

Requirements Engineering Management Training

(FAA/EASA/DOD/NASA)

TRAINING PROGRAM DESCRIPTION

Two and a half days of
focused **International** and **Online** Training
on **Requirements Engineering Management Training**

by

 Dr Ismail Cicek
GDS Engineering R&D, Inc.

Document Revision Date:
13 January 2023

Foreword

The requirements engineering literature is vast and applications vary widely even within a given industry. Therefore, the training is geared towards the real-time, embedded systems field and specifically to the avionics industry. Due to the rapidly increasing importance of software in these systems, emphasis is placed on applications that facilitate the transition from system to software requirements. The training includes the contents of the FAA Requirements Engineering Management Handbook (DOT/FAA/AR-08/32, June 2009), as well as examples and presentations created by the trainer's own experiences.

The training generally includes the studies and methods that should be done in order to create the requirements of a system in defense and aerospace projects within the scope of systems engineering.

The following link describes Dr Cicek's experiences in the topic with more detail:

<https://www.globaldynamicsystems.com/dr-i-cicek/>

Text based explanations in the standards are presented with tables, graphics, and videos for a clear and easier understanding

GDS TRAINING PROGRAMS
Systems Engineering
Requirements Management (MIL/FAA/NASA)
MIL-STD-810
RTCA-DO-160
MIL-STD-461
MIL-STD-704



Training material is provided before the class.

Purpose of the Training

The two and a half day of training presents a set of recommended practices on how to collect, write, validate, and organize requirements. It attempts to bring together the best ideas from several approaches, organize them into a coherent whole, and illustrate them with concrete examples that make their benefits clear. The presentations use some information from the FAA Requirements Engineering Handbook (DOT/FAA/AR-08/32), however, it is not limited to this handbook only.

The literature on requirements engineering is vast, and practices vary widely even within a particular industry. For this reason, the Handbook is targeted to the domain of real-time, embedded systems and specifically to the avionics industry. Due to the rapidly growing importance of software in these systems, it emphasizes practices that ease the transition from system to software requirements.

The main-level recommended practices are presented roughly in the order they would be performed on a program, but there is no requirement that this order must be strictly adhered to.

As with most processes, significant iteration between the different activities is expected as the requirements are refined. Rather than trying to specify a detailed process, the Handbook focuses on identifying what information is needed in a requirements specification and providing recommendations on how that information can be collected and organized.

Training Scope

The FAA Handbook addresses the real-time embedded systems field and specifically the avionics industry. It identifies a set of recommended practices in which core concepts can be applied individually but reinforce each other when applied as a whole. These applications allow developers to progress from an initial, high-level overview of a system to a detailed explanation of its behavior and performance requirements. Due to the increasing importance of software in avionics systems, these applications highlight techniques that facilitate the transition from system to software requirements.

Concrete examples are used throughout the Handbook to clarify concepts, but there are many other formats that can be used to achieve the same goals. It is expected that most organizations wishing to use these applications will want to modify them, perhaps significantly, to integrate them with their existing processes and tools.

The training is geared towards the real-time embedded systems field and specifically to the avionics industry. It identifies a set of recommended practices in which core concepts can be applied individually but reinforce each other when applied as a whole. These applications allow developers to progress from an initial, high-level overview of a system to a detailed explanation of its behavior and performance requirements. Due to the increasing importance of software in avionics systems, these applications highlight techniques that facilitate the transition from system to software requirements.

Read more details about this training content and schedule at the [GDS Website:](http://www.GlobalDynamicSystems.com)
<http://www.GlobalDynamicSystems.com>

Instructors

Training is provided by Dr Ismail Cicek with over 20 years of experience in the environmental qualification testing of products in accordance with, such as, MIL-STD-810H and RTCA-DO-160G. An Avionics Chief Engineer, a Certified Verification Engineer (FAA/EASA) with over 18 years of experience presents the EMI/EMC sections. Training is also assisted by our personnel experienced in design and environmental testing of military and aerospace equipment.

Training Schedule and Execution Type

- Online training using ZOOM.
- Led by live two instructors experienced in the field by both testing and lecturing.
- Two and a half days of focused online training schedule is typically as follows
 - 1st Day: 09:00 – 17:00 (Lunch Break between 12:30 and 13:30)
 - 2nd Day: 09:00 – 17:00 (Lunch Break between 12:30 and 13:30)
 - 3rd Day: 09:00 – 13:00
 - Time zone: Central European Time (CET)
- Attendees will receive a Training Certificate.
- Training includes knowledge check quizzes, a competition type, fun way or learning with prizes.

