Industrial Embedded Systems
- Design for Harsh Environment -

Dr. Alexander Walsch
alexander.walsch@ge.com

IN2244

Part I - Introduction

WS 2014/15

Technische Universität München

Learning Objective

- Requirements analysis (strategy, tools)
- Reliability in embedded systems design (HW + SW)
- Functional safety
- Design patterns (HW+SW)
- Verification +Validation

We will focus on small footprint systems. A detailed understanding of hardware is essential. An organized approach to software development is key to address quality.

Why you Should attend

- You get credit
- Computer system design is about understanding and answering high-level requirements. Those requirements are usually very similar at a high level:
  - Functionality
  - Reduced life-cycle cost
  - Increased availability
  - Safety
  - User experience

Lecture Organization

13.10.2014	Lecture
27.10.2014	Lecture
10.11.2014	Lecture
24.11.2014	Lecture
08.12.2014	Lecture
22.12.2014	Lecture
19.01.2015	Lecture/Q&A
tbd	exam (oral or written)

Time?

Handout?

http://www6.in.tum.de/Main/TeachingWs2014IndEmbSystems

Motivation

- What is critical infrastructure?

Motivation

- What is critical infrastructure?
  (from wikipedia) a term used by governments to describe assets that are essential for the functioning of a society and economy. Most commonly associated with the term are facilities for:
  - electricity generation, transmission and distribution
  - oil and gas production, transport and distribution
  - telecommunication
  - water and food supply
  - heating (e.g. natural gas, fuel oil, district heating)
  - public health (hospitals, ambulances)
  - transportation systems (fuel supply, railway network, airports, harbors)
  - financial services (banking, clearing)
  - security services (police, military).

Source: GE O&G

Motivation II

- How does this relate to this lecture?
  Critical infrastructure relies on computer systems, sometimes deeply embedded (depending on the hierarchical control level and the control strategy) – a hidden technology:
- No-one cares if they do their job. A disaster if they fail.
- It is not only important what we do but also how and how well we do it
  - process (especially in a regulated environment)
  - integrity (there are a few but one counts more than others): *reliability*
- Why will it be even more important in the future?
  - More and more electric systems (electrification)
  - Autonomous systems (no human in the loop or teleoperation)

