Context of testing

- Verification and Validation
- The V-model
- Testing vs. Debugging
- Testing vs. Inspection
- How much is enough?
  - Good enough
  - Safe enough
  - Competition

Test planning

- Determine goals
- Ensure testability
- Design tests
- Evaluation criteria - what is good enough?
- Test tools?
- Time plan
- Documentation

What is a test?

- A **test suite** is a set of test cases run together for a single purpose.
- A **test case** consists of
  - Test data
  - Expected outcome
  - Expected behaviour

Reusability:

- *Regression test* - rerun a test suite for every new version of the system
- *Back-to-back testing* - use a previous version of the system as the oracle

Problems:

- Which input data should be tested?
- How do we know that the output is correct (*oracle problem*)?

The oracle problem

What is the correct answer?
1. ... at least the program didn't crash ...
2. Compute by hand and compare
3. Back-to-back testing
4. The answer is "reasonable"
   - Is the list sorted?
   - Is the yellow ball yellow and round?
   - Is the area of the triangle between ... and ...
**Classification of testing**

- Classification by goal:
  - finding defects
  - acceptance / validation
  - measurement: reliability, performance, ...

- Classification by level
  - system
  - subsystem
  - module

**Acceptance test (system)**

- Factory acceptance test (FAT)
  - Installation
- Site acceptance test (SAT)

Goals:
- is the contract fulfilled? (verification)
- is the product usable? (validation)

**Measurement**

- Statistical test ("random" test)
  - "normal" input to measure how the system "normally" behaves.
  - test data selected from user profile

- Stress test
  - How the system handles increasing / extreme load
  - graceful degrading / total collapse
  - may reveal defects

- Profiling: 10% of the code takes 90% of the time

**Integration / Interface testing**

- Top-down vs. Bottom-up
- Needs scaffolding: stubs for unfinished parts.

- Test for
  - Miscommunication (arguments, ...)
  - Timing (mutex, deadlock)
  - Environmental assumptions (available services, memory, etc.)

**Test tools**

- Automated testing
  - Record, Replay
- Test environment, scaffolding
- Large test suites (stress, statistical test)

- Evaluation
  - Profiling
  - Coverage

- Documentation, traceability

**Defect testing**

Goals:
- detect as many defects as possible
- detect the most damaging defects
- detect the most likely defects - statistical test!

**Black-box testing:** the source code is not considered (maybe even not known).

**Glass-box testing:** the tests are chosen based on the source code.