

# Software processes, quality, and standards

## White-box testing

Jaak Tepandi, Jekaterina Tšukrejeva, Stanislav Vassiljev, Pille Haug

Tallinn University of Technology

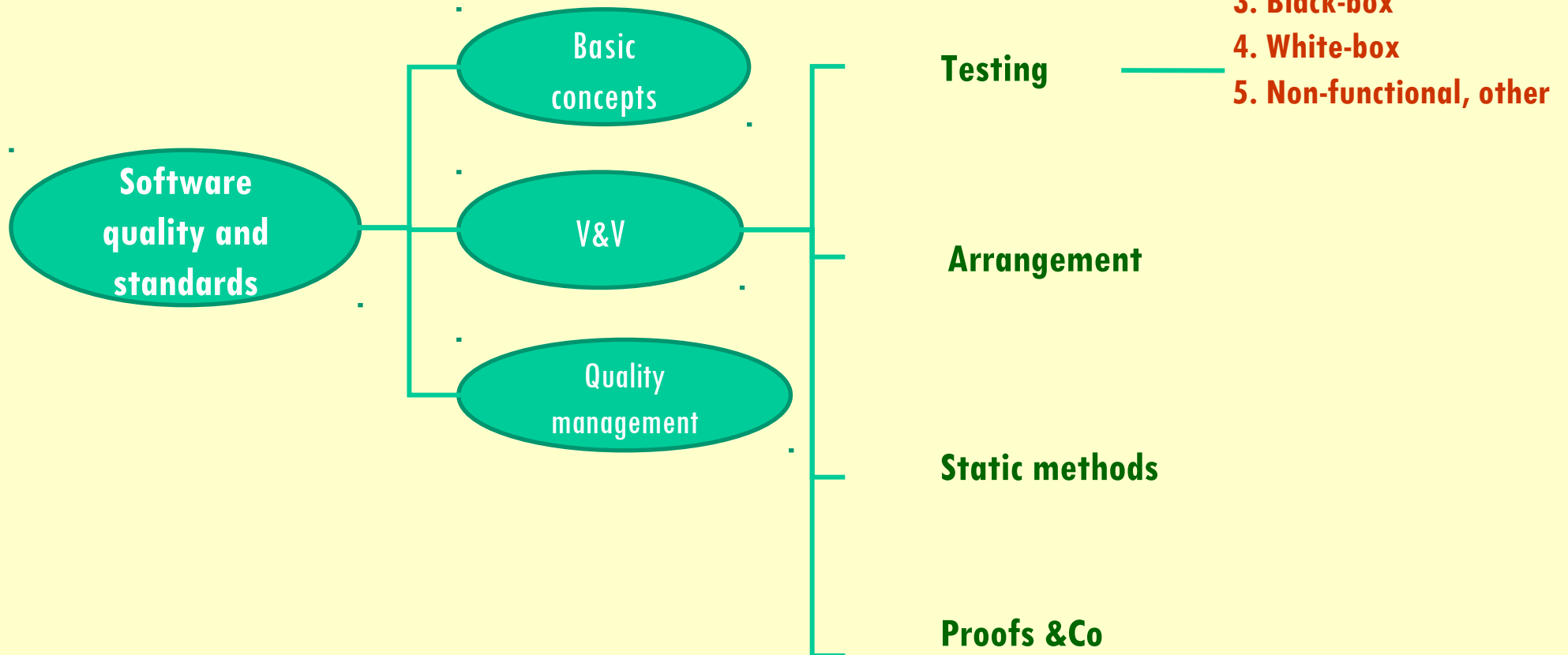
Department of Software Science

Moodle: „Software Quality (Tarkvara kvaliteet)”

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

Version 11.10.2017

# Context and content



# White Box Testing

- The concept
- Statement coverage
- Branch coverage
- Path coverage
- Data coverage
- Loops
- Heuristics



# White-box testing

Test

Input

Program

Output

Based upon the program

Based upon the task /spec

# White-box testing and analysis

Testing based on an analysis of the internal structure of the component or system.

Coverage tests: all / % of statements/paths/branches/... are executed. What for? What do they give? What do they miss?

