Raytheon

# Model Based Test Architecture: An Abstraction to Improve Reality

**Raytheon Missile Systems**
George M Hollenbeck
Javier O Villafañe

October 3-6, 2016

# Agenda

Raytheon

- Introduction
- Test Architecture Overview
- Test Architecture from Engineers Point of View
- Test Architecture Using Model Based Elements
- Knowledge Points and Development Process
- Test Architecture Implementation
- Conclusion

# Introduction

- **Test Architects** – chief builder of test
- **Test Architecture (TA)**– the explicit definition of a test program that Test Architects defines the structure in terms of components, connections, and constraints of a product, process or element[1]
- **Model Based Test Architecture (MBTA)**– the explicit definition of a model that supports the Test Architect to define the Test Architecture

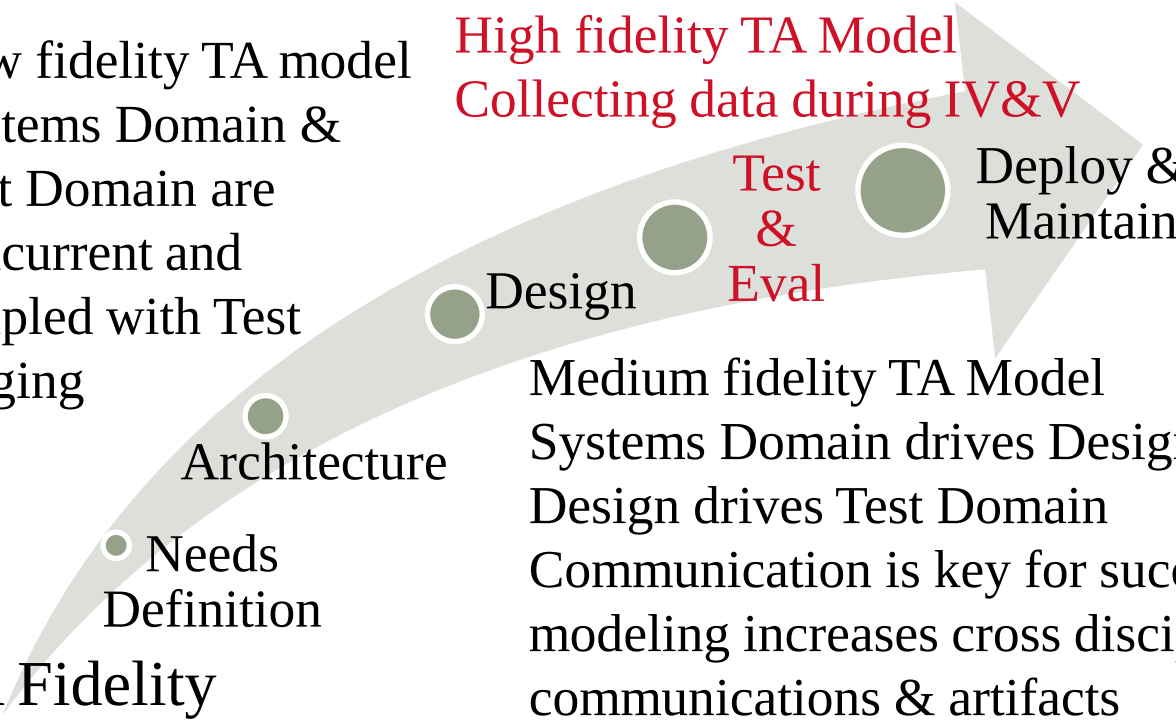

[1]Maier, Mark W. and Eberhardt Rechtin. 2009. *The Art of Systems Architecting - 3rd Edition*. NewYork: CRC Press.

