

# **Software processes, quality, and standards**

## **Static analysis**

**Jaak Tepandi, 19.10.2016**

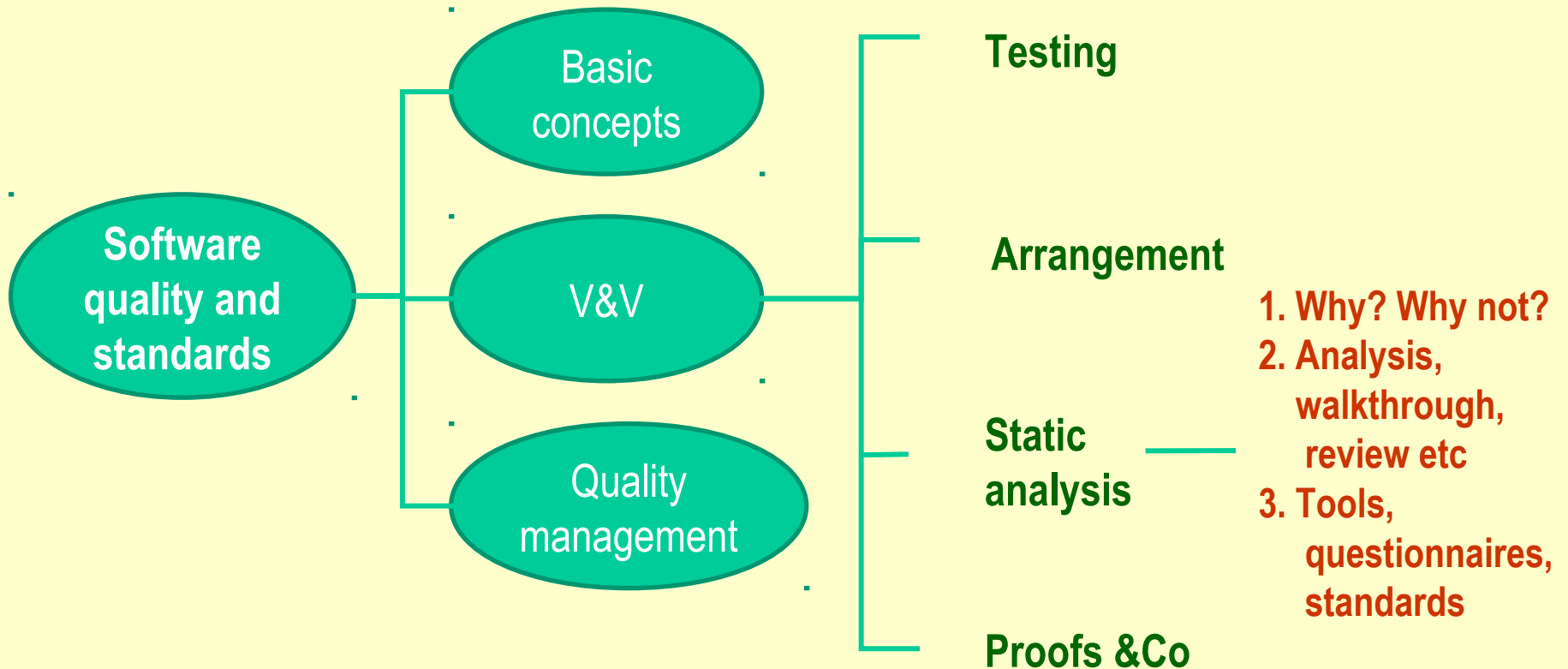
**Tallinn University of Technology**

**Institute of Informatics**

Moodle: „Software Quality (Tarkvara kvaliteet)”

Alternate download: [tepandi.ee](http://tepandi.ee)