- statement, branch, path coverage
- criteria comparison
- Data flow testing, based on the definition and usage of variables

## Static methods

- Reviews etc
- Formal methods, used eg in cleanroom software development and Common Criteria

# Statement coverage criterion

- As a result of testing every statement in a program must have been working at least once / X % of statements must have been...
- Is it possible to fulfil?
- Is it enough?
- How many tests are needed for testing program statement coverage?

# Statement coverage?

1 Function max (a,b)

2 Read a,b

3 max := a

4 If b>max

5 Then max := b

6 End If

7 End Function

# Statement coverage?

```
<?php
$h = date("H");
if ($h < "10") {
    echo "Have a good morning!";
} elseif ($h < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```



# How many tests?

```
Public Function GetStatusCode(status As Long) As String
```

```
Dim msg As String
```

```
Select Case status
```

```
Case IP_SUCCESS: msg = "ip success"
```

```
Case IP_BUF_TOO_SMALL: msg = "ip buf too_small"
```

```
Case IP_DEST_NET_UNREACHABLE: msg = "ip dest net unreachable"
```

```
Case IP_DEST_HOST_UNREACHABLE: msg = "ip dest host unreachable"
```

```
Case IP_DEST_PROT_UNREACHABLE: msg = "ip dest protocol unreachable"
```

```
Case IP_DEST_PORT_UNREACHABLE: msg = "ip dest port unreachable"
```

```
End Select
```

# Example: statement coverage?