# TA from Engineers POV

Raytheon

Low fidelity TA model
Systems Domain & Test Domain are concurrent and coupled with Test lagging

High fidelity TA Model
Collecting data during IV&V




Medium fidelity TA Model
Systems Domain drives Design
Design drives Test Domain
Communication is key for success, modeling increases cross discipline communications & artifacts

Modeling fidelity increases as the program matures

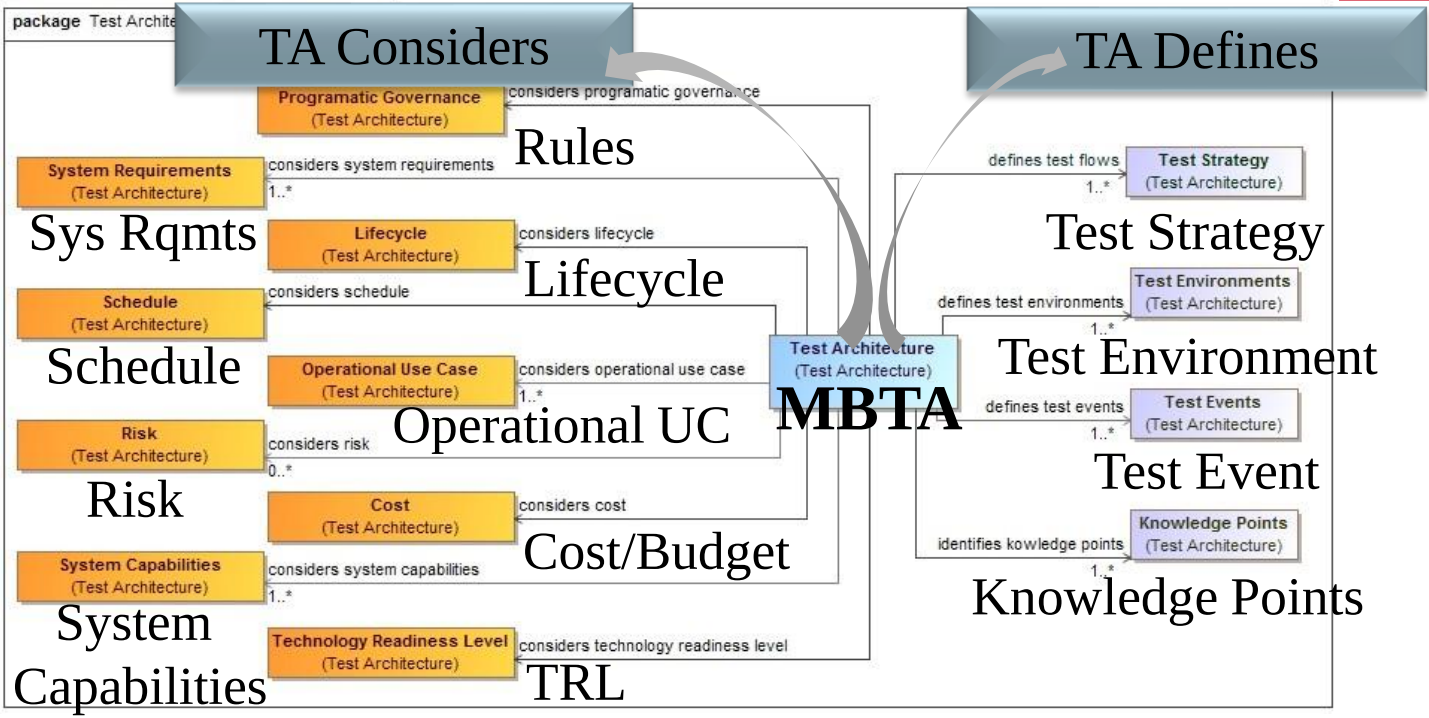
# TA from Engineers POV

Raytheon

- Systems Engineering & Test Architects working on modeling the Unit Under Test (UUT) and the Test Architecture (TA) concurrently will provide higher understanding with less defects by increasing multi-disciplined communication using a industry support language SysML and it associated tools
- Model Based Test Architecture (MBTA) enables managing data, generating artifacts and having information available to the entire team which is easier than traditional methods (distributed documents and sources)
- Architecture leads into design and design implementation becomes a UUT in its test environments (TEn)

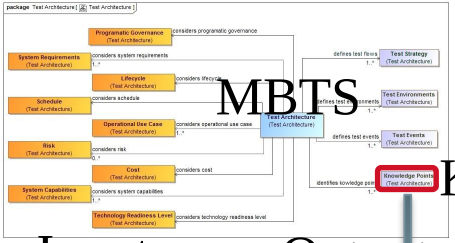
As the slide rule is to the calculator the MBTA will be to test

# Test Architecture Overview




Foundation of the Model Based Test Architecture (MBTA)

