Introduction

1. Embedded Systems

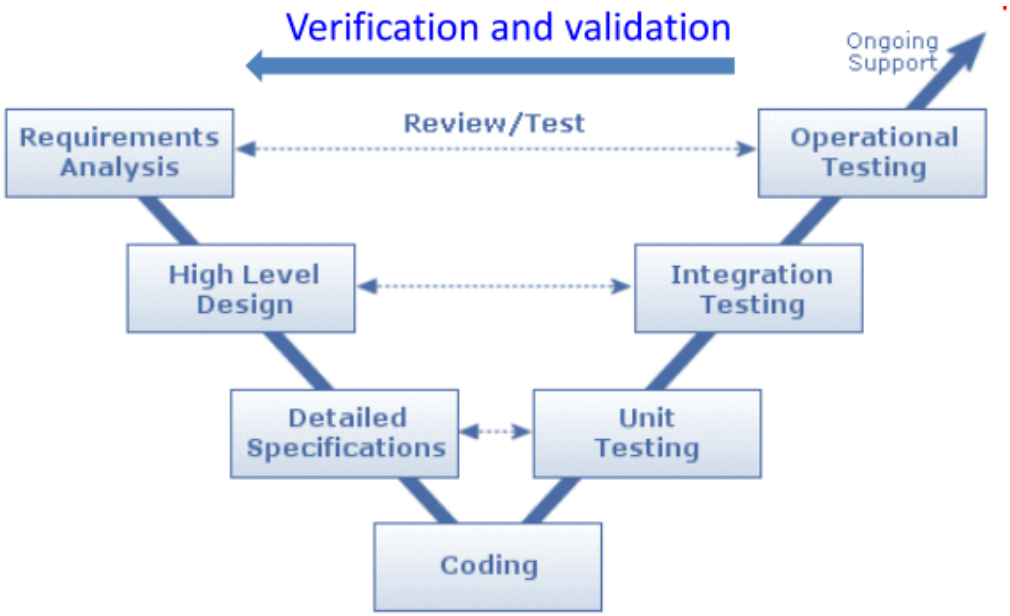
1.1. Embedded Systems

It is an **information processing system** that is **embedded into** an enclosing physical product

Unlike a PC or servers, an embedded system "**interacts**" with its physical world

1.2. Cyber-Physical Systems

A CPS is typically designed as a network of interacting **elements** with physical input and output instead of as standalone devices



1.3. System

A system is **an entity that interacts with other entities**, i.e., other systems, including hardware, software, humans, etc

1.4. Function

The function of a system is **what the system is intended to do** and is described by the functional specification in terms of functionality and performance

1.5. Behaviour

The behavior of a system is **what the system does to implement its function**. The behavior, for example, can be described by a sequence of states

1.6. Structure

The structure of a system is **what enables it to generate the behavior**.

1.7. Specifications

They describe the **functional and non-functional** requirements of the system

1.7.1. Non-functional Requirement (NFR)

A **non-functional requirement (NFR)** is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

2. Quantitative Evaluation

2.1. Quantitative Measures

Example of Quantitative measures:

1. Extrema (worst/best case)
2. Reachability (expected time until something happens)
3. long-run average

2.2. Quantitative Properties

There are lots of Quantitative Properties, there are some examples:

2.2.1. Dependability

Definition 1: The ability to deliver service that can **justifiably be trusted**

Definition 2: The ability to **avoid** service **failures** that are more frequent and more severe than is **acceptable**

2.2.2. Service/function Failure

1. Fault(Bug)

- A static defect in software (incorrect lines of code) or hardware
- A fault is the adjudged or hypothesized cause of an **error**
- "physical" exists

2. Error

- An incorrect internal state (unobserved)
- An error is the part of total state of the system that may lead to its subsequent service failure
- a fault that is on the program path

3. Failure

- External, incorrect behavior with respect to the expected behavior (observed)
- an error that caused an observable deviation of correct behavior

Attention: A design without specifications cannot be right or wrong, it can only be surprising. So first we need to know the **desired behavior**

2.2.3. Ways to deal with faults

1. Fault preventation: Better design, better tools
2. Fault detection: testing, debugging
3. Fault removal: fixing, patching
4. Fault tolerance: redundancy, isolation
5. Fault Forecasting: estimating

3. Modeling, design, analysis

3.1. Modeling

Modeling is the process of gaining a deeper understanding of a system through imitation

Models specify **what** a system does

3.2. Design

Design is the structured creation of artifacts.

It specifies **how** a system does what it does. This includes optimization.

3.3. Analysis

Analysis is the process of gaining a deeper understanding of a system through dissection.

It specifies **why** a system does what it does (or fails to do what a model says it should do).