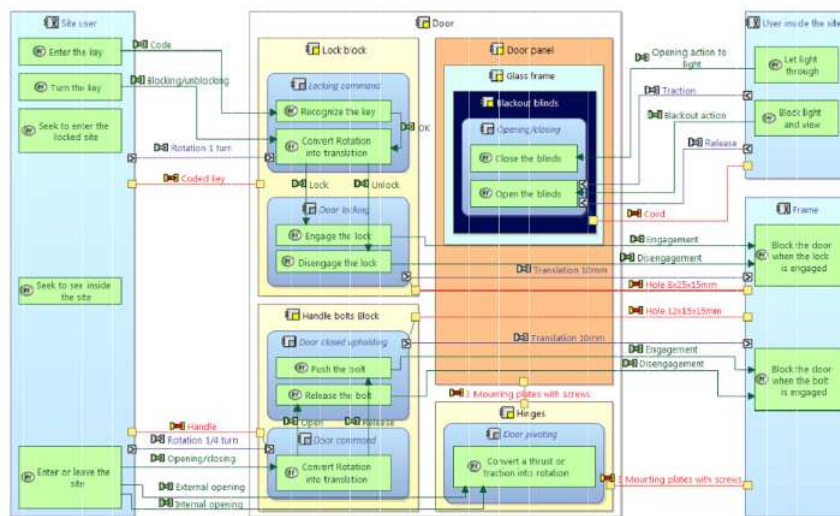


FIGURA 2.7 – Exemplo de arquitetura física de uma porta (ROGUES, 2017).

2.2.5 Product building strategy - Estratégia de produção de produto

Nessa perspectiva, as condições que cada componente deve cumprir são deduzidas a partir das restrições e escolhas do *design* da arquitetura. Através dela é possível definir o método de integração, verificação e validação do sistema.

2.3 Capella

O Capella é uma solução de *software* do tipo *open-source* (código aberto não-licenciado) para soluções de engenharia de sistemas baseada em modelos (MBSE - *Model-Based Systems Engineering*). Desta forma, o Capella é uma implementação em *software* da linguagem Arcadia (THALES ALENIA,). A Fig. 2.8 ilustra uma interface de uso do Capella.

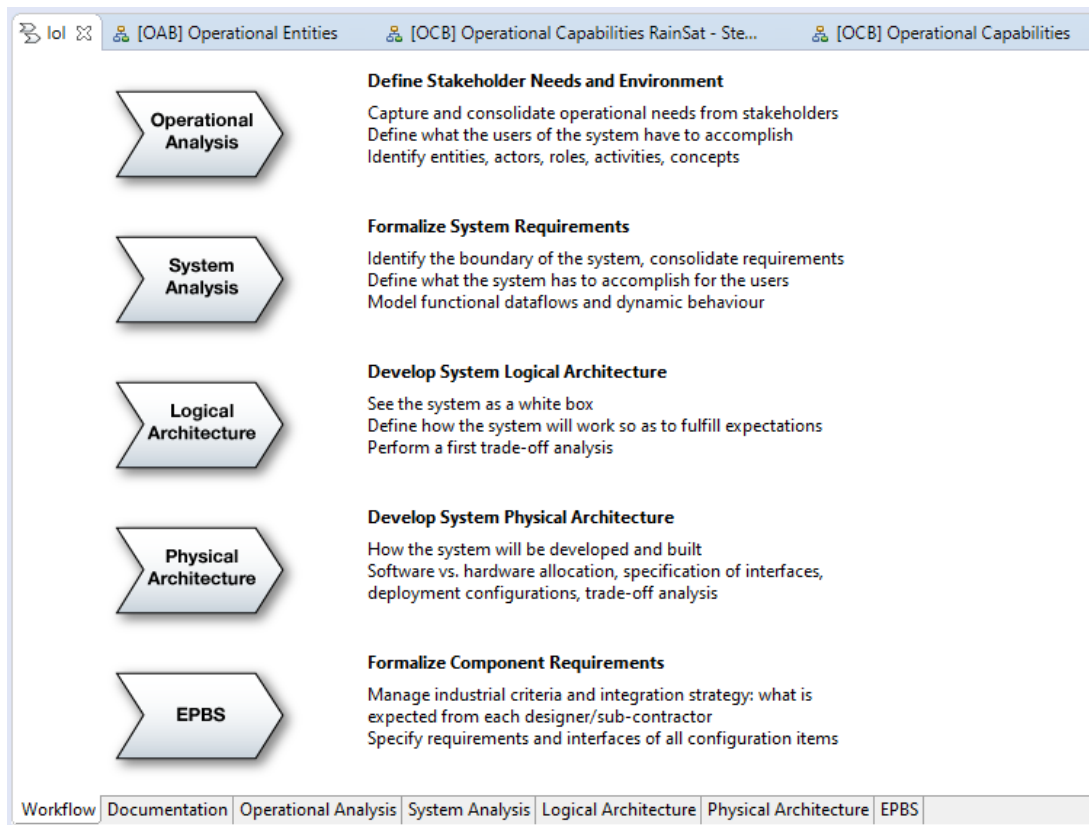


FIGURA 2.8 – Interface inicial de *workflow* no ambiente do programa Capella. É possível ver as etapas sequenciais do Arcadia, da análise operacional até a estratégia de produção de produto.

3 RaInCube

O projeto do RaInCube, desenvolvido pelo laboratório JPL (NASA), obteve financiamento do ESTO, *Earth Science Technology Office*, em fevereiro de 2016.

O custo e escala de produção de satélites com radar meteorológico tradicional é proibitivo frente uma abordagem de observações de alta frequência, que exige uma grande quantidade de satélites em órbita. A abordagem de CubeSats, por permitir observações mais frequentes, entre minutos e poucas horas, é crucial para a evolução dos modelos computacionais meteorológicos atuais.

Frente a esse desafio, a plataforma CubeSat se mostrou adequada para escalabilidade e viabilidade das constelações de satélites meteorológicos, por seu baixo custo e suas reduzidas dimensões.

O objetivo da missão do RaInCube foi então desenvolver, lançar e operar um radar meteorológico de banda Ka, de 35.75 GHz, integrado a uma plataforma CubeSat do tipo 6U, Fig. 4.1. A tecnologia do radar meteorológico miniaturizado de banda Ka já havia sido desenvolvida pelo JPL.