# Context



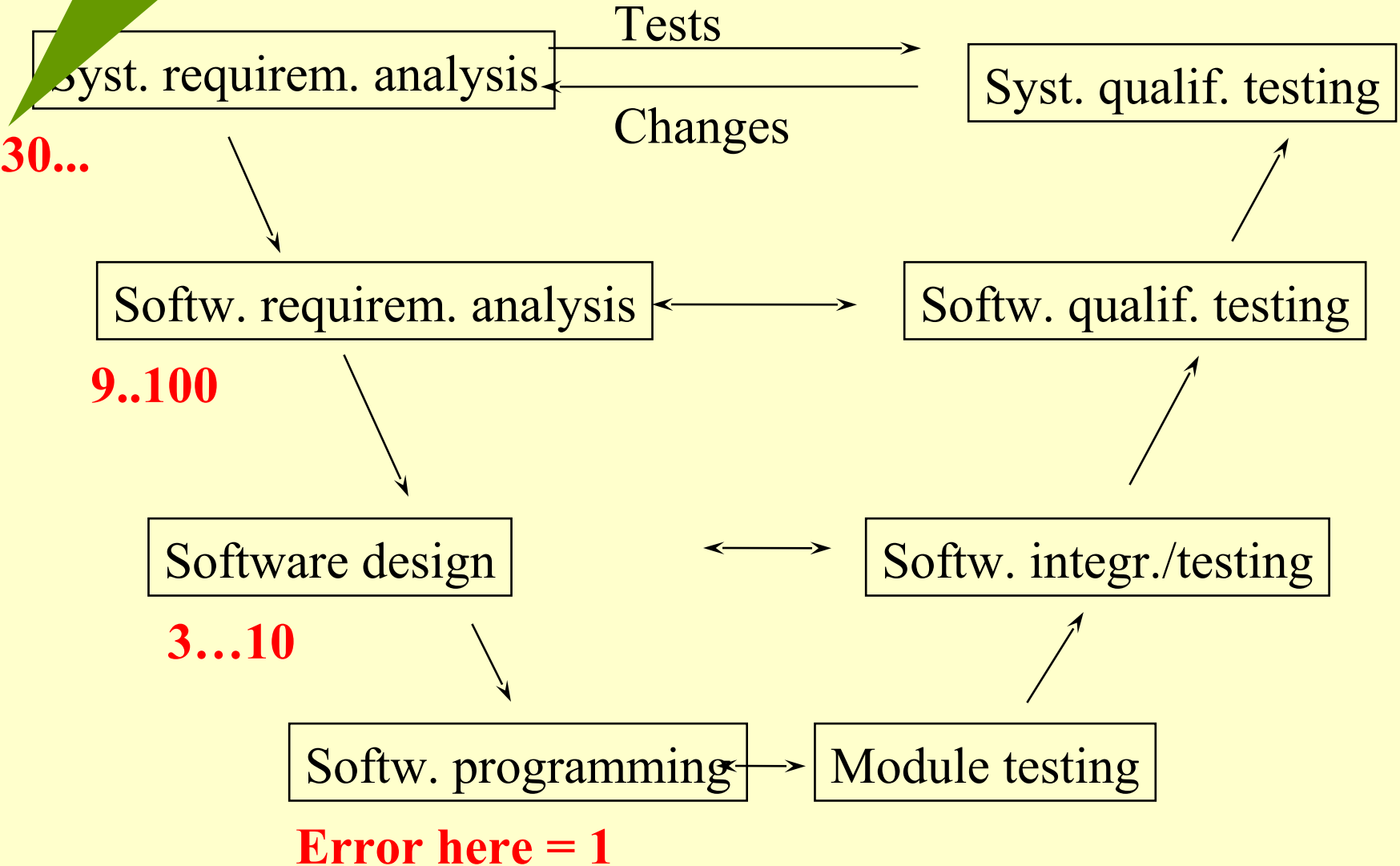
# Example: Therac 25

- At least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation, approximately 100 times the intended dose
- Two of the six patients died as a direct consequence
- The failure only occurred when a particular nonstandard sequence of keystrokes was entered, the operator changed the setup too quickly, and an arithmetic overflow occurred
- Testing?



# *In agile software development as well*

**When? When not?**



# Static analysis: Why? Why not?



Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software artifacts (ISTQB).

Why?

- Errors are discovered at the early steps of development
- It is not possible to test all situations
- Reliability gained with testing has limits
- Several important system qualities (for example maintenance criteria) are difficult to evaluate by testing

Why not?