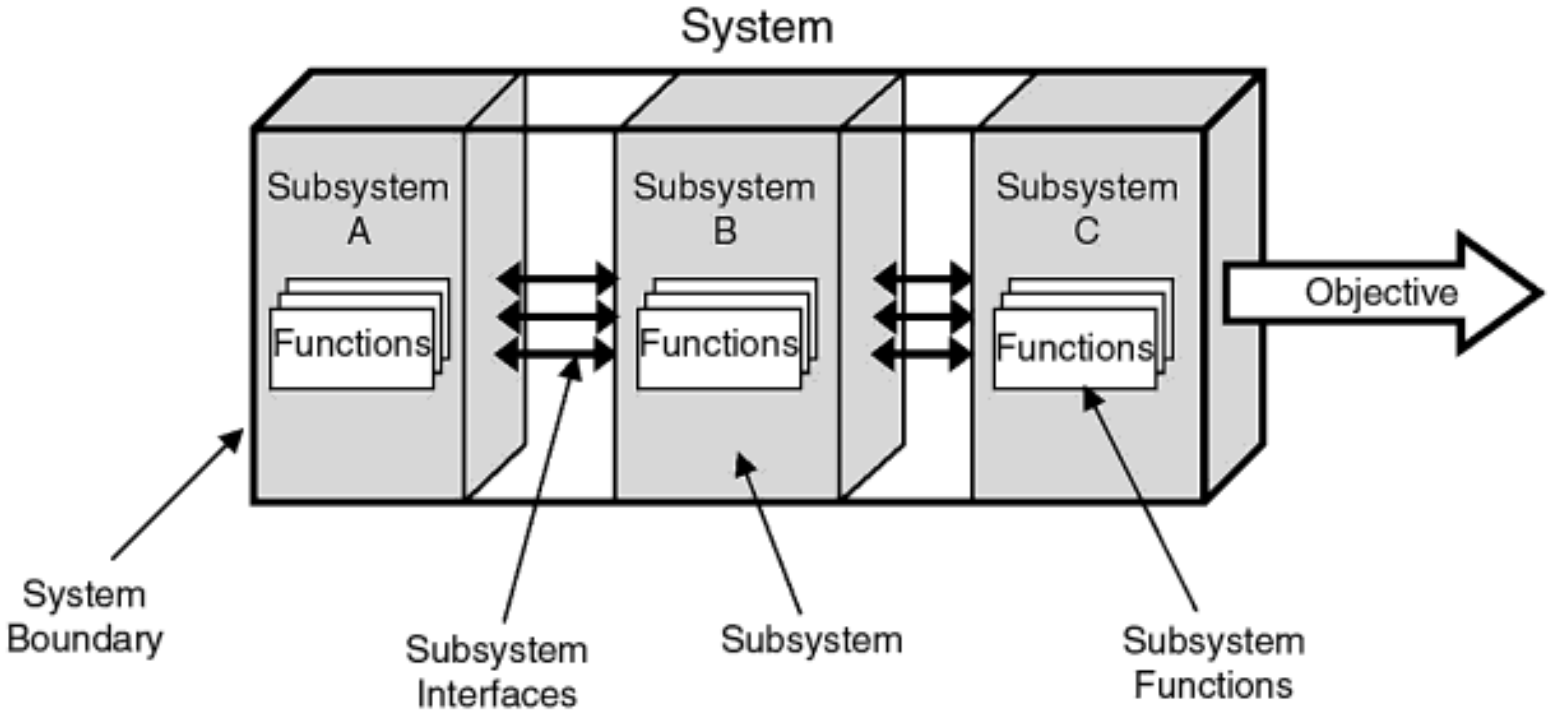
Some Books I Recommend

D. Patterson, J. Hennessy	Computer Organization and Design: The Hardware/Software Interface	Computer architecture textbook
A. Tanenbaum	Operating Systems Design and Implementation	Operating systems textbook
J. Labrosse	MicroC/OS-II	Good introduction to RTOS
D. Smith	Reliability, Maintainability and Risk	Reliability textbook
B. Kernighan, D. Ritchie	The C Programming Language	Introduction to C
L. Hatton	Safer C	Introduction to common pitfalls when using C
D. Smith	Safety Critical Systems Handbook	Introduction to safety critical systems, especially taking IEC61508 into account
C. Ericsson III	Hazard Analysis Techniques for System Safety	Various techniques (FTA, ETA, FMEA, Markov, ...)
Dimitri P. Bertsekas, John N. Tsitsiklis	Introduction to Probability	Probability theory
A. Spillner, T. Linz	Basiswissen Softwaretest	General textbook on testing