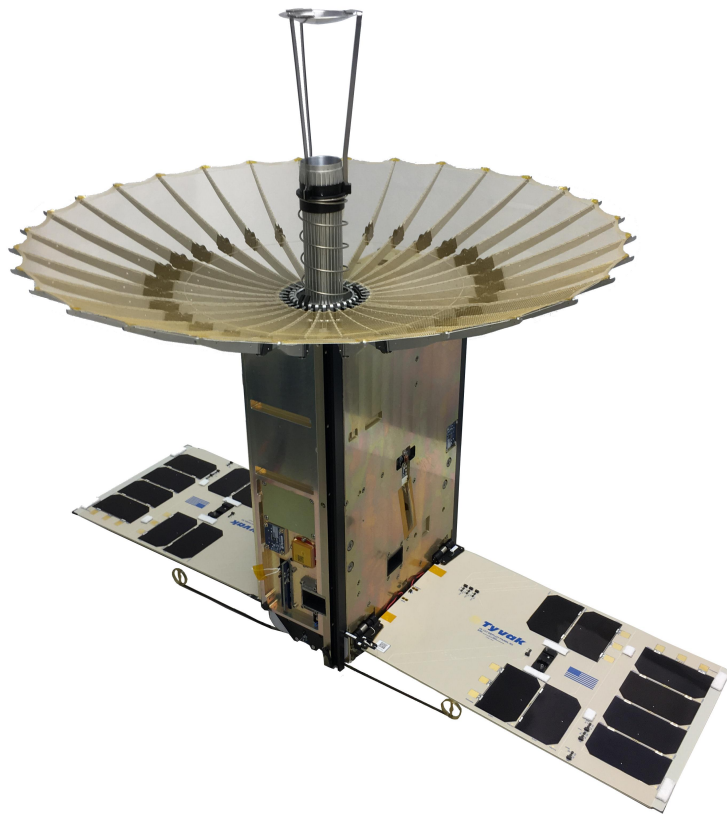


FIGURA 3.1 – Imagem do RaInCube em sua configuração operacional, com antena e painéis solares abertos (JET PROPULSION LABORATORY,).

A arquitetura do RaInCube pode ser dividida em duas partes principais: o *payload* e o *spacecraft bus*.

O *payload* consiste no radar, antena (Fig. 3.2) e sua eletrônica de suporte. O radar utilizado foi desenvolvido pelo JPL e é denominado miniKaAR-C, *miniaturized Ka-band Atmospheric Radar for CubeSats*. A antena é chamada KaRPDA, *Ka-band Radar Parabolic Deployable Antenna*.

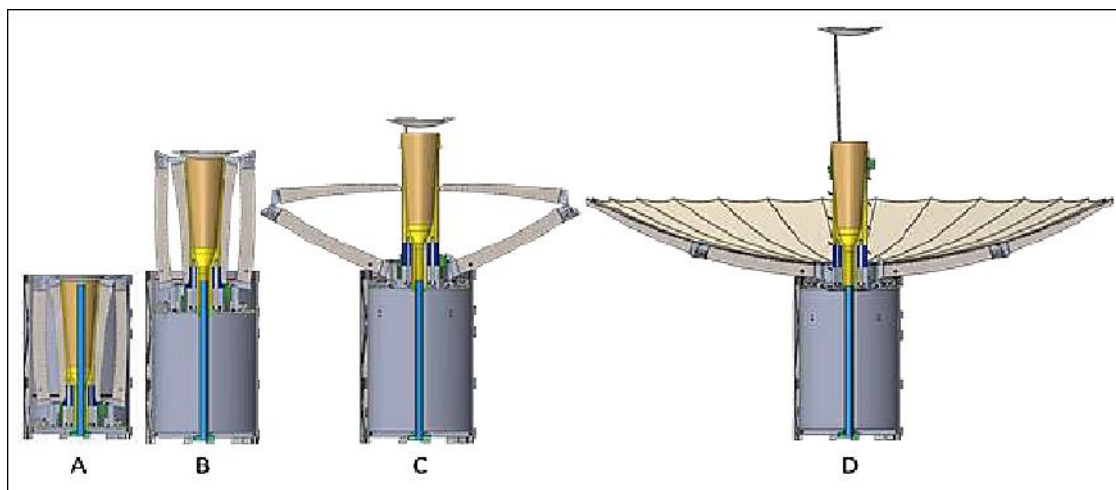


FIGURA 3.2 – Esquema sequenciado de abertura da antena do radar do RaInCube. A antena, de diâmetro 0,5 m, precisa se conformar antes a um cilindro de diâmetro 0,1 m. (RAINCUBE...,).

O *spacecraft bus* consiste nos subsistemas de suporte ao radar: energia, dados e interface térmica. A energia é capturada pelos painéis solares e armazenada na bateria de capacidade 70 W hr.

O radar e o *spacecraft bus* são arranjados espacialmente no CubeSat tal como ilustrado na Fig. 3.3. As faces 1Ux3U do RaInCube são utilizadas para alocação de eletrônicos, antenas GPS e sensores solares. Já as faces maiores 2Ux3U são utilizadas para controle termodinâmico - elas são revestidas de material Teflon.

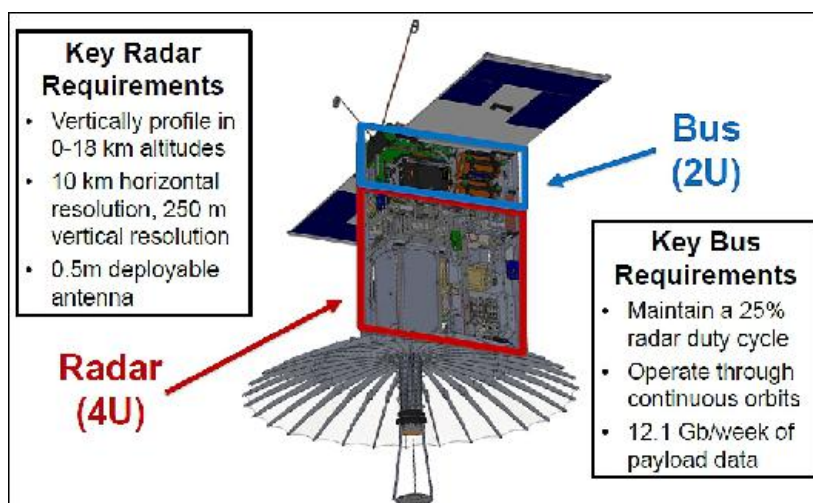


FIGURA 3.3 – Radar e *spacecraft bus* como componentes do RaInCube (IMKEN, 2017).

4 Proposta de Framework Arcadia SMAD

4.1 Metodologia

A fim de se obter uma boa correlação entre os procedimentos utilizados para modelar missões espaciais, SMAD, e a modelagem pelo Arcadia, os passos foram decompostos em uma planilha e inicialmente foram tomadas as premissas de que cada passo em uma metodologia deveria ter um correspondente na outra.