- Does not replace testing (even proofs)
- Time, coordination, possible side-effects



# Static analysis: arrangement, topics, tools

## Arrangement, examples

- Walkthrough/Inspection/Review
- Analysis by the author
- ....Programmer`s evaluation
- ...Strong methods, program proofs
- ...Some methods from V&V arrangement, quality management

## Topics, examples

- Contract review, design walkthrough, code inspection

## Tools

- Example: coding questionnaire
- Standard as questionnaire
- Automated tools



# Analysis by the author

Recommended:

- finds errors
- cost-effective
- train if not used

Not sufficient: the author

- follows own logic
- is motivated to finish
- may be not motivated to destroy own work



# Walkthrough

- A walkthrough or walk-through is a form of software peer review in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems (ANSI/IEEE 1028)





# Walkthrough: advantages



- errors can be found at early steps of development
- the best way of reducing errors
- team contacts improve
- productivity and quality are improving
- people can be replaced



# Walkthrough problems

- group members can be from different departments
- group members can be different: with high IQ, impatient, conservative, not very interested of “real world”, prefer privacy etc
- nobody likes criticism, in extreme cases co-operation declines
- wrong meeting management

[https://www.youtube.com/watch?v=oLmDe8pAc6I&index=2&list=PLaD4FvsFdarQytrGAmCo2qW\\_rAWcfBKeV](https://www.youtube.com/watch?v=oLmDe8pAc6I&index=2&list=PLaD4FvsFdarQytrGAmCo2qW_rAWcfBKeV)



# Preconditions

- all group members should have expectation of what they are expected to do
- good co-operation
- materials are available
- participants have prepared for the session, for example everyone have one positive and one negative comment about materials



# Participants and their roles

- presenter (not necessarily the author)
- coordinator (manager)
- secretary
- members: team members, standards expert, user's representative
- these roles can be combined
- management participation is not recommended



# Walkthrough arrangement

- As many preparations (texts, documents) as needed, as few as possible
- Length 30..60 min



# Recommendations for the walkthrough

- Analyse the product, not the author
- Prepare a plan and follow it
- Do not try to solve all the problems
- Take written notes
- Limit the number of participants, make preparations
- Prepare the questionnaire for each reviewed product
- Reserve resources, including time
- Train the participants
- Learn from past reviews



# Results of a walkthrough

- the problems are found and corrected
- a better system
- a signed protocol
- NB not changes in the status or wages of the participants



# ANSI/IEEE Std 1028. IEEE Standard for Software Reviews and Audits

Product

Product ja Project

Project

- |  |  |  |
|--|--|--|
| <ul style="list-style-type: none"><li>•Simulation</li><li>•Formal proofs</li></ul> | <ul style="list-style-type: none"><li>•Technical review</li><li>•Software inspection</li><li>•Walkthrough</li><li>•Audit</li></ul> | <ul style="list-style-type: none"><li>•Management review</li></ul> |
|--|--|--|





# Example: Scrum meetings

- Sprint planning. At the beginning of the sprint cycle (every 7–30 days)
- Daily Scrum. A daily sprint meeting, 15 minutes
- Sprint review (concerns work done). 2 h
- Sprint retrospective (concerns the process). 1, 5 h
- Backlog refinement.
- Scrum of Scrums. Coordinating multiple teams



# *Example: XP (planning)*

- User stories are written.
- Release planning creates the schedule.
- Make frequent small releases.
- The Project Velocity is measured.
- The project is divided into iterations.
- Iteration planning starts each iteration.
- Move people around.
- A stand-up meeting starts each day.
- Fix XP when it breaks



# Static analysis tools

- Questionnaires
- Standards
- General code analysis tools for many programming languages  
eg, .NET, Java, JavaScript, Python...
- Analysis for security vulnerabilities
- Analysis for accessibility
- Analysis for performance degradation
- Design verification tools
- Proofing tools
- Support for analysis arrangement
- ....