Systems
- What is a System? -

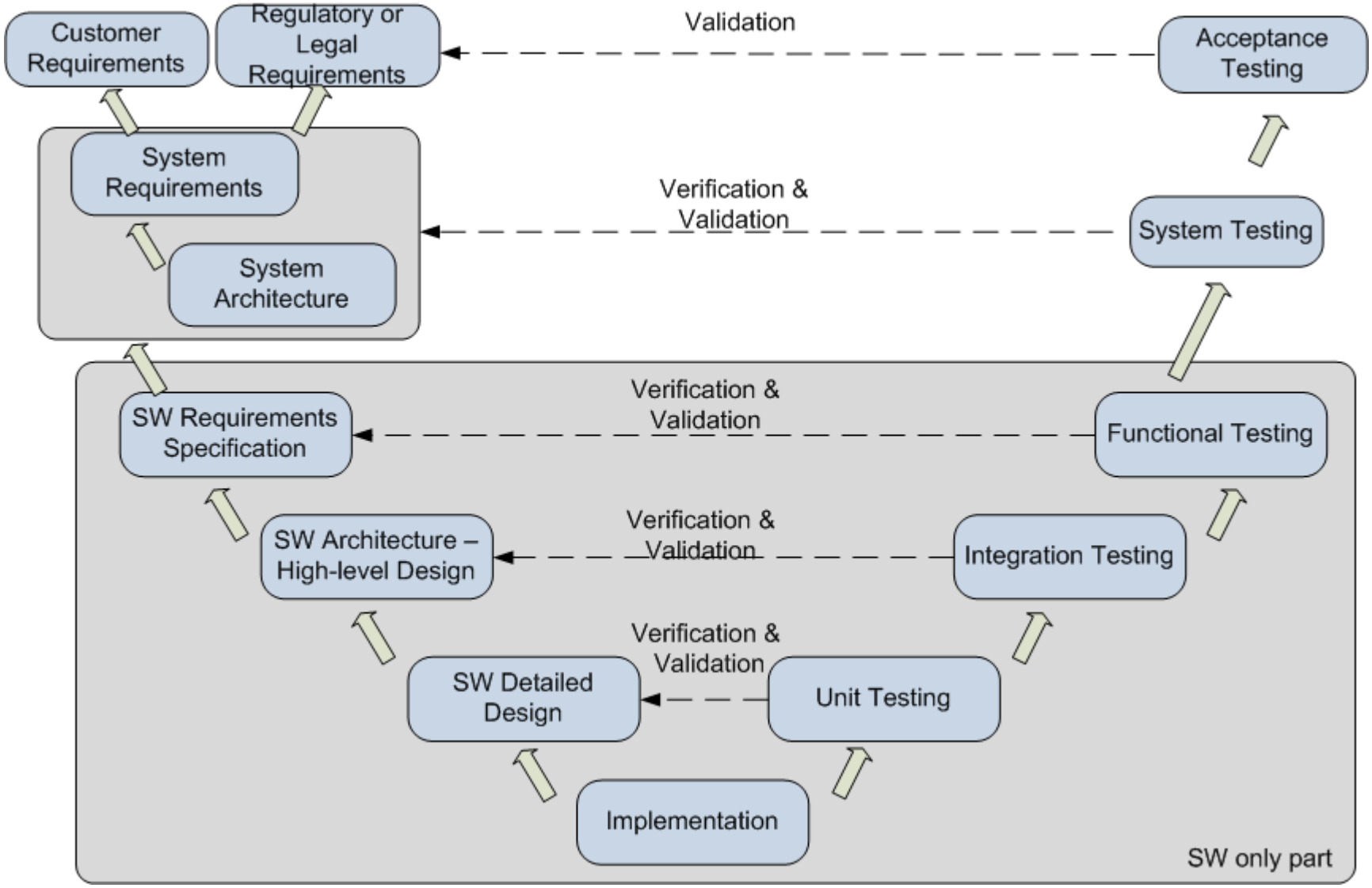
System border, subsystems, interfaces, functions, objectives, environment, external interfaces

High-level functions can be spread across subsystems. Components are basic building blocks (e.g. CPU).



Source: Ericson, Hazard Analysis Techniques for System Safety

The V-Model


Customer Requirements
Regulatory or Legal Requirements
Validation
Acceptance Testing
System Requirements
System Architecture
Verification & Validation
System Testing
SW Requirements Specification
Verification & Validation
Functional Testing
SW Architecture – High-level Design
Verification & Validation
Integration Testing
SW Detailed Design
Verification & Validation
Unit Testing
Implementation
SW only part

The V-Model II

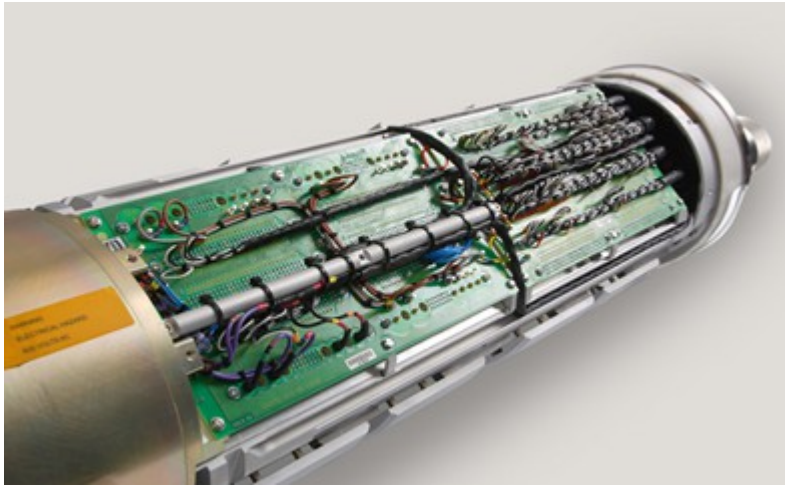
- V-model on previous slide (covers system and software, y (vertical) = refinenement, x (horizontal) = time)
- A requirements driven approach
- Design gets more detailed as it proceeds in time
- Tests are defined at the time of design (testability - left wing of V)
- Tests are executed at the V+V stage (e.g. pass/fail - right wing of V)
- Iterative process that defines phases for design and testing
- Used for system, HW, SW
- Other processes: waterfall, spiral, agile
- Development process in general: defines steps, deliverables, reminds us so we do not forget anything