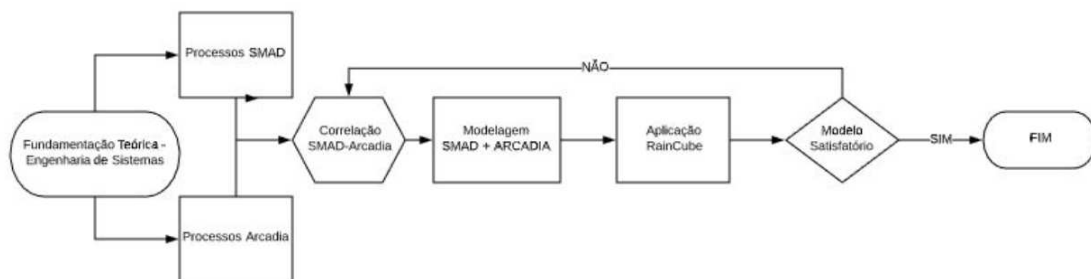


FIGURA 4.1 – Metodologia proposta para obtenção do *framework*.

Aos passos anteriormente definidos na metodologia do Arcadia, foram atribuídos códigos e breve descrição acerca do sub-nível explorado, visando facilitar a implementação e visualização do método através dos aspectos principais que devem ser observados em cada um. Além disso, as medidas necessárias para descrever uma missão pelo Arcadia também foram expandidas em sub-passos inerentes a cada um de seus cinco domínios.

4.1.1 Análise Operacional

1. OA.1 - Definir entidades operacionais e capacidades

Identificar o domínio operacional: quem são os atores e entidades, e quais são seus

objetivos.

2. **OA.2 - Definir atividades operacionais e descrever as interações**

Detalhar as atividades operacionais (quebrar em árvore), descrever as interações entre entidades e modelar os processos operacionais.

3. **OA.3 - Alocar atividades operacionais para os atores e entidades envolvidos -**

Atores e entidades operacionais são responsáveis por performar as atividades operacionais. Gerenciar alocações e deduzir os meios de comunicação entre entidades. Criar cenários a fim de ilustrar as interações entre atores e entidades operacionais.

4.1.2 Análise de Sistema

1. **SA.1 - Definir atores, missões e capacidades**

Identificar as fronteiras do sistema: quem são seus atores e quais são seus objetivos. As missões dão uma visão global sobre os principais objetivos/metast do sistema e seus usos apropriados.

As capacidades fornecem *insights* refinados sobre o aspecto operacional, diretamente relacionados aos requisitos dos clientes. As capacidades devem ser ilustradas com cenários.

2. **SA.2 - Refinar funções do sistema, descrever trocas funcionais**

Enriquecer e detalhar o *breakdown* de funções com novas funções de sistema. Descrever os fluxos de informação entre funções do sistema e identificar cadeias funcionais específicas.

3. **SA.3 - Alocar funções ao sistema e atores**

Gerenciar alocações usando um diagrama de arquitetura e deduzir trocas de componentes implementando trocas funcionais.

Criar cenários de *dataflows* para ilustrar as trocas funcionais entre o sistema e atores.

4. **SA.4 - Definir interfaces e descrever cenários de interface**

Detalhar as interfaces do sistema e também as dos atores, efetivamente desenhando a fronteira dos sistema.

Descrever cenários a fim de especificar a dinâmica comportamental do sistema.

Definir as sequências de interações e identificar as interfaces são atividades bem correlacionadas.

4.1.3 Arquitetura Lógica

1. **LA.1 - Refinar funções lógicas, descrever trocas funcionais**

Detalha a repartição funcional com novas funções lógicas, descreve os fluxos de dados entre as funções e identifica cadeias funcionais.

2. **LA.2 - Definir componentes lógicos e atores**

Usar um diagrama de arquitetura para descrever o sistema interno. Componentes lógicos interagem uns com os outros para atingir objetivos funcionais do sistema.

3. **LA.3 - Alocar funções lógicas a componentes lógicos**

Componentes lógicos implementam a lógica das funções. Através de um diagrama de arquitetura, pode-se gerir as atribuições e deduzir trocas de componentes de aplicação das trocas funcionais. Criar fluxos de dados para ilustrar trocas funcionais entre os componentes.

4. **LA.4 - Delegar interfaces de sistema e criar interfaces lógicas**

Delega cada interface do sistema para os componentes lógicos, cria interfaces internas entre os subcomponentes.

4.1.4 Arquitetura física

1. **PA.1 - Refinar funções físicas, descrever trocas funcionais**

Aprimora a repartição funcional com novas funções lógicas. Descreve os fluxos de dados entre as funções lógicas. Busca identificar cadeias funcionais.

2. **PA.2 - Definir componentes físicos e atores, gerenciar *deployments***

Definem os componentes físicos. Responsável por dar a implementação dos componentes lógicos, podendo o componente físico ser um nó ou um comportamento.

3. **PA.3 - Alocar funções físicas a componentes físicos**

Os componentes físicos comportamentais são responsáveis pela implementação das funções físicas. Ilustrar as trocas funcionais entre os componentes.

4. **PA.4 - Delegar interfaces lógicas e criar interfaces físicas**

Delegar cada interface lógica para um componente físico. Cria novas interfaces físicas entre os componentes.

4.1.5 EPBS

1. **EPBS.1 - Definir componentes CI (itens de configuração)**

Inicialização e atualização automática da Arquitetura EPBS de acordo com o modelo

de arquitetura física. Define itens de configuração adicionais, se necessário.

Utilizando essa nova denominação em códigos e as definições, os dados foram expostos numa planilha e a coluna referente aos passos do SMAD no ambiente Capella, coluna "Passos do SMAD", foi iterada até retornar uma boa representação, que viabilizasse a utilização do *framework* para a análise de um sistema espacial.