```
$number_three = 3;
```

```
if ( $number_three == 3 ) {  
  echo "The if statement evaluated to true";  
} else {  
  echo "The if statement evaluated to false";  
}
```

# Possible to satisfy?

...

goto next

x := 56

...

...

if (age >= 20 and age <= 50)

    then begin No := No + 1;

        if (age >= 51) then print "OK"

...

...

print "Input four numbers"; read x,y,z,a;

B := 0;

if (int(x) = x and int(y) = y and int(z) = z and int(a) = a and

    x > 0 and y > 0 and z > 0 and a > 2 and x\*\*a + y\*\*a = z\*\*a)

then B := 1

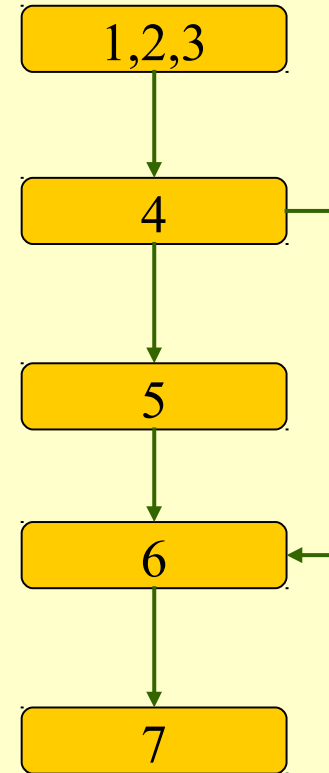
...

# Summary: statement coverage criterion

- Concept: all parts of program have been working (or % working)
- Preconditions: the program text exists and can be analysed; time, skills, tools, other resources exist
- Advantages: program has been systematically tested. Few tests. Clear concept
- Disadvantages: weak coverage. Does not give data tests. Does not test empty branches. The program text is not always available. Not always achievable
- Results: a test set, which covers the program
- Relationship to others: use with other methods
- Evaluation: use when white-box testing is needed, but time is scarce
- Tools: there are tools which help to design tests for and evaluate statement coverage

# Branch coverage

- 1 Function max (a,b)
- 2 Read a,b
- 3 max := a
- 4 If b>max
- 5 Then max := b
- 6 End If
- 7 End Function



# Branch coverage?

Public Function GetStatusCode(status As Long) As String

Dim msg As String

Select Case status

Case IP\_SUCCESS: msg = "ip success"

Case IP\_BUF\_TOO\_SMALL: msg = "ip buf too\_small"

Case IP\_DEST\_NET\_UNREACHABLE: msg = "ip dest net unreachable"

Case IP\_DEST\_HOST\_UNREACHABLE: msg = "ip dest host unreachable"

Case IP\_DEST\_PROT\_UNREACHABLE: msg = "ip dest protocol unreachable"

Case IP\_DEST\_PORT\_UNREACHABLE: msg = "ip dest port unreachable"

End Select

# Program graph? Tests?

Dim userInput As String

userInput = "not ok"

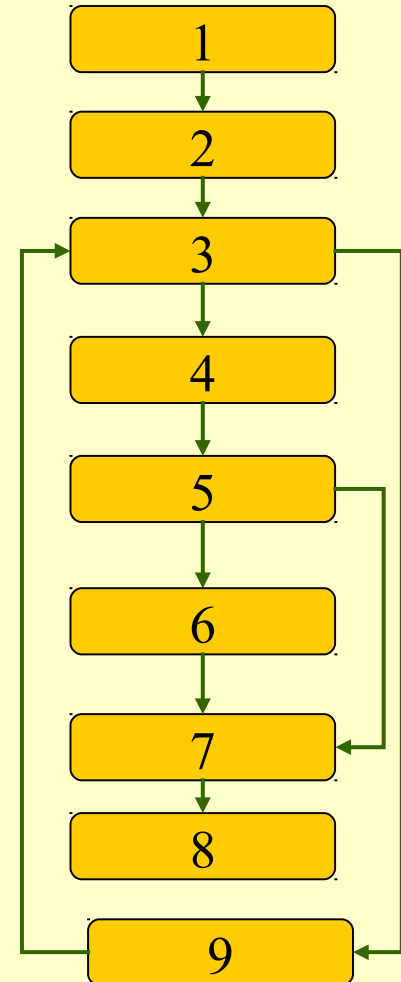
Do Until userInput = "ok"

    userInput = InputBox("Please Enter Secret Word")

Loop

# Branch and multiple condition coverage

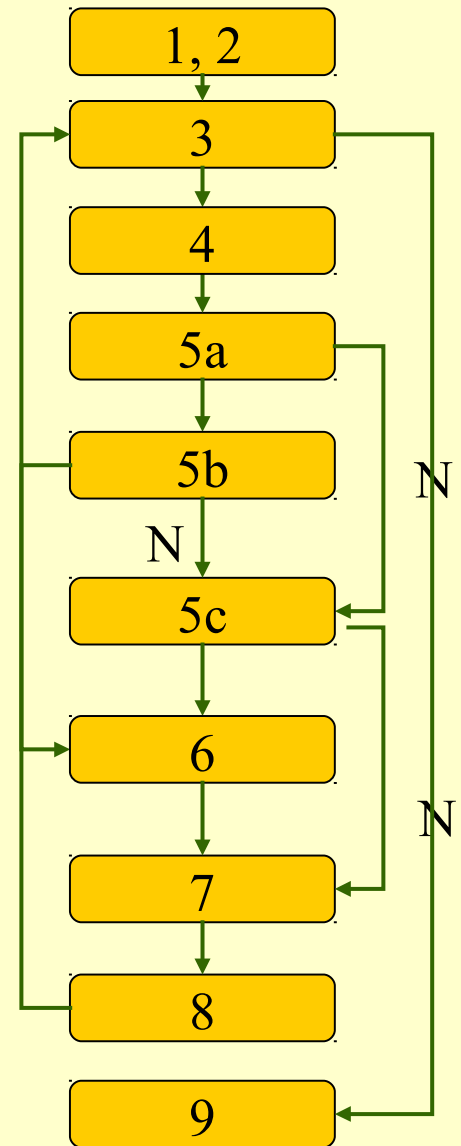
```
1 Function Select (P)
2   I := 1;
3   While not end-of-file (P)
4     Begin
5       If (P(I, age)>20 And P(I, age) < 25)
6         Or (P(I, education)= "high")
7         Then Print P(I, name);
8     End While;
9 End Function
```





# Multiple condition coverage