IEC 61508

- General safety standard as a guideline
- Risk based approach to safety
- Importance of architecture
- Contains a lot of guidelines on hardware and software design
- We will try to not use terminology of the standard but rather concentrate on its concepts
- Link: http://en.wikipedia.org/wiki/IEC_61508

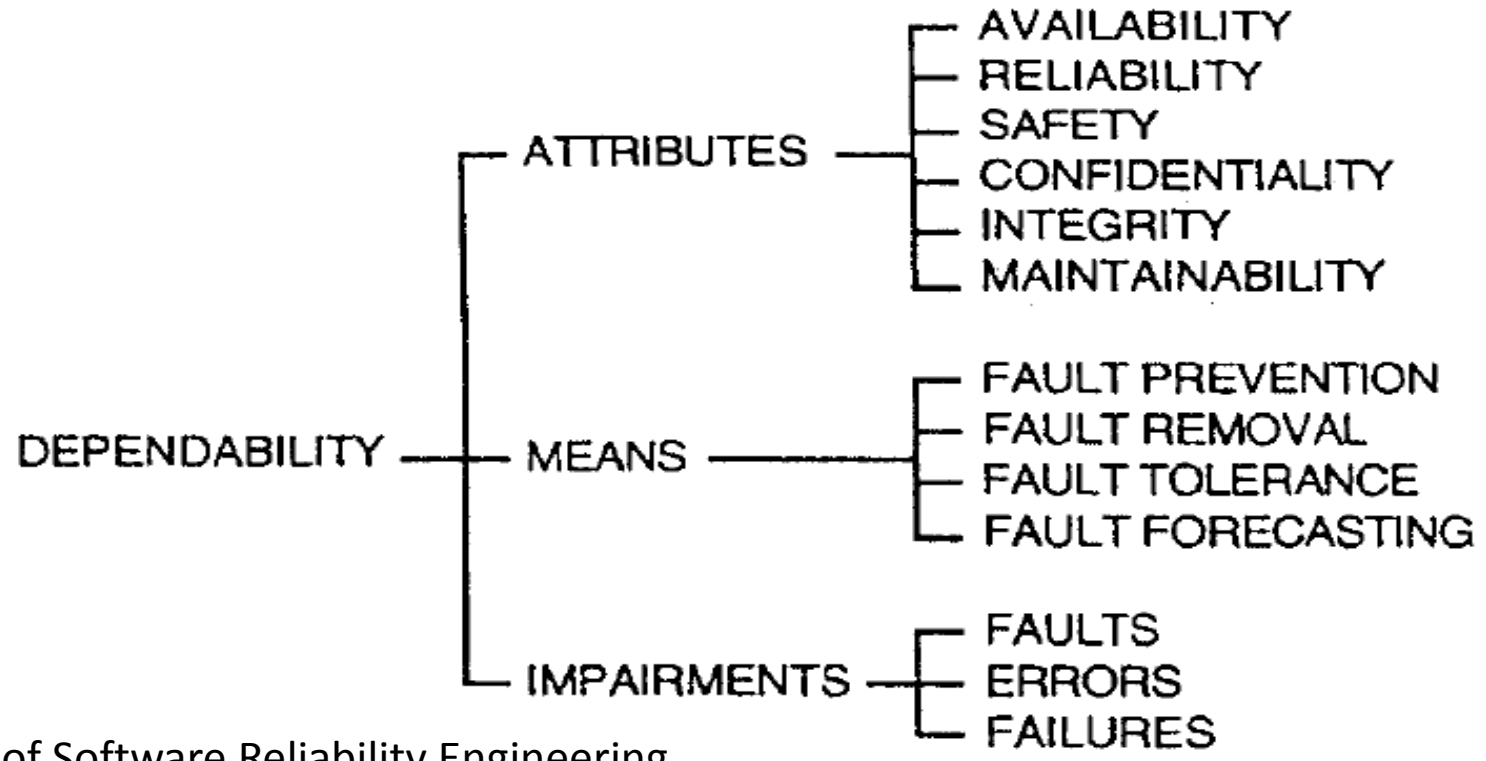
The safety standard will serve as a design cookbook.

Example Systems – from GE



Embedded System Development

Main Drivers:
Cost, Function, Performance, Dependability (trustworthiness)



Source:
Handbook of Software Reliability Engineering
IEEE 1996, Michael R. Lyu