Visit [GDS Website](#) to check the calendar of scheduled training classes and for registration information. Or, send an email to us with your registration request: info@GlobalDynamicSystems.com.

Or call us to further discuss about your training needs. Ph: +1 (937) 912-1220 (USA) | Ph: +90 (537) 210-4068 (Turkey)

[Our training calendar](#) includes all open training classes, including RTCA-DO-160, MIL-STD-810, and MIL-STD-461.

Training Material

Requirements Engineering Management Handbook, DOT/FAA/AR-08/32, June 2009. This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. U.S. Department of Transportation Federal Aviation Administration. This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

The FAA Handbook is targeted to the domain of real-time, embedded systems and specifically to the avionics industry. It describes a set of recommended practices in which basic concepts can be practiced in isolation, but reinforce each other when practiced as a whole. These practices allow developers to progress from an initial, high-level overview of a system to a detailed description of its behavioral and performance requirements. Due to the growing importance of software in avionics systems, these practices emphasize techniques to ease the transition from system to software requirements.

Concrete examples are used throughout the Handbook to make the concepts clear, but there are many other formats that could be used to obtain the same objectives. It is expected that most organizations wanting to use these practices will want to modify them, perhaps significantly, to integrate them with their existing processes and tools.

Training is targeted to the domain of real-time, embedded systems and specifically to the avionics industry. It describes a set of recommended practices in which basic concepts can be practiced in isolation, but reinforce each other when practiced as a whole. These practices allow developers to progress from an initial, high-level overview of a system to a detailed description of its behavioral and performance requirements. Due to the growing importance of software in avionics systems, these practices emphasize techniques to ease the transition from system to software requirements.

The slides and other sharable course material will be shared with the registered students before the class using GOOGLE DRIVE.

- Registration includes all presentations and additional material shared before the class.

The RTCA-DO-160G standard must be purchased separately through RTCA, Inc. website at <https://www.rtca.org/standards/publications/>.

Training Contents (Detailed)

The training contents are synchronized with the contents of the FAA Requirements Engineering Management Handbook, which is as follows:

1. INTRODUCTION

- 1.1 Purpose
- 1.2 Background

2. RECOMMENDED PRACTICES

- 2.1 Developing the System Overview
 - 2.1.1 Develop System Overview Early
 - 2.1.2 Provide System Synopsis
 - 2.1.3 Identify System Contexts
 - 2.1.4 Use Context Diagrams
 - 2.1.5 Describe External Entities
 - 2.1.6 Capture Preliminary System Goals
 - 2.1.7 Maintain System Goal Information
- 2.2 Identify the System Boundary
 - 2.2.1 Identify the System Boundary Early
 - 2.2.2 Choose Environmental Variables
 - 2.2.3 Choose Controlled Variables
 - 2.2.4 Choose Monitored Variables
 - 2.2.5 Ensure Environmental Variables are Sufficiently Abstract
 - 2.2.6 Avoid Presentation Details in Environmental Variables
 - 2.2.7 Define All Physical Interfaces
- 2.3 Develop the Operational Concepts

- 2.3.1 Document Sunny Day System Behavior
- 2.3.2 Include How the System is Used in its Operating Environment
- 2.3.3 Employ the Use Case Goal as its Title
- 2.3.4 Trace Each Use Case to System Goals
- 2.3.5 Identify Primary Actor, Preconditions, and Postconditions
- 2.3.6 Ensure Each Use Case Describes a Dialogue
- 2.3.7 Link Use Case Steps to System Functions
- 2.3.8 Consolidate Repeated Actions Into a Single Use Case
- 2.3.9 Describe Exceptional Situations as Exception Cases
- 2.3.10 Describe Alternate Ways to Satisfy Postconditions as Alternate Courses
- 2.3.11 Use Names of External Entities or Environmental Variables
- 2.3.12 Avoid Operator Interface Details
- 2.3.13 Update the System Boundary
- 2.3.14 Assemble a Preliminary Set of System Functions
- 2.4 Identify the Environmental Assumptions
- 2.4.1 Define the Type, Range, Precision, and Units
- 2.4.2 Provide Rationale for the Assumptions
- 2.4.3 Organize Assumptions Constraining a Single Entity
- 2.4.4 Organize Assumptions Constraining Several Entities
- 2.4.5 Define a Status Attribute for Each Monitored Variable
- 2.4.6 Summary
- 2.5 Develop the Functional Architecture
- 2.5.1 Organize System Functions Into Related Groups
- 2.5.2 Use Data Flow Diagrams to Depict System Functions
- 2.5.3 Minimize Dependencies Between Functions
- 2.5.4 Define Internal Variables
- 2.5.5 Nest Functions and Data Dependencies for Large Specifications
- 2.5.6 Provide High-Level Requirements That are Really High Level
- 2.5.7 Do Not Incorporate Rationale Into the Requirements
- 2.6 Revise the Architecture to Meet Implementation Constraints
- 2.6.1 Modify the Architecture to Meet Implementation Constraints
- 2.6.2 Keep Final System Architecture Close to Ideal Functional Architecture
- 2.6.3 Revise the System Overview
- 2.6.4 Revise the Operational Concepts
- 2.6.5 Develop Exception Cases
- 2.6.6 Link Exception Cases to Use Cases