# Knowledge Points (KP)

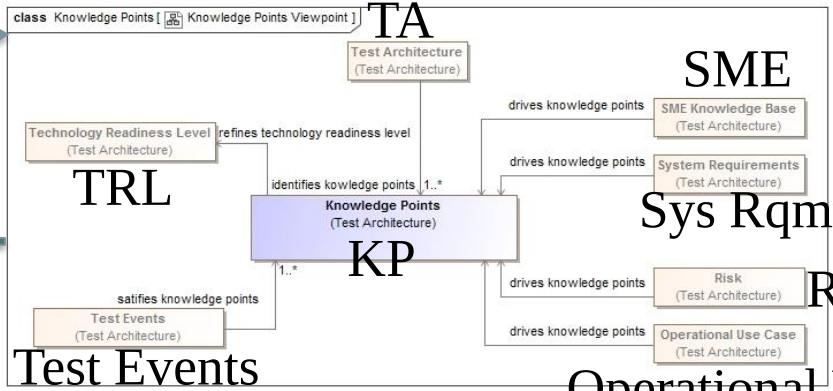



**Captures what activities or what we need to know about the system that allows it to meet the system intended purpose**

Knowledge Point Ontology

SysML Element

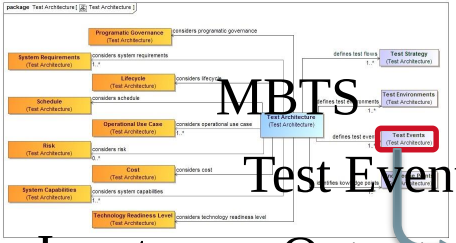
«Knowledge Point»
Knowledge Point #1




Maintaining the focus on what is required!

# Test Events

Raytheon




MBTS

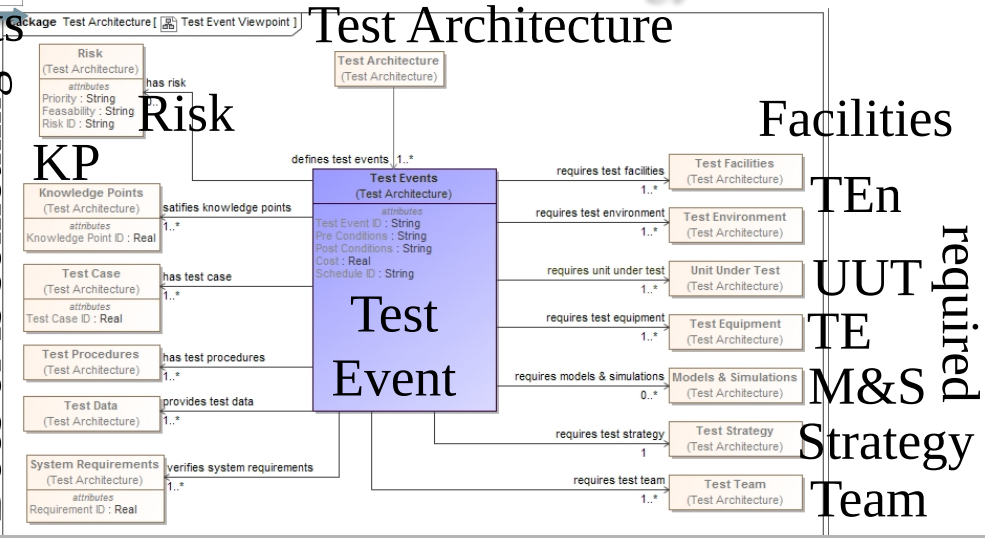
Test Events

**Primary purpose is to satisfy a KPs and to verify requirements when the data is produced and analyzed**

Test Events Ontology

Inputs

Outputs




Test Architecture

Risk

Facilities

KP

TEn

UUT

TE

M&S

Strategy

Team

SysML Element


Test Case

Procedure

Test Data

UUT Rqmts

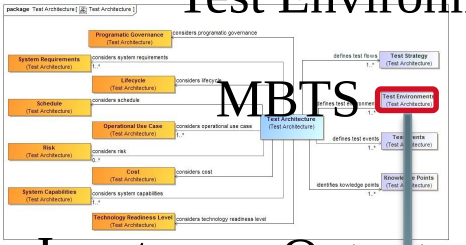
Does or is something required

Captures approach to obtain objective evidence!