# How to find by testing?



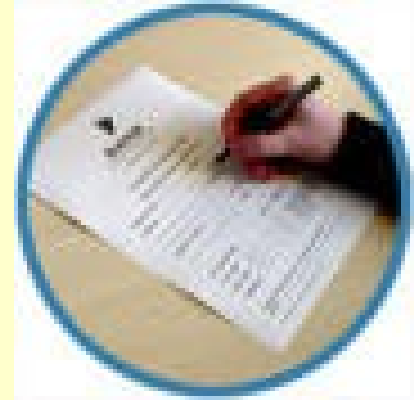
```
if ...|| strcmp(request->user_agent,  
"xmlset_roodkcableoj28840ybtide") == 0)
```

```
{ return AUTH_OK; }
```

```
lw    $a0, 0xB8($s0)  
la    $t9, strstr  
nop  
jalr  $t9 ; strstr  
nop  
lw    $gp, 0x3B8+saved_gp($sp)  
nop  
la    $a1, 0x470000  
nop  
addiu $a1, (laXmlset_roodk_0 - 0x470000) # "xmlset_roodkcableoj28840ybtide"  
bnez  $v0, end  
li    $v1, 1
```

```
lw    $a0, 0xD0($s0)  
la    $t9, strcmp  
nop  
jalr  $t9 ; strcmp  
nop  
lw    $gp, 0x3B8+saved_gp($sp)  
beqz  $v0, end  
li    $v1, 1
```





# Questionnaires

- Different volume (10...1000)
- Different aspects
  - Compliance to coding best practice
  - Application area related
  - Process related
  - ...
- + tools



# Example of a simple questionnaire: coding

## Topics

- Declarations / usage of variables
- Assignments
- Calculations
- Comparisons
- Iterations / Halting
- Calls, input/output
- Security

Compare eg Robert C. Martin. Clean Code



# Declarations / usage of variables

- Variable names meaningful?
- Variables declared?
- Default attributes correct?
- Initialization correct?
- Identifiers? VOLT, VOLTS, I1, O0?

Etc



# Assignments

- Data conversions correct?
- Value exists?
- Index outside limits?
- Index integer?
- Common data structures defined in the same way?
- Etc





# Computations

- Incorrect data types?
- Mixed data types?
- Under- / overflow?
- /0?
- Values outside expected limits?
- Multiple small errors?
- Integer arithmetics? Cf  $17/3*2$ ,  $17*2/3$
- Operation priorities?
- Etc



# Comparisons

- Mixed types in comparisons?
- Specification => code correct?
- Priorities correct?
- Result depends on compiler?
- Real variable compared with a value?
- Etc