Passos do Capella		Qual dos passos do SMAD são cumpridos com essa atividade?	
Passos do Smad	Código	Análise Operacional	
1, 2	OA.1	Definir entidades operacionais e capacidades	<i>Identificar o domínio operacional: quem são os atores e entidades, e quais são seus objetivos.</i>
1, 2, 3	OA.2	Definir atividades operacionais e descrever as interações	<i>Detalhar as atividades operacionais (quebrar em árvore), descrever as interações entre entidades e modelar os processos operacionais.</i>
1, 2, 3	OA.3	Alocar atividades operacionais para os atores e entidades envolvidos	<i>Atores e entidades operacionais são responsáveis por realizar as atividades operacionais. Gerenciar alocações e deduzir os meios de comunicação entre entidades. Criar cenários a fim de ilustrar as interações entre atores e entidades operacionais.</i>
Análise de Sistema			
2, 4	SA. 1	Definir atores, missões e capacidades	<i>Identificar as fronteiras do sistema: quem são seus atores e quais são seus objetivos? As missões dão uma visão global sobre os principais objetivos/metabolismos do sistema e seus usos apropriados. As capacidades fornecem insights refinados sobre o aspecto operacional, diretamente relacionados aos requerimentos dos clientes. As capacidades deve ser ilustradas com cenários.</i>
4	SA. 2	Refinar funções do sistema, descrever trocas funcionais	<i>Enriquecer e detalhar o breakdown de funções com novas funções de sistema. Descrever os fluxos de informação entre funções do sistema e identificar cadeias funcionais específicas.</i>
5	SA. 3	Alocar funções ao sistema e atores	<i>Gerenciar alocações usando um diagrama de arquitetura e deduzir trocas de componentes implementando trocas funcionais. Criar cenários de dataflows para ilustrar as trocas funcionais entre o sistema e atores.</i>
6, 7, 8	SA. 4	Definir interfaces e descrever cenários de interface	<i>Detalhar as interfaces do sistema e também as dos atores, efetivamente desenhando a fronteira dos sistema. Descrever cenários a fim de especificar a dinâmica comportamental do sistema. Definir as sequências de interações e identificar as interfaces são atividades bem correlacionadas.</i>
Arquitetura Lógica			
8	LA. 1	Refinar funções lógicas, descrever trocas funcionais	<i>Detalha a repartição funcional com novas funções lógicas, descreve os fluxos de dados entre as funções e identifica cadeias funcionais.</i>
10,13	LA. 2	Definir componentes lógicos e atores	<i>Usa um diagrama de arquitetura para descrever o sistema interno. Componentes logicos interagem um com os outros para atingir objetivos funcionais do sistema.</i>
14	LA. 3	Alocar funções lógicas a componentes lógicos	<i>Componentes lógicos implementam a lógica das funções. Através de um diagrama de arquitetura, pode-se gerir as atribuições e deduzir trocas de componentes de aplicação das trocas funcionais. Criar fluxos de dados para ilustrar trocas funcionais entre os componentes.</i>
10	LA. 4	Delegar interfaces de sistema e criar interfaces lógicas	<i>Delega cada interface do sistema para os componentes lógicos, cria interfaces internas entre os subcomponentes.</i>
Arquitetura Física			
8,9,11	PA. 1	Refinar funções físicas, descrever trocas funcionais	<i>Aprimora a repartição funcional com novas funções lógicas. Descreve os fluxos de dados entre as funções lógicas. Busca identificar cadeias funcionais.</i>
13	PA. 2	Definir componentes físicos e atores, gerenciar "deployments"	<i>Definem os componentes físicos. Responsável por dar a implementação dos componentes lógicos, podendo o componente fisico ser um nó ou um "comportamento".</i>
14	PA. 3	Alocar funções físicas a componentes físicos	<i>Os componentes físicos comportamentais são responsáveis pela implementação das funções físicas. Ilustrar as trocas funcionais entre os componentes</i>
10	PA. 4	Delegar interfaces lógicas e criar interfaces físicas	<i>Delegar cada interface lógica para um componente fisico. Cria um novo interfaces físicas entre os componentes.</i>
EPBS			
5,9,12	EPBS. 1	Definir componentes CI (itens de configuração)	<i>Inicialização e atualização automática da Arquitetura EPBS de acordo com o modelo de arquitetura fisica. Define itens de configuração adicional se necessário.</i>

FIGURA 4.2 – Planilha de correlação SMAD-Arcadia.

De posse do resultado obtido, as variáveis relativas à missão do RaInCube serão introduzidas no *software* Capella para obter a visão geral da missão. Cabe destacar que,

apesar de o *framework* ter sido desenvolvido para todos os passos do SMAD e do Arcadia, as análises do RaInCube serão feitas somente até o passo 7 do SMAD, que corresponderá ao passo de análise de sistema na outra metodologia.

TABELA 4.1 – Passos do SMAD a ser correlacionado ao ARCADIA.

Passos do SMAD	Descrição
1	Objectives and Constraints Definition
2	Principal Player Definition
3	Program Timescale Definition
4	Quantitative Needs, Requirements and Constraints Definition
5	Alternative System Architectures Definition
6	Alternative System Concepts Definition
7	Possible System Drivers and Key Requirements Definition
8	Performance Assessment and System Trades Conduction
9	System Utility Evaluation
10	System Concept and Architecture Baseline Definition
11	Quantitative Requirements and Constraints Revision
12	Alternatives Exploration and Iteration
13	System Requirements Definition
14	Requirements Allocation to System Elements

TABELA 4.2 – SMAD x CAPELLA - Análise Operacional

Passo SMAD	Passo CAPELLA
1-2	OA.1
1-2-3	OA.2
1-2-3	OA.3

TABELA 4.3 – SMAD x CAPELLA - Análise de Sistema

Passo SMAD	Passo CAPELLA
2-4	SA.1
4	SA.2
5	SA.3
6-7-8	SA.4

TABELA 4.4 – SMAD x CAPELLA - Arquitetura Lógica

Passo SMAD	Passo CAPELLA
8	LA.1
10-13	LA.2
14	LA.3
10	LA.4

TABELA 4.5 – SMAD x CAPELLA - Arquitetura Física

Passo SMAD	Passo CAPELLA
8-9-11	PA.1
13	PA.2
14	PA.3
10	PA.4

TABELA 4.6 – SMAD x CAPELLA - Arquitetura Física

Passo SMAD	Passo CAPELLA
5-9-12	EPBS.1

5 Aplicação do Framework no Estudo de Caso RaInCube

Para descrever a missão através do *framework* obtido, representado na Fig. 5.1, foram coletados dados relativos à missão espacial, tais como os *players* envolvidos no processo de desenvolvimento do protótipo, os que fizeram o lançamento de uma base em solo e demais processos envolvidos para que fosse possível operar o satélite em órbita.

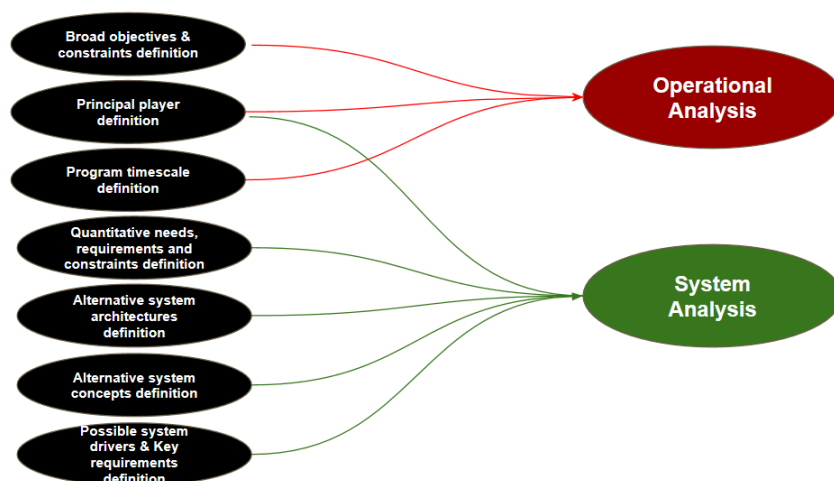


FIGURA 5.1 – Framework obtido.

