Verification and Validation

- Verification: to detect errors in a verification object (design model, program, HW-circuit) with respect to a requirements specification
  "To build the system right"

- Validation: to determine whether requirements reflect "needs of the user"
  "To build the right system"

V&V methods

In reality done using a combination of following activities:
- Testing
- Inspection
- Simulation
- Formal verification
- Prototyping
- Use cases (scenarios)

V&V in development process

Inputs to verification?

1. Req. spec
2. Verification object

Fault in verification object (or "fault" in req. spec)
Testing of HW circuits

1. Test vectors generated by test synthesis (ATPG eller BIST)
2. VHDL code or manufactured chip

Trend: Design For Testability
How: Autonomous test equipment (ATE) costs $5m and more!

Multi-domain systems

Transition from hydraulics to elektromechanical and heterogenous systems
Object-oriented simulators with modern model libraries, Dymola (Modelica)

System on the chip

Problems with imported subsystems (IP: intellectual property)
Each tested separately but the behaviour/performance of the composition difficult to test
Combination of circuit types, synchronous/asynchronous, digital/analog circuits

Input to software testing?

1.
2.
Tools?

Simulation of hydraulic systems

1. Tables, drawings, test database over earlier test runs
2. Library modules for hydraulic components and standard couplings
Tool: In-house simulators (large Fortran library)

Testing - possible?

No method (alone) works!
40% of medical systems that are withdrawn by FDA are rejected due to program errors
In a typical application ~ 35% of the code is tested
Is it possible to perform full testing of critical systems?
State space search

- With 55 variables, at 1 MHz, it would take (in the worst case) over 1 billion years to visit every state!

Inspections

- Find the fault directly instead of finding the error (the symptom) like testing does
- Needs special training and planning
- Often carried out in groups

Other problems

- Testing of heterogenous systems expensive (hardware in the loop simulations)
- Some systems can not be tested (nuclear reactors, ...)
- The system must be tested during maintenance and adaptation to new requirements (flight control systems)
- Microsoft evolutionary model

A success story

- C130J Hercules safety-critical software
- At time of sale - after all certification :-(
- Combination of inspections, static analysis (formal verification)
- 70 man-years, 11590 anomalies
- 3% av anomalies safety-critical
- Check the article on the course webpage!

Combination of techniques

- Errors left after the requirements specification phase 70 times more expensive to fix when discovered at acceptance tests
- Can one find more errors at the early stages of development?

Formal verification!