Embedded Systems Development Process

Prof. Hugo Vieira Neto
2020/1

Objective

- To study the main stages of the embedded systems development process :
  - V-Model development
  - Requirements engineering
  - System design
  - Implementation and tests
  - Useful UML diagrams

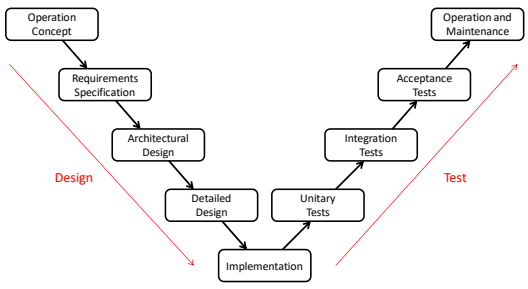
Development Stages

- Design phase
  1. Product design
  2. Requirements and systems engineering
  3. Architectural and detailed design
- Implementation phase
  4. Hardware development
  5. Software development
  6. Mechanics development

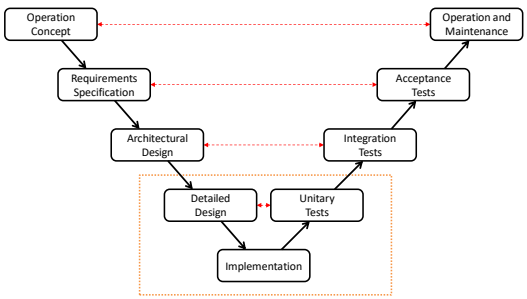
Etapas do Processo

- Test phase
  7. System integration
  8. Laboratory tests
  9. Field tests
- Production phase
  10. Product and production documentation
  11. Product packaging
  12. Product disposal

V-Model Development



V-Model Development



Requirements Engineering

- Input: requirements (informal, from customer, regulatory standards)
- Objective: to understand the problem
- Means: modeling, prototyping
- Output: requirements specification (formal, clear, precise, consistent, complete)
  - Functional: what the system does, how to use it
  - Non-functional: performance, robustness, development aspects
- **WHAT**

System Design

- Input: requirements specification
- Objective: to plan the solution
- Means: modeling, prototyping
- Output: design documentation (description of the planned solution)
  - Study of the hardware platform, available tools and libraries
  - System architecture design, detailed design of each block
- **HOW**

Implementation and Tests

- Input: design documentation
- Objective: to generate product
- Means: hardware assembly, software coding, integration, testing
- Output: production prototype, production documentation
- **ACTION**

Application Example

- GPS track viewer
  - The GPS has no display, just a standard RS-232 serial interface
  - The GPS is capable of collecting point sequences (latitude-longitude)
  - The GPS is able to transfer point sequences in GPX format
  - A device is desired to connect to the GPS and visualize the track at different scales

Device Functionalities

1. Data transfer in GPX format
2. Map and track visualization
3. Scale selection
4. Configuration

Problem Domain

- Georeferencing
- Latitude and longitude
- Distance computation
- Visualization scales
- GPX format specifications

Interfaces

- Physical:
  - RS-232 interface: voltage levels and connector
  - Color display with 1024 x 768 pixels
  - Buttons for human-machine interface
- Logical:
  - GPX format

Functional Requirements (1)

- FR1: The system should present a splash screen
  - FR1.1: The splash screen must display the device name
  - FR1.2: The splash screen must display the manufacturer brand
  - FR1.3: The splash screen must display the message “Press <OK> to proceed”

Functional Requirements (2)

- FR2: The system must be initialized using the “OK” button
  - FR2.1: The system must clear the splash screen and prepare to receive messages via the RS-232 serial interface
- FR3: The system must handle the reception of messages in GPX format
- FR4: The system must convert geographic coordinates to Cartesian coordinates

Functional Requirements (3)

- FR5: The system must plot on the screen the portion of the map that encompasses the coordinates of the track points
- FR6: The system must plot on the screen straight lines between the coordinates of the track points over the map
- FR7: The system must allow scaling using the “+” and “-” buttons
- ...

Non-functional Requirements

- NFR1: The system hardware must be based on the TM4C1294 microcontroller
- NFR2: The system development environment must be IAR EWARM V8
- NFR3: The system must be able to plot a minimum of 100 track points per second on the screen
- ...

Platform Study

- What is the plataform?
  - ARM Cortex-M4F core
  - Internal peripherals on the device
  - External peripherals available on the kit
  - Existing libraries for the chosen hardware
- Which part of the problem is already solved?
  - Graphic library for displaying points, lines, etc. on the display

Architectural Design

- Functional architecture (abstract):
  - Block diagram
  - Each block represents a system function
- Physical architecture:
  - Hardware block: functional unit
  - Software block: thread, function, structure
    - Class/object diagrams (UML) with active class/object notation are used

Detailed Design

- Hardware:
  - Schematic diagram for each functional unit
- Software:
  - Dynamic models for each thread or function
    - Statechart diagrams (UML)
    - Activity diagrams (UML)

Software Design: UML

	Estructural (Static)	Behavioral (Dynamic)
High level (Architectural)	• **Class/Object Diagrams (general)**	• Statechart diagrams • Sequence diagrams
Low level (Detailed)	• Class/Object Diagrams (each thread)	• **Statechart Diagrams** • **Activity diagrams**

Extra-class Activity

- Review the specifications for Lab Work 1 given by the professor and rewrite them in the form of functional and non-functional requirements
- When writing the requirements, have in mind what kind of test will be necessary to conduct in order to validate those requirements
- Sketch a block diagram for the solution for Lab Work 1 – what are the essential blocks of the system an their relationships?
  - https://www.smartdraw.com/block-diagram/