```
1 Function Select (P)
2   I := 1;
3   While not end-of-file (P)
4     Begin
5abc   If (P(I, age)>20 And P(I, age) < 25)
        Or (P(I, education)= "high")
6       Then Print P(I, name);
7     I:= I+1;
8   End While;
9 End Function
```

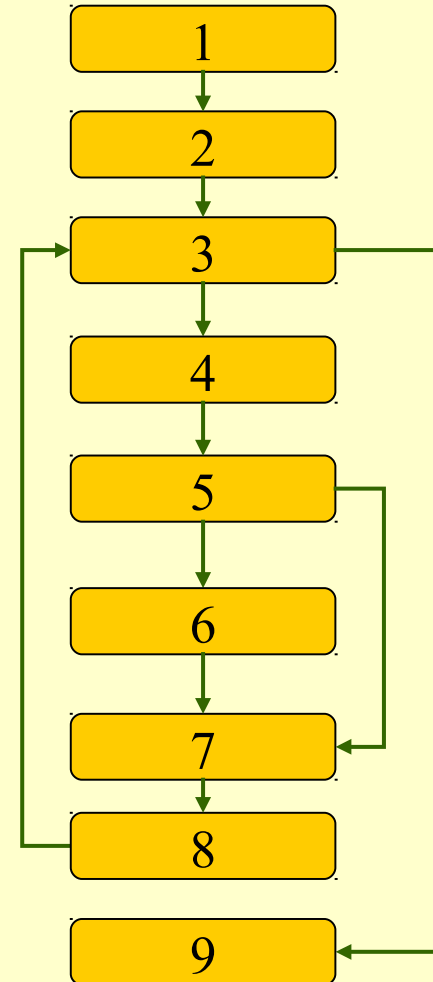


# Summary: branch coverage criterion

- Concept: all program branches have been tested (or % tested)
- Preconditions: the program text exists and can be analysed; time, skills, tools, other resources exist
- Advantages: program has been systematically tested. More tests. Clear concept.
- Disadvantages: does not give data tests. The program text is not always available. Not always achievable
- Results: a test set, which covers the program branches
- Relationship to others: use with other methods
- Evaluation: use if possible
- Tools: there are tools which help to design tests for / evaluate branch coverage

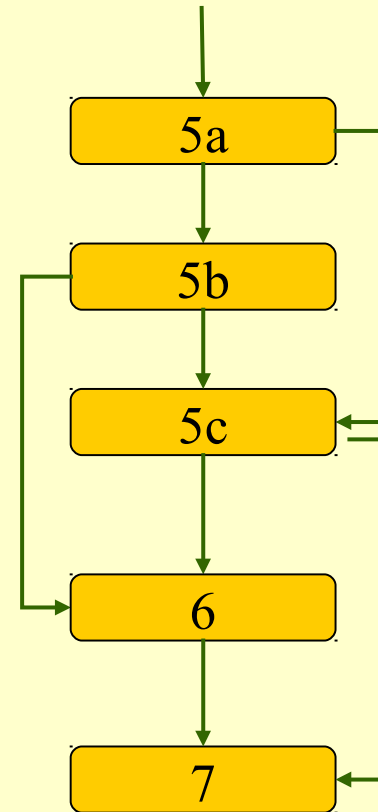
# Path coverage if max 1000 records?

- 1 Function Max (price)
- 2   Max := price(1); I := 2;
- 3   While not end-of-file (price)
- 4    Begin
- 5     If price(I)>Max
- 6     Then Max := price(I);
- 7     I:= I+1;
- 8    End While
- 9 End Function

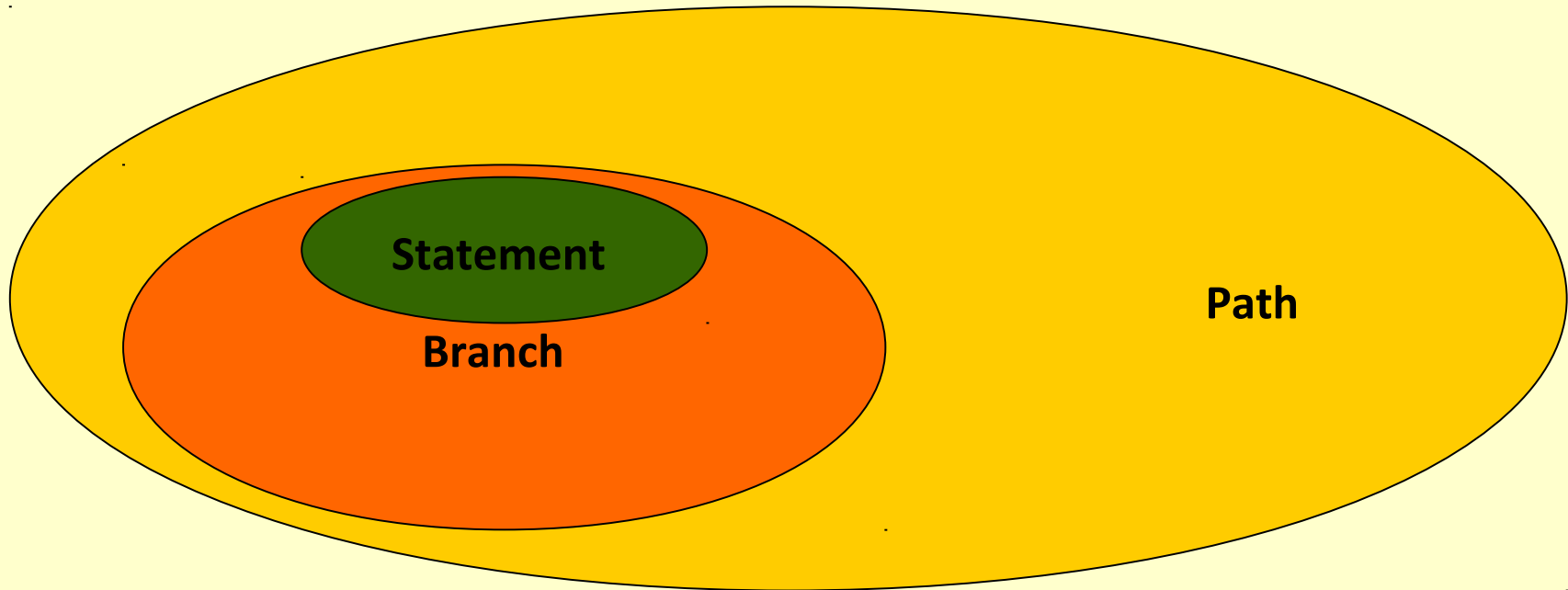


# Paths?