Além disso, foi feita uma representação que abrange a troca de informações e entidades envolvidas, como as nuvens e o satélite, visando facilitar a atribuição das funções do sistema ao longo da realização do trabalho no Capella, uma vez que para isso é necessário possuir dados relativos à capacidade das entidades e das funções dos sistemas, conforme esquematizado na Fig 5.2.

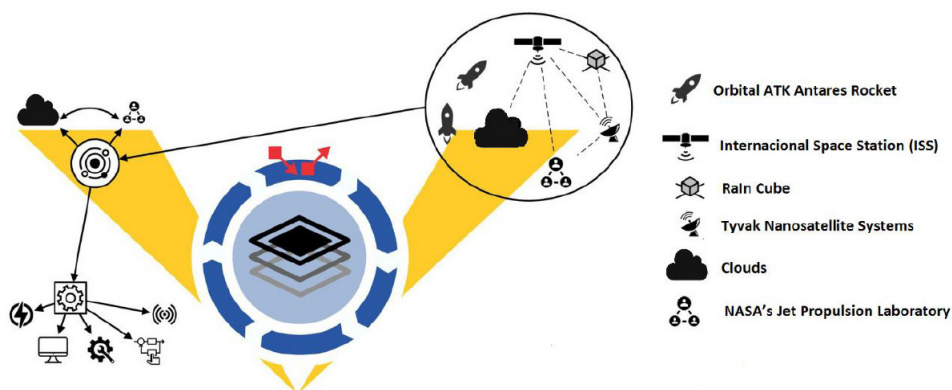


FIGURA 5.2 – Missão RaInCube (CERQUEIRA, 2020), adaptado pelo autor.

5.1 Análise Operacional - RaInCube

O início da descrição do sistema do RaInCube será através da análise operacional, seguindo os 3 passos definidos anteriormente.

5.1.1 Definir entidades operacionais e capacidades

Primeiramente, foram definidas as entidades envolvidas no projeto, devidamente associadas às suas atividades operacionais - Fig. 5.3. São elas:

- NASA → possui o RaInCube e monitora nuvens
- JPL (contido na entidade NASA) → possui o RaInCube e monitora nuvens
- Precipitação → possui nuvens
- Usuário final → opera o RaInCube
- Patrocinador → possui recurso financeiro

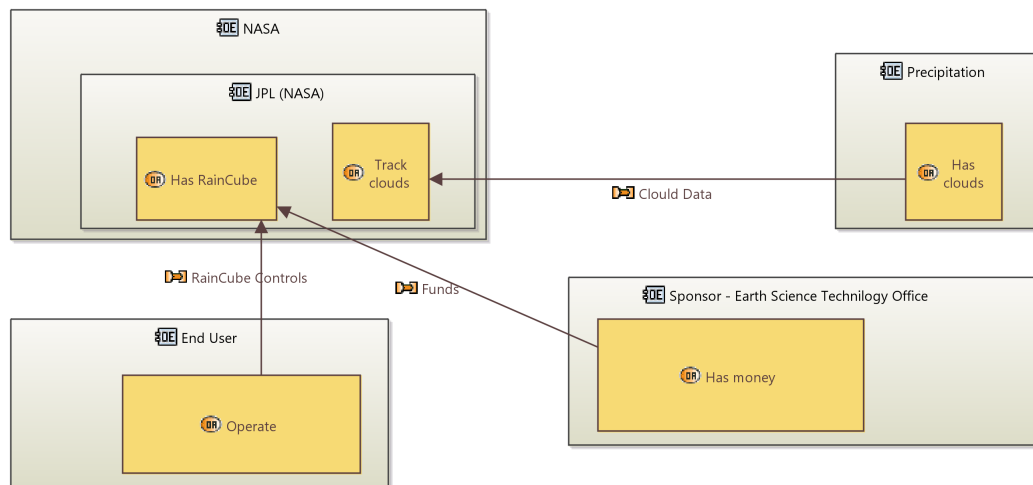


FIGURA 5.3 – Diagrama de definição das entidades operacionais.

5.1.2 Definir atividades operacionais e descrever as interações

São definidas também as trocas funcionais entre entidades, a saber:

- Dados meteorológicos são trocados da entidade precipitação para JPL (NASA).
- Controles do RaInCube são trocados do usuário final para o JPL (NASA).
- Fundos financeiros são trocados entre do patrocinador para o JPL (NASA).

Paralelamente à identificação das entidades operacionais, foi feita a expansão do *Mission Statement* em capacidades operacionais - Fig. 5.4. Ou seja, a identificação das capacidades operacionais envolvidas no cumprimento da missão.

As capacidades operacionais são:

- Captação de recursos financeiros
- Identificar e eliminar riscos técnicos
- Lançar um radar meteorológico de 35.75 GHz
- Operar um radar meteorológico de 35.75 GHz
- Demonstrar a viabilidade de uma carga radar numa plataforma do tipo CubeSat
- Desenvolver um radar meteorológico de 35.75 GHz
- Tornar futuras missões meteorológicas espaciais viáveis
- Validar a banda Ka

