0. Overview

SAnToS Laboratory
Kansas State University
FAA Requirements Engineering Management Handbook (REMH)

- DOT/FAA/AR-08/32
- Written for the FAA by engineers at Rockwell Collins
  - David L. Lempia
  - Steven P. Miller
- DOT/FAA/AR-08/34
  - Companion document that provides background info and construction of a survey for collection requirements management issues from industrial engineers.

http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/research/
Purpose of the REMH

- Presents a set of recommended practices on how to:
  - Collect,
  - Write,
  - Validate, and
  - Organize requirements

- 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.
High-Level Goals of the REMH

- Targeted to the domain of real-time, embedded systems
  - Specifically to the avionics industry.
- Describes a set of recommended practices
  - Basic concepts can be practiced in isolation
  - Reinforce each other when practiced as a whole
- Enable progression from
  - An initial, high-level overview of a system, to
  - A detailed description of its behavioral and performance requirements.
Examples Used in REMH

Isolate – Thermostat for an infant incubator

“The purpose of the Isolette Thermostat is to maintain the air temperature of an Isolette within a desired range. It senses the Current Temperature of the Isolette and turns the Heat Source on and off to warm the air as needed. ...”

The Isolate example will be used as the primary running example in our lectures.
Examples Used in REMH

Flight Control System – Provides flight guidance and autopilot functionality for aircraft pilots

“The system being specified is a portion of an FCS. The FCS compares the measured Aircraft Attitude to a Reference Attitude and generates Flight Director (FD) Guidance commands that are displayed as visible cues, i.e., the FD, on the left and right Primary Flight Displays (PFD). ...”
Examples Used in REMH

Flight Guidance System – Illustrates how the FGS of the previous FCS example could be factored out into a separate subsystem spec can given to a subcontractor.

“The system being specified is a portion of an FCS. The FCS compares the measured Aircraft Attitude to a Reference Attitude and generates Flight Director (FD) Guidance commands that are displayed as visible cues, i.e., the FD, on the left and right Primary Flight Displays (PFD). ...”
Specific Goals of the REMH

- Determine methods that enable successful
  - Management,
  - Control,
  - Integration,
  - Verification, and
  - Validation of system and software requirements (potentially developed by multiple entities)
What is a “Good” Requirement?

- “Describes everything necessary to produce the correct system, nothing more.”
  - David Parnas (paraphrased)
- The balance that requirements need to achieve:
  - Specifying everything needed of the system to be built,
  - Not overconstraining the developers by venturing into design.
- The requirements should specify what the system will do – Not how the system will do it!
Good Requirements are a Progression

- The development of the requirements is a progression from:
  - A state in which relatively little is known about the system, to
  - One in which a great deal is known.
- The requirements engineering process needs to progress in a similar fashion
  - From informal practices early in requirements definition, to
  - More rigorous practices as the requirements are completed.
Requirements and Architecture

Develop Simultaneously

- It is usually impractical to state the detailed requirements of the system independent of the system architecture.

- Instead:
  1. High-level requirements are developed, then
  2. The next level of design is completed, then
  3. More detailed requirements are developed for each component.

- This process is continued until the necessary level of detail is reached.

- Requirements specification is interleaved with developing the system architecture.
Steps in the REMH

1. Develop the System Overview
2. Identify the System Boundary
3. Develop the Operational Concepts
4. Identify the Environmental Assumptions
5. Develop the Functional Architecture
6. Revise the Architecture to Meet Implementation Constraints
7. Identify System Modes
8. Develop the Detailed Behavior and Performance Requirements
9. Define the Software Requirements
10. Allocate System Requirements to Subsystems
11. Provide Rationale

Our lecture series will provide a lecture on each of the topics above!
Beware!

We don’t simply write requirements then move onto other steps in development (i.e., the classic “waterfall” model of development does not match reality). Instead, requirements are refined throughout the development process.

Many steps in development feedback and cause us to modify requirements. FAA REMH does a good job of acknowledging this, but it doesn’t fully acknowledge the details of various forms of safety analysis that we would probably do in practice.

http://www.learnaccessvba.com/application_development/waterfall_method.htm
Intertwined With Other Processes

Consider Leveson’s Outline of Safety Evaluation and Assessment Integrated with a System’s Engineering Process

Activities in Common with REMH

Safety evaluations occurring well after architecture development drive changes in requirements

(from “Safer World”, Ch. 10, Leveson)
Examples of other processes that proceed in parallel or are intertwined with requirements development

(from ARP 4761 – Avionics Recommended Practices)
Summary

Requirements development/management is a crucial part of any significant software development project.

- FAA REMH is not the only source on requirements management for embedded systems, but it does a good job pulling together best practices and illustrating them.
- Requirements development/management is a progression and even an iterative process.
- FAA REMH focuses on requirements development for safety critical systems.
For You To Do

- List the steps in the FAA REMH requirements management process.
Acknowledgements

- The material in this lecture is based almost entirely on