```
1 Function Vali (P)
2   I := 1;
3   While not end-of-file (P)
4     Begin
5abc  If (P(I, V)>20 And P(I, V) < 25)
        Or (P(I,H)= "kesk")
6       Then Print P(I, N);
7     I:= I+1;
8   End While;
9 End Function
```



# Comparison of coverage criteria





# Coverage tool examples

EclEmma, a free Java code coverage tool for Eclipse <http://www.eclemma.org/>

Venus.js <http://java.dzone.com/articles/javascript-unit-tests-and-code>

Istanbul, test coverage for JavaScript software

<http://www.gregjopa.com/2014/02/testing-and-code-coverage-with-node-js-apps/>

EMMA, test coverage for Java software.

<http://users.csc.calpoly.edu/~jdalbey/309/Lectures/emmademo.html>

**What do we have additional here?**

Parasoft <http://www.parasoft.com/coverage-analysis>

Squish Coco <http://www.froglogic.com/squish/coco/>

LDRA,

<http://www.ldra.com/en/software-quality-test-tools/group/by-software-life-cycle/code-coverage-analysis>

# Data coverage criterion

- Apply black box testing ideas in white box testing
- Determine equivalence classes and boundary values on the basis of data structures, source code, etc
- Use as in black box testing
- Outputs based on specification
- Examples: testing data structures, loops, etc

# Usage of equivalence classes / boundary values

- Distinguish the input and output data equivalence classes, if necessary also input/output combinations
- Select data in each equivalence class and in boundary values (optimisation 1)
- Combine those data into tests (optimisation 2)
  - As many correct data equivalence classes as possible into one test
  - Errors one by one (sometimes also the boundary values)
- Evaluate corresponding outputs
- Identify tests, compose a testing plan
- Test, estimate



# Testing loops

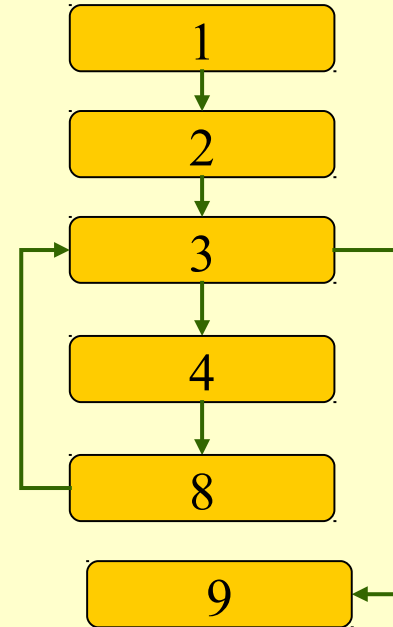
A. Control flow based: statement, branch,... coverage?

B. Data coverage based:

- Lower and upper bounds are boundaries, between them is an equivalence class
- Determine equivalence classes and boundary values
- Use as in black box testing
- Outputs based on specification

# Simple loop

- max n passes
- test 0, 1, 2,  $m < n$ ,  $n-1$ ,  $n$ ,  $n+1$  etc

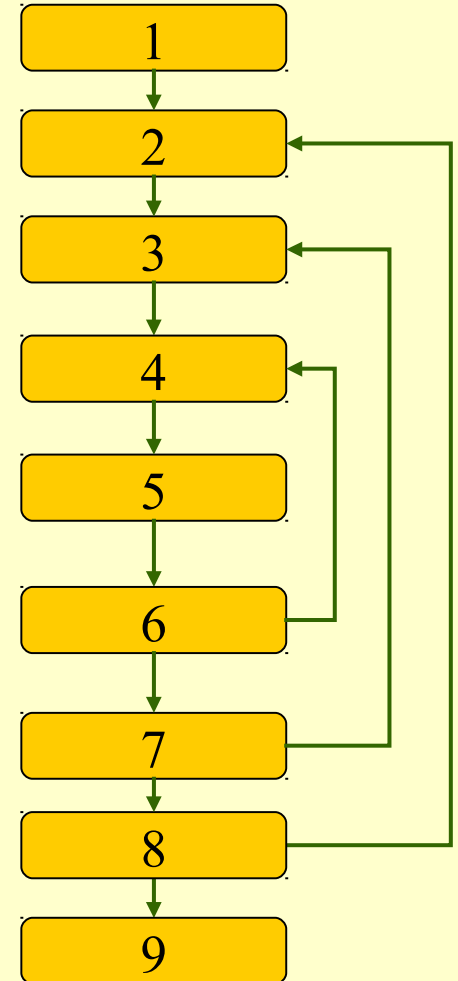


# Nested loops

Instead of all combinations of simple loop test values:

- All test values on each level together with middle values on the other levels
- (Test all equivalence classes and boundary values, but not all their combinations)