# Test Environment (TEn)

Raytheon

Test Environment



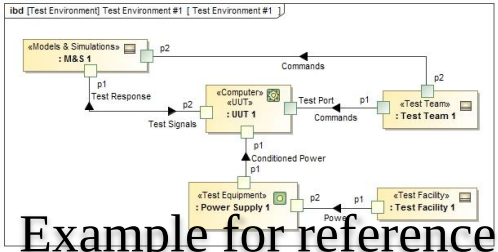
MBTS

Inputs

Outputs

SysML Element

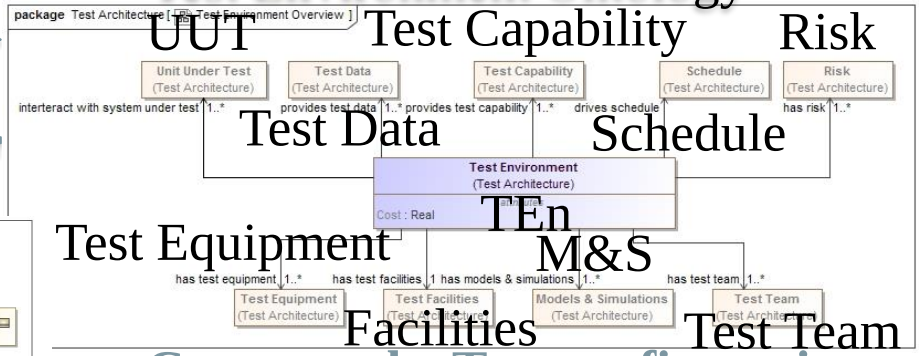

IBD Diagram




Example for reference

**Provides the capabilities awareness that enable Test Engineers to determine which assets and artifacts are required to execute a test event.**