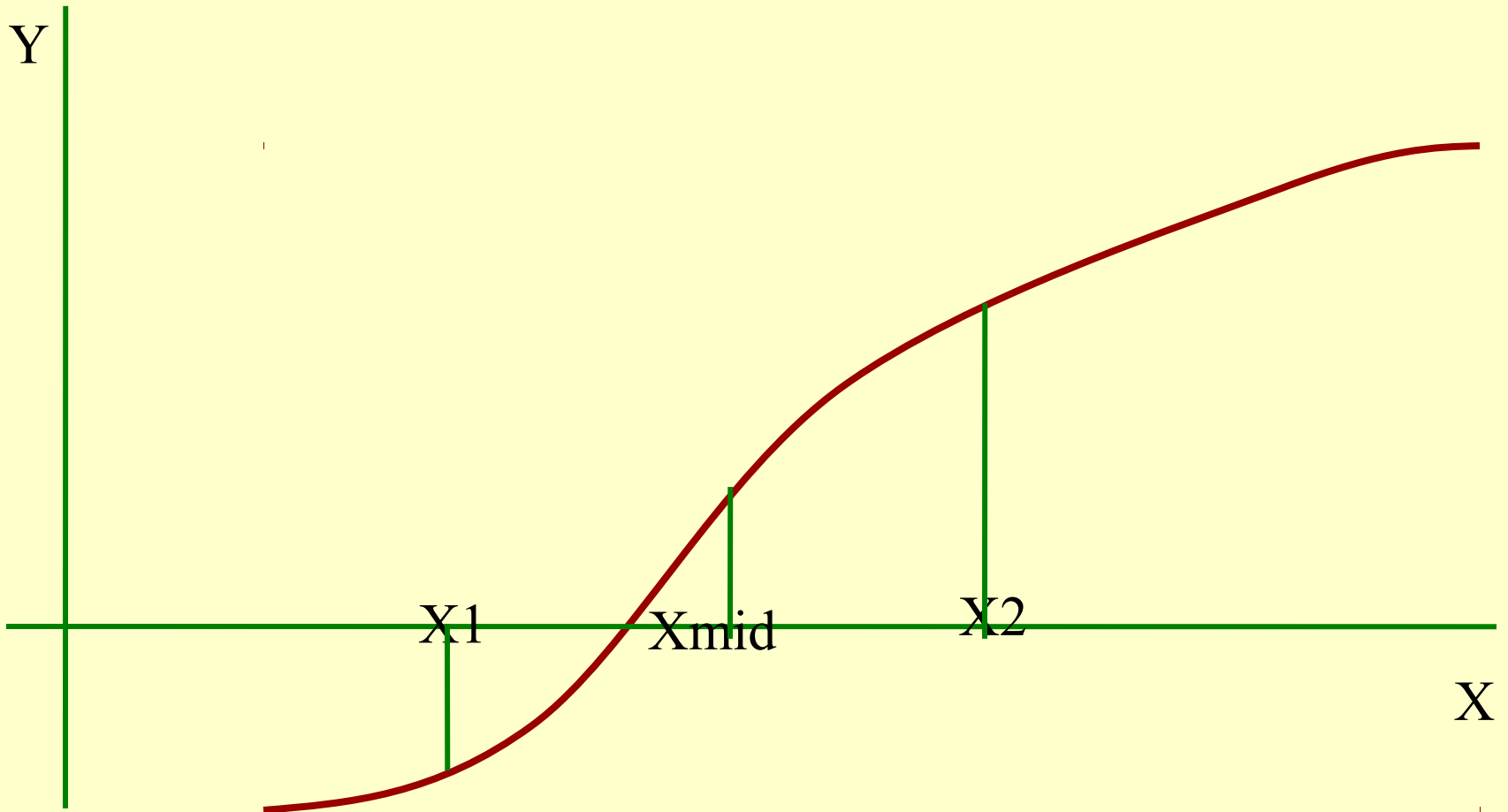


# Iterations / Halting

- Will it stop?
- Condition is F at the start?
- FOR counter start > stop?
- Large number of iterations?
- Etc



# Finding a solution



# Functions, calls, input/output

- Long list of arguments?
- Multi-purpose functions?
- Are units of measure in the calling program the same as in the function?
- Is parameter ordering the same?
- Are windows closed properly?
- Is error handling correct?
- Etc