```
.....  
While....  
Begin.....  
  
.....  
While...  
Begin  
  
.....  
End While  
  
.....  
End While  
  
.....
```



# Given a program, what can be done?

- “How many tests are needed to test this program?”
- “How many tests are needed to test this program by branch coverage?”
- “What are the branch coverage tests?”

# Smells and Heuristics: Tests

## Insufficient Tests

Use a Coverage Tool!

Don't Skip Trivial Tests

An Ignored Test Is a Question about an Ambiguity

Test Boundary Conditions

Exhaustively Test Near Bugs

Patterns of Failure Are Revealing

Test Coverage Patterns Can Be Revealing

Tests Should Be Fast

Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008, Ch 17

# White box vs black box?

„Black box” used to be specification-based, often with equivalence partitioning / boundaries

- compare state transition testing

„White box” used to be code-based (for test inputs), often analyzing the control flow

- compare equivalence partitioning /boundaries in white box testing

Current software development: multiple artefacts on many levels of abstraction (from high-level requirements to source code and deeper)

- All these may be used in testing

Both methods are usable on most life-cycle phases (for all kinds of SDLC)

- For example, control flow based testing of use cases

The WB/BB distinction may be still useful, eg for learning and management

# Key points to know

White-box testing is based on an analysis of the internal structure of the component or system

Code coverage: function, statement, branch, path, ...

Data coverage, loops

White box vs black box

# Additional reading (examples)

Ian Sommerville. Software Engineering. Ninth Edition. Addison-Wesley, Ch 8.

Daniel Galin, Software Quality assurance from theory to implementation, Pearson - Addison-Wesley. Chapter 9.4.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 4.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 2.

Moodle: „Software Quality (Tarkvara kvaliteet)”. Alternate download: [tepandi.ee](http://tepandi.ee)