Test Environment Ontology




UUT

Test Capability

Risk

Test Data

Schedule

TEn

Test Equipment

M&S

Facilities

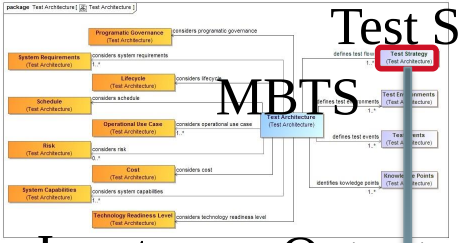
Test Team

**Captures the Test configuration and Information flow.**

Modeling the details of Test Environment

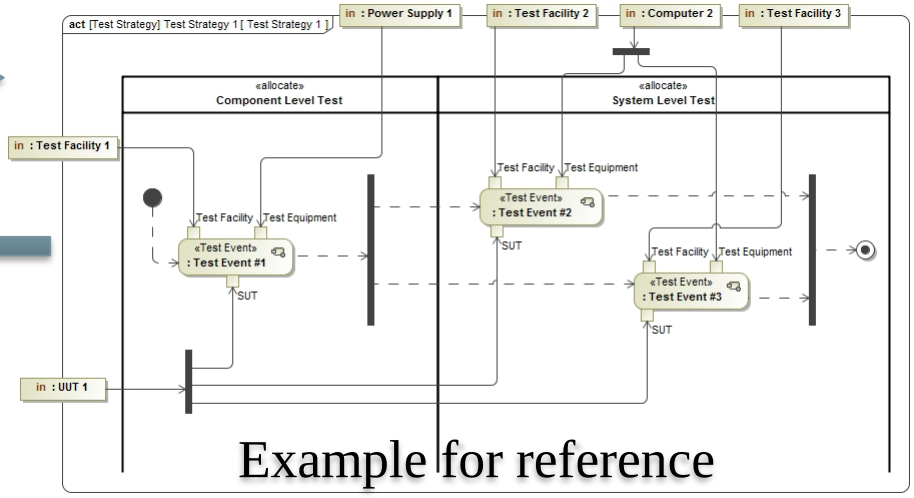
# Test Strategy

Raytheon

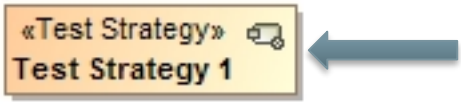



**Test strategy is capable of evolving as the system progresses throughout its lifecycle.**

SysML Activity Diagram




SysML Element




Modeling and generating a test strategy

# Test Architecture Modeling

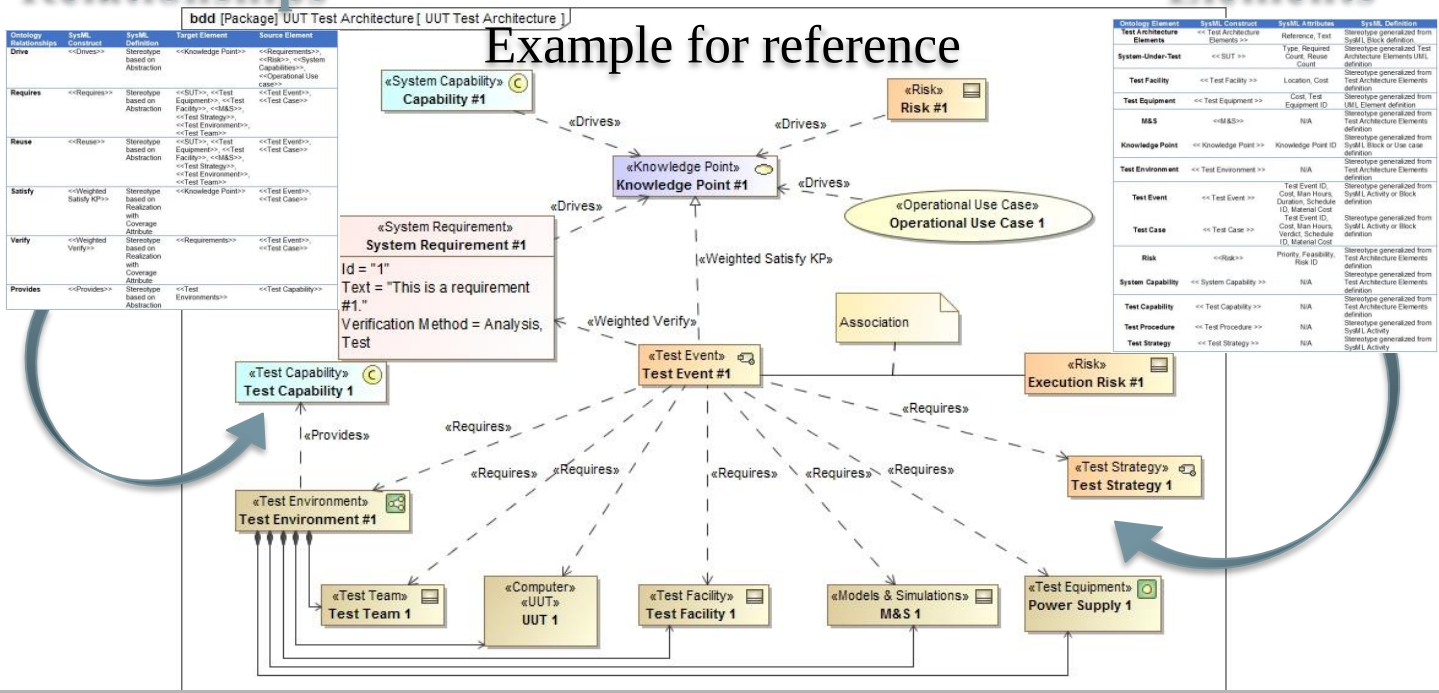
Raytheon

## SysML Test Architecture Model

Relationships

Elements

Example for reference

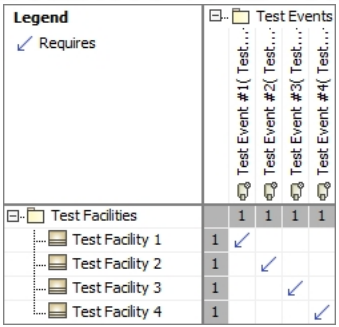



Test Architecture Taxonomy from the Test Event perspective

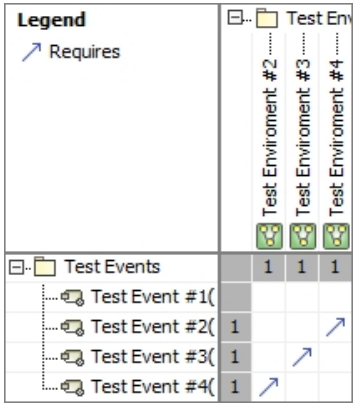
# Test Architecture perspectives

Raytheon

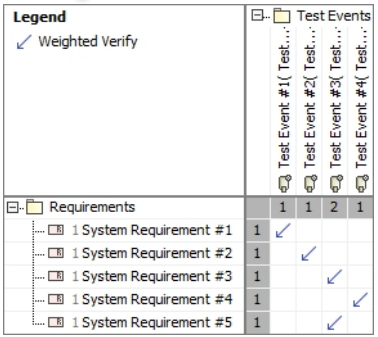
## Facilities to Events




## Events to TEn




## Req's to Events




## God's Eye View of the TA

| # | Id | Name | Text | Verification Method | Verified By | Test Facilities | Test Environment | Test Strategy | Test Equipment | Knowledge Point |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | System Requirement #1 | This is a requirement #1. | Analysis<br>Test | Test Event #1( Test F | Test Facility 1 | Test Environment #1 | Test Strategy 1( : Test Fa | Power Supply 1<br>Computer 2 | Knowledge Point #1<br>Knowledge Point #4 |
| 2 | 1 | System Requirement #2 | This is a requirement #2. | Analysis | Test Event #2( Test F | Test Facility 2 | Test Enviroment #4 | Test Strategy 2( : Test Fa | Computer 3 | Knowledge Point #2 |
| 3 | 1 | System Requirement #3 | This is a requirement #3. | Demonstration | Test Event #3( Test F | Test Facility 3 | Test Enviroment #3 | Test Strategy 3( : Test Fa | Power Supply 2 | Knowledge Point #3 |
| 4 | 1 | System Requirement #4 | This is a requirement #4. | Analysis<br>Test | Test Event #4( Test F | Test Facility 4 | Test Enviroment #2 | Test Strategy 4( : Test Fa | Power Supply 3 | Knowledge Point #4 |