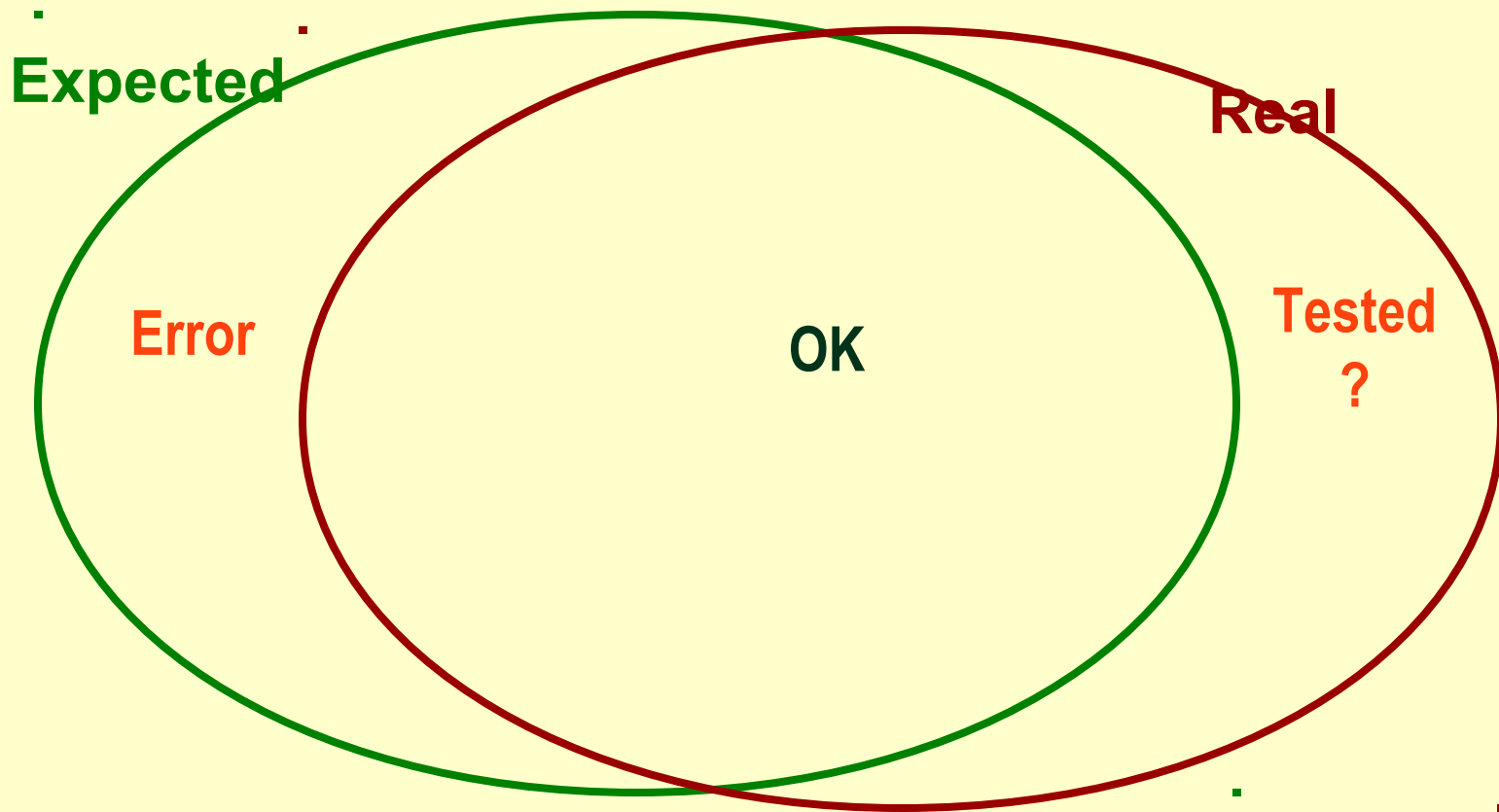


# NASA: Human error caused loss of Mars orbiter

- Failure to convert English measures to metric values caused the loss of the Mars Climate Orbiter, a spacecraft that smashed into the planet instead of reaching a safe orbit, a NASA investigation concluded Wednesday.
- An investigation board concluded that NASA engineers failed to convert English measures of rocket thrusts to newton, a metric system measuring rocket force. One English pound of force equals 4.45 newtons. The difference between the two values caused the spacecraft to approach Mars at too low an altitude and the craft is thought to have smashed into the planet's atmosphere and was destroyed.
- Ground software - orbiter software



# Unspecified behaviour?



# Problems?

```
void SomeFunction( char *pszInput )  
{  
    char szBuffer[10];  
    strcpy(szBuffer, pszInput); . . .  
}
```





# Problems?

```
Aut-data = "SELECT Username FROM  
Aut-table WHERE Username = ' " &  
Request.Form("Name") & " 'Password = "  
Request.Form("Password") & " ' "
```



# Standard as questionnaire (example based on ISO/IEC 12207)

- The acquirer shall conduct acceptance review and acceptance testing of the deliverable software product or service and shall accept it from the supplier when all acceptance conditions are satisfied. The acquirer shall define and document the acceptance strategy and conditions (criteria).

=>

- Has the acquirer conducted acceptance review and acceptance testing of the deliverable software product or service?
- Has the acquirer accepted the product or service when all acceptance conditions are satisfied?
- Has the acquirer defined and documented the acceptance strategy and conditions (criteria)?



# Takeaway: static analysis

- What? Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software artifacts
- Why? Errors are discovered at the early steps of development; it is not possible to test all situations; several important system qualities are difficult to evaluate by testing
- But: does not replace testing (even proofs); requires time, coordination; possible negative side-effects
- Method examples: Walkthrough/Inspection/Review; analysis by the author; programmer`s evaluation; formal methods; methods from V&V arrangement and quality management
- Examples: contract review, design walkthrough, code inspection
- Tool examples: coding questionnaire; standard as questionnaire; automated analysis tools; bug tracking



# Additional reading (examples)

Daniel Galin, Software Quality assurance from theory to implementation, Pearson - Addison-Wesley. Chapters 5,8.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 10 Section 2.3.

Robert C. Martin, Clean Code, Prentice-Hall. Chapter 17.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 3.