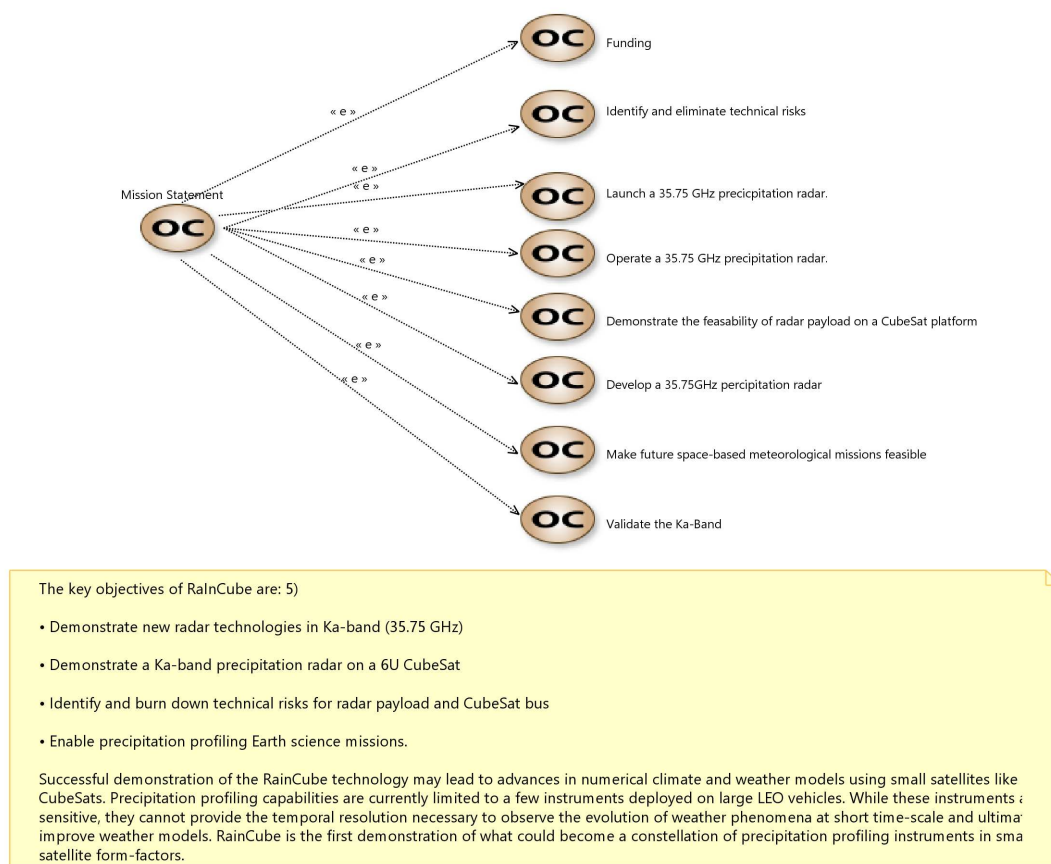


FIGURA 5.4 – Diagrama de definição das capacidades operacionais - 1.

5.1.3 Alocar atividades operacionais para os atores e entidades envolvidas

Em seguida, foram estabelecidas as relações entre entidades e capacidades operacionais, Fig. 5.5. É necessário atribuir cada capacidade operacional a pelo menos uma entidade responsável por desempenhá-la. As atribuições identificadas foram:

- Captação de recursos financeiros é responsabilidade do patrocinador.
- Identificar e eliminar riscos técnicos é atribuição do JPL (NASA).
- Operar um radar meteorológico de 35,75 GHz é atribuição do usuário final.
- Demonstrar a viabilidade de uma carga radar numa plataforma do tipo CubeSat é responsabilidade do JPL (NASA).
- Validar a banda Ka: estão diretamente envolvidas com as entidades de precipitação e JPL (NASA).
- Desenvolver um radar de precipitação de 35.75 GHz é atribuição do JPL (NASA).

- Lançar radar de precipitação de 35.75 GHz é atribuição do JPL (NASA).
- Tornar futuras missões meteorológicas espaciais viáveis é atribuição do JPL (NASA).

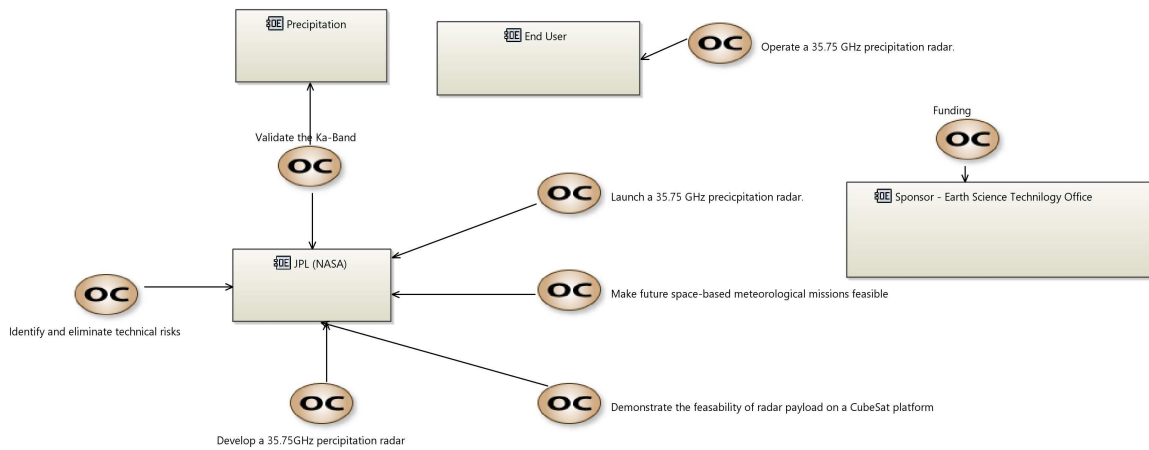


FIGURA 5.5 – Diagrama de definição das capacidades operacionais - 2.

5.2 Análise de Sistema - RaInCube

5.2.1 Definir atores, missões e capacidades

Os atores foram definidos tal como na Fig. 5.6. São eles:

- Precipitação
- ISS
- JPL (NASA)
- Orbital ATK Antares Rocket
- Tyvak Nanosatellite Systems

Os atores advém das entidades operacionais adaptadas ao ambiente de sistema.

5.2.2 Refinar funções do sistema, descrever trocas funcionais

Cabe destacar que o desenvolvimento do método previa a aplicação de processos iterativos. Dessa forma, o resultado final obtido já consta com todos os refinamentos feitos durante a confecção do trabalho e conseqüentemente era esperado que as funções de fato já estivessem otimizadas no primeiro passo.

As funções do sistema, e suas respectivas entidades, obtidas foram:

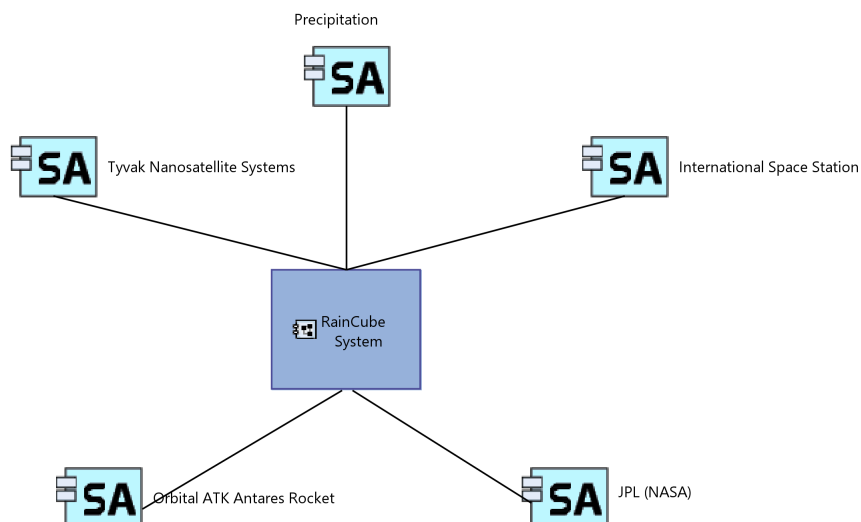


FIGURA 5.6 – Diagrama de atores do sistema.