| 5 | 1 | System Requirement #5 | This is a requirement #5. | Inspection | Test Event #3( Test F | Test Facility 3 | Test Enviroment #3 | Test Strategy 3( : Test Fa | Power Supply 2 | Knowledge Point #5 |

Tool generated artifacts and views

# Knowledge Points and Dev Process

Raytheon

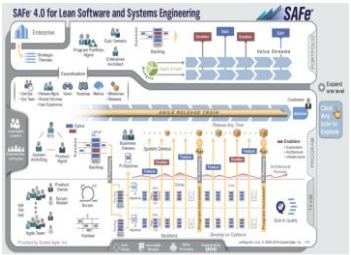
**Establishes the framework that enables the creation of actionable events and assist in the overall synchronization of the test effort towards a common goal.**

KP Identification

KP Definition

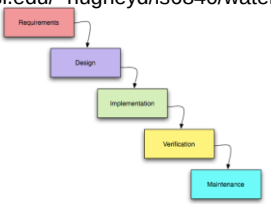
KP Generation

Source: http://www.scaledagileframework.com/




Agile Methodology

Source: http://www.umsl.edu/~hugheyd/is6840/waterfall.html




Waterfall Methodology

## Sources of Knowledge

- Subject Matter Experts
- Risk
- System Requirements

KPs work with different workflow methods

# Test Architecture Implementation

Raytheon

- There is a cost of creating a Test Architecture from the start of a program and maintaining that model through the lifecycle
- The infrastructure and work processes will be different with the net results also being improved quality and execution of a product satisfying the customers needs which should justify the return on the investment (ROI)
- Increases repeatability of the work flow processes via the use of a model and a defined ontology
- The test architecture will provide the entire team increased knowledge of the test strategy and test details using a modeling tool

ROI is justified by increased quality and effective test results

# Conclusion

Raytheon

- Increased cross-disciplined communications equates to decreased defects and reduces over all costs and schedule complications
- Modeling using the language of test with the same language modeling the design enables interoperability modeling
- Defining the test language and defining the test architecture is critical for ensuring a test strategy is understood and available
- Modeling the test program supports traditional and alternative work flow process such Agile and waterfall methods

# Questions?

Raytheon