- 2.6.7 Revise the System Boundary
- 2.6.8 Document Changes to Environmental Assumptions
- 2.6.9 Revise Dependency Diagrams
- 2.6.10 Revise High-Level Requirements
- 2.7 Identify the System Modes
 - 2.7.1 Identify Major System Modes
 - 2.7.2 Define How System Transitions Between Modes
 - 2.7.3 Introduce Modes for Externally Visible Discontinuities
- 2.8 Develop the Detailed Behavior and Performance Requirements
 - 2.8.1 Specify the Behavior of Each Controlled Variable
 - 2.8.2 Specify the Requirement as a Condition and an Assigned Value
 - 2.8.3 Ensure That Detailed Requirements are Complete
 - 2.8.4 Ensure That Detailed Requirements are Consistent
 - 2.8.5 Ensure That Detailed Requirements are not Duplicated
 - 2.8.6 Organize the Requirements
 - 2.8.7 Define Acceptable Latency for Each Controlled Variable
 - 2.8.8 Define Acceptable Tolerance for Each Controlled Variable
 - 2.8.9 Do Not Define Latency and Tolerance for Internal Variables
 - 2.8.10 Alternative Ways to Specify Requirements
- 2.9 Define the Software Requirements
 - 2.9.1 Specify the Input Variables
 - 2.9.2 Specify the Accuracy of Each Input Variable
 - 2.9.3 Specify the Latency of Each Input Variable
 - 2.9.4 Specify IN' for Each Monitored Variable
 - 2.9.5 Specify the Status of Each Monitored Variable
 - 2.9.6 Flag Design Decisions as Derived Requirements
 - 2.9.7 Specify the Output Variables
 - 2.9.8 Specify the Latency of Each Output Variable
 - 2.9.9 Specify the Accuracy of Each Output Variable
 - 2.9.10 Specify OUT' for Each Controlled Variable
 - 2.9.11 Confirm Overall Latency and Accuracy
- 2.10 Allocate System Requirements to Subsystems
 - 2.10.1 Identify Subsystem Functions
 - 2.10.2 Duplicate Overlapping System to Subsystem Functions
 - 2.10.3 Develop a System Overview for Each Subsystem
 - 2.10.4 Identify the Subsystem Monitored and Controlled Variables

- 2.10.5 Create New Monitored and Controlled Variables
- 2.10.6 Specify the Subsystem Operational Concepts
- 2.10.7 Identify Subsystem Environmental Assumptions Shared With Parent System
- 2.10.8 Identify Environmental Assumptions of the New Monitored and Controlled Variables
- 2.10.9 Complete the Subsystem Requirements Specification
- 2.10.10 Ensure Latencies and Tolerances are Consistent
- 2.11 Provide Rationale
- 2.11.1 Provide Rationale to Explain why a Requirement Exists
- 2.11.2 Avoid Specifying Requirements in the Rationale
- 2.11.3 Provide Rationale When the Reason a Requirement is not Obvious
- 2.11.4 Provide Rationale for Environmental Assumptions
- 2.11.5 Provide Rationale for Values and Ranges
- 2.11.6 Keep Rationale Short and Relevant
- 2.11.7 Capture Rationale as Soon as Possible

3. SUMMARY

4. REFERENCES

APPENDICES

- A—Isolette Thermostat Example
- B—Flight Control System Example
- C—Flight Guidance System Example
- D—Autopilot Example

Our References

We have provided training courses to more than 100 companies and organizations and over 500 individual trainees so far.