- Possuir Nuvens → Agente de precipitação
- Rastrear Nuvens → JPL
- Lançar do solo → Orbital ATK Antares Rocket
- Lançar do espaço → ISS
- Operar o RaInCube → Tyvak Nanosatellite Systems
- Funcionar no espaço → RainCube
- Coletar dados → RaInCube
- Comunicar com o solo → RaInCube

Foi possível definir algumas trocas funcionais entre as funções acima, cabe destacar que duas funções podem desempenhar mais de uma troca funcional, e com sentidos opostos. Pode-se exemplificar essas trocas conforme abaixo junto às respectivas funções e o sentido da interação.

- Dados das nuvens: Possuir nuvens → Coletar dados
- Coleta de dados: Coletar dados → Rastrear nuvens
- Telemetria: Operar o RaInCube → Comunicar com o solo
- Telecomando: Comunicar com o solo → Operar o RaInCube
- *Mission Delivery*: Lançar do solo → Lançar do espaço
- *Mission Delivery*: Lançar do espaço → Funcionar no espaço

5.2.3 Alocar funções do sistema

Cada função do sistema é vinculada a um ator, e um ator pode possuir mais de uma função. Dessa forma, no ambiente do Capella cada função é posicionada dentro do espaço definido pelo ator de modo deixar evidente este vínculo.

Dessa forma, foram atribuídas as seguintes funções:

- Localizar nuvens - atribuída ao ator JPL (NASA)
- Possuir nuvens - atribuída ao ator precipitação
- Operar o RaInCube - atribuída ao ator Tyvak Nanosatellite Systems
- Lançar a partir do solo - atribuída ao ator Orbital ATK Antares Rocket
- Lançar a partir do espaço - atribuída ao ator ISS
- Coletar dados, comunicar com o solo e funcionar no espaço - atribuídas ao sistema RaInCube

Adicionalmente, são explicitadas as chamadas trocas funcionais, relações entre duas funções com atribuição de direção de fluxo. Por exemplo, a troca funcional “telecomando” flui da função comunicar com o solo para a função operar o RaInCube; e por sua vez, a troca funcional “telemetria” flui entre estas duas funções mas em sentido contrário. Todas as trocas funcionais são dispostas na Fig. 5.7, e são elas:

- Dados coletados: de coletar dados para localizar nuvens
- Dados de nuvens: de possuir nuvens para coletar dados
- Telecomandos: de comunicar com o solo para operar o RaInCube
- Telemetria: de operar o RaInCube para comunicar com o solo
- *Mission delivery 1*: de lançamento do solo para lançamento do espaço
- *Mission delivery 2*: de lançamento do espaço para funcionar no espaço

5.2.4 Definir interfaces e descrever canários de interface

Todas etapas anteriores culminam no diagrama de estrutura do RaInCube - Fig. 5.7. Nele, são efetivamente esquematizados os relacionamentos entre atores, o sistema RaInCube, as funções, as trocas funcionais e as trocas entre componentes, evidenciando suas interfaces.

São também definidos e alocados os requisitos:

- Enviar dados para o solo através de ondas de rádio: atribuído à função coletar dados e à troca funcional dados coletados
- Detectar nuvens utilizando banda Ka: atribuído ao sistema RaInCube
- Possuir ciclo de vida de 1 ano: atribuído ao sistema RaInCube
- Detectar perfis de precipitação entre 0 e 18 km acima da superfície terrestre: atribuído às funções coletar dados e localizar nuvens
- Refletir ondas eletromagnéticas da frequência Ka: atribuído à função possuir nuvens
- Ser funcional em uma órbita de 400 km de altitude: atribuído à função funcionar no espaço
- Ser lançado no veículo lançador Cygnus CRS-9 Antares-230 da empresa Orbital ATK: atribuído às funções lançamento a partir do solo e do espaço
- Entrega a uma altitude de 400 km: atribuído à troca funcional *mission delivery 2*
- Enviar/receber sinais UHF e banda S: atribuído à função comunicar com o solo e às trocas funcionais telecomando e telemetria
- Utilizar o *software* C2D2 (Tyvak): atribuído à função operar o RaInCube

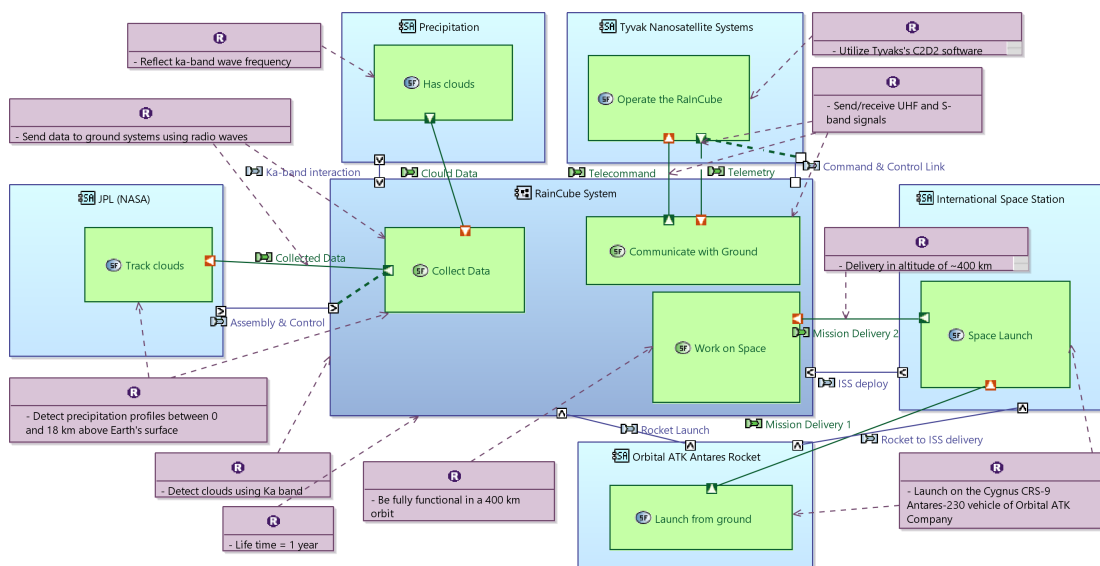


FIGURA 5.7 – Diagrama de estrutura do RaInCube.

6 Conclusão

O desdobramento utilizado para realizar o acoplamento das metodologias se mostrou eficaz, no sentido de que o modelo proposto para a correlação se adequou ao estudo de caso, o RaInCube, e o descreveu de maneira satisfatória, indicando a interação entre os elementos que compunham a missão sem que houvesse perda no detalhamento.

Ao realizar o estudo inicial, acreditava-se que deveria haver um passo correspondente no Arcadia para cada um dos 14 passos do SMAD, no entanto, ao longo do estudo foi identificado que não ocorreria essa unicidade, mas sim que poderiam acontecer bijeções dos elementos do SMAD para que pudesse haver a adequação ao ambiente do Capella, como pôde-se inferir no passo *"Principal players definition"*, que teve importância tanto no domínio da análise operacional como da análise de sistema. Caso a premissa tivesse sido mantida, o *framework* que se esperava obter se mostraria inconsistente, uma vez que não haveria elemento que desempenhasse as funções na análise de sistema. Esse resultado, ainda que preliminar, se mostra importante no quesito de que, durante a confecção de um estudo de sistemas baseado em modelo, diferentes entidades do ponto de vista SMAD podem desempenhar mais de uma função na linguagem Arcadia.

A engenharia de sistemas é um campo de estudo relativamente recente e em constante evolução. Em raras circunstâncias existe apenas uma solução para determinado problema - geralmente há inúmeras formas de se abordar uma mesma questão. Reconhecemos que o mesmo vale para o estudo realizado neste trabalho. O mérito deste trabalho, em nossa opinião, está em precisamente reconhecer mais de uma forma de analisar um problema, e buscar estabelecer uma ponte entre essas abordagens, com o objetivo de tornar a fronteira metodológica entre elas mais clara, e, assim, tornar mais completo e nuançado o estudo.

Deixa-se como sugestão para trabalhos posteriores a continuação do estabelecimento de correlação entre SMAD e Arcadia, que aqui foi concluída até o passo 7 do SMAD. Seria também enriquecedor para este domínio da engenharia de sistemas a aplicação em outros estudos de caso.

Referências

CERQUEIRA, C. S. **ARCADIA PRIMER**. São José Dos Campos: CEI, 2020. 264 p. Material de curso.

HELVAJIAN, H.; JANSON, S. **Small Satellites: Past, Present, and Future**. 1st. ed. El Segundo, CA: Aerospace Press, 2008.

IMKEN, T. **RainCube, a Ka-band precipitation radar in a 6U CubeSat**. 2017. Disponível em: <[http://mstl.atl.calpoly.edu/~workshop/archive/2017/Spring/Day1/Session 2/2_TravisImken.pdf](http://mstl.atl.calpoly.edu/~workshop/archive/2017/Spring/Day1/Session%202/2_TravisImken.pdf)>. Acesso em: 22 nov. 2020.

JET PROPULSION LABORATORY. **RainCube 6U CubeSat**. Disponível em: <<https://www.jpl.nasa.gov/spaceimages/details.php?id=PIA22457>>. Acesso em: 22 nov. 2020.

NASA. **RainCube Demonstrates Miniature Radar Technology to Measure Storms**. 2018. Disponível em: <<https://science.nasa.gov/technology/technology-highlights/raincube-demonstrates-miniature-radar-technology-to-measure-storms>>. Acesso em: 22 nov. 2020.

RAINCUBE (Radar In a CubeSat) - A Precipitation Profiling Mission. Disponível em: <<https://directory.eoportal.org/web/eoportal/satellite-missions/r/raincube>>. Acesso em: 22 nov. 2020.

ROGUES, P. **Systems Architecture Modeling with the Arcadia Method**. 1st. ed. Kidlington, Oxford: ISTE Press - Elsevier, 2017.

THALES ALENIA. **Open Source Solution for Model-Based Systems Engineering**. Disponível em: <<https://www.eclipse.org/capella/>>. Acesso em: 22 nov. 2020.

WERTZ, J. R.; EVERETT, D. F.; PUSCHELL, J. J. **Space Mission Engineering: The New SMAD**. 1st. ed. Hawthorne, CA: Microcosm Press, 2011.

FOLHA DE REGISTRO DO DOCUMENTO

¹ CLASSIFICAÇÃO/TIPO <p style="text-align: center;">TC</p>	² DATA <p style="text-align: center;">23 de novembro de 2020</p>	³ REGISTRO N° <p style="text-align: center;">DCTA/ITA/TC-051/2020</p>	⁴ N° DE PÁGINAS <p style="text-align: center;">51</p>
⁵ TÍTULO E SUBTÍTULO: Proposta de framework SMAD-Capella utilizando como estudo de caso o cubesat raincube.			
⁶ AUTOR(ES): Fernando Oliveira Lisboa, Lorenzo Pellizzaro Lima			
⁷ INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
⁸ PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: SMAD; Arcadia; RaInCube.			
⁹ PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Satélites artificiais; Estruturas (processamento de dados); Estudo de casos; Radar meteorológico; Engenharia aeroespacial.			
¹⁰ APRESENTAÇÃO: <p style="text-align: right;">X Nacional</p> <p>Internacional</p> ITA, São José dos Campos. Curso de Graduação em Engenharia Aeroespacial. Orientador: Prof. Dr. Luis Eduardo Vergueiro Loures da Costa; coorientador: Prof. Christopher Shneider Cerqueira. Publicado em 2020.			
¹¹ RESUMO: O trabalho tem por finalidade o estudo de caso de um satélite CubeSat, o RaInCube, visando validar um modelo que conecte duas metodologias distintas de engenharia de sistemas, buscando, para isso, extrair suas principais informações técnicas, de forma que seja possível replicar parcial ou completamente sua estrutura funcional e operacional.. O RaInCube é um satélite experimental do tipo CubeSat, satélite miniatura aplicado em pesquisas espaciais, que foi desenvolvido pelo Jet Propulsion Laboratory(JPL), um laboratório de pesquisa da NASA, em 2018. Esse protótipo teve por finalidade a validação de novas tecnologias que pudessem expandir a aplicabilidade de radares em pequenos satélites, com aplicações meteorológicas. A aplicação de radares meteorológicos em satélites do tipo CubeSat se mostrou potencialmente atraente porque esse tipo de satélite tem custo e complexidade consideravelmente menores em relação a modelos tradicionais. A engenharia de sistemas do RaInCube foi desenvolvida com a teoria de Wertz, a metodologia Arcadia e depois aplicada com a ferramenta Capella. Foi realizada uma integração entre a teoria SMAD e a metodologia Arcadia, que permitiu a implementação na ferramenta Capella, pois esta é baseada em Arcadia. Os resultados obtidos em Capella poderão ser utilizados como referência para modelagem de engenharia de sistemas de futuros projetos aeroespaciais desenvolvidos pelo CEI, Centro Espacial do ITA.			
¹² GRAU DE SIGILO: <p style="text-align: center;"> <input checked="" type="checkbox"/> OSTENSIVO <input type="checkbox"/> RESERVADO <input type="checkbox"/> SECRETO </p>			