

Czech Test Conference, Praha, March 9-10, 2011

# Good Enough Testing Foundation or What You Should Have Learned in Testing Kindergarten

Karol Frühauf

*INFOGEM AG*, CH-5400 Baden

Karol.Fruehauf@infogem.ch

- Contents
1. Introduction
  2. Six illusions about testing
  3. A kind of conclusion



"Big test today, Mam,  
got any BrainFlakes?"

Copyright 2010 *INFOGEM AG*

## Karol Frühauf

since ages a sporty software engineer interested in arts

since 1987 Co-founder *INFOGEM AG*, consultant

1975 Brown Boveri & Cie, Power System Control

1975 completed studies in 'Technical Informatics' the  
Department of Electrical Engineering of RWTH Aachen in  
Germany

1968 emigrated

since 1960 reads

since 1956 plays tennis

since 1954 can read

since 1953 plays table tennis

Co-author of the books

"Software Project Management and Quality Assurance" and  
"Software Checking – a Guide for Testing and Inspections"

Copyright 2010 *INFOGEM AG*

## Member, Contributions



Swiss Association for Quality  
Fachgruppe Informatik



European Organisation for Quality  
Software Group



1995 San Francisco  
2000 Yokohama  
2005 München  
2008 Bethesda  
2011 Shanghai



American Society  
for Quality

Software Quality  
Professional

international  
requirements  
engineering  
board



CASTB  
Czech and Slovak Testing Board

Copyright 2010 INFOGEM AG

## Initiator Bridge Guard Residence Centre



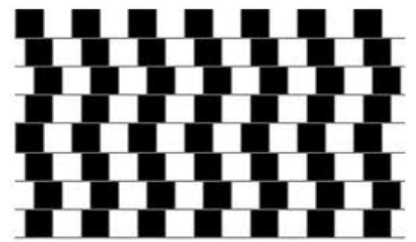
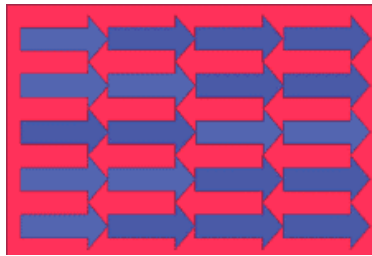
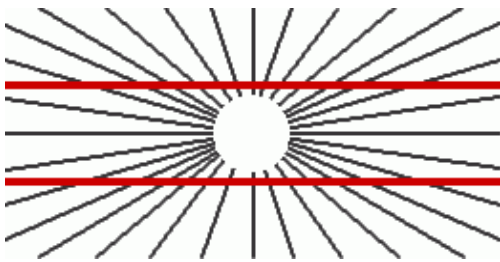
In the year 2001 the Mária Valéria bridge between Štúrovo (Slovakia) and Esztergom (Hungary) was reopened. During its history, this bridge was destroyed and unusable for a longer time than it was actually connecting the two towns. The rebuilt bridge deserves to be saved from further destruction by people. To this aim, mental protection is more important than physical protection. As long as the mental connection between people is intact, the bridge is not endangered.

The post of Bridge Guard requires a person in whose work boundaries of countries or eras are bridged, mental, social, religious or political boundaries are crossed, different scientific fields are connected, or various artistic media are utilized.

Copyright 2010 INFOGEM AG

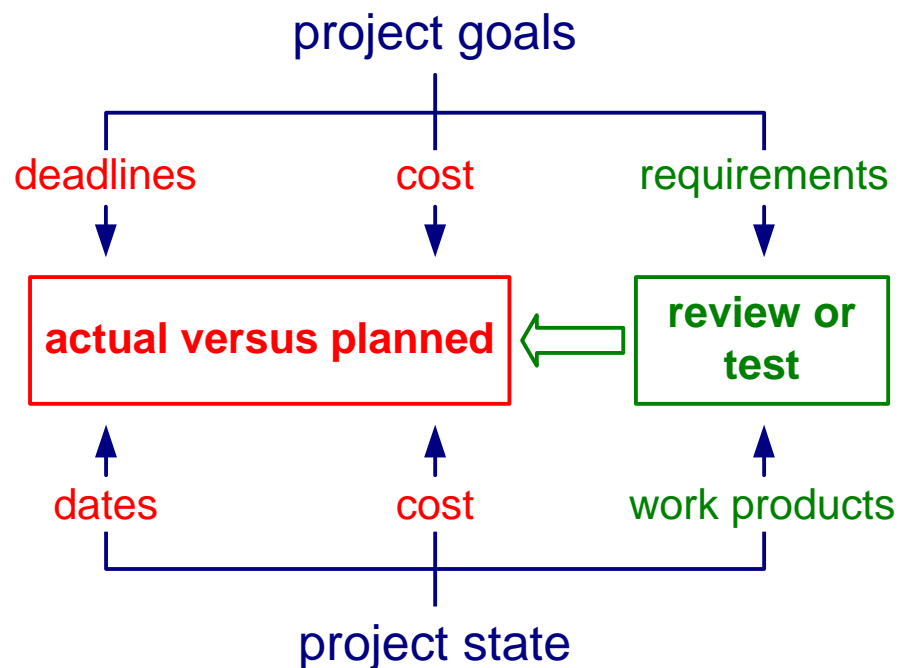
## Six illusions about testing

- I Testing is a hobby of quality people
- II The quickest way to deployment is ping-pong testing
- III Tester don't need to know the context
- IV Integration testing is interface testing
- V Test coverage is a glass box test concept
- VI Test planning is an easy task



Copyright 2010 INFOGEM AG

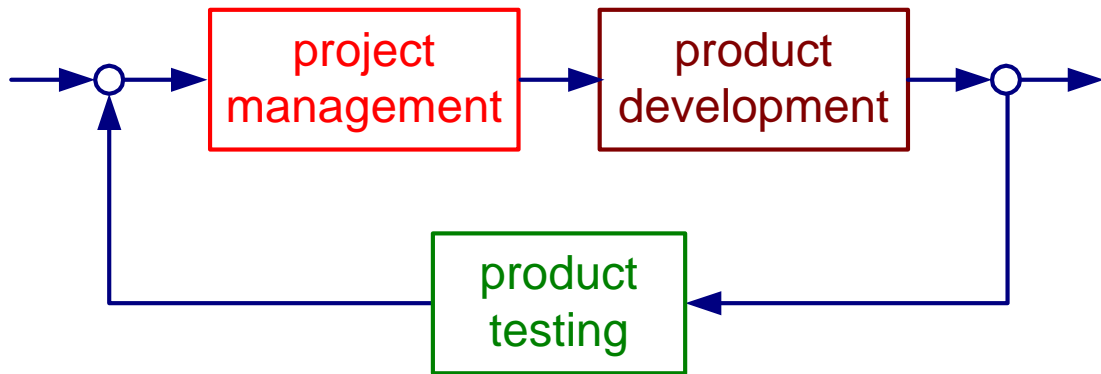
## I Testing is a hobby of quality people



⇒ **without review and test no real progress control**

Copyright 2010 INFOGEM AG

## I Testing is a hobby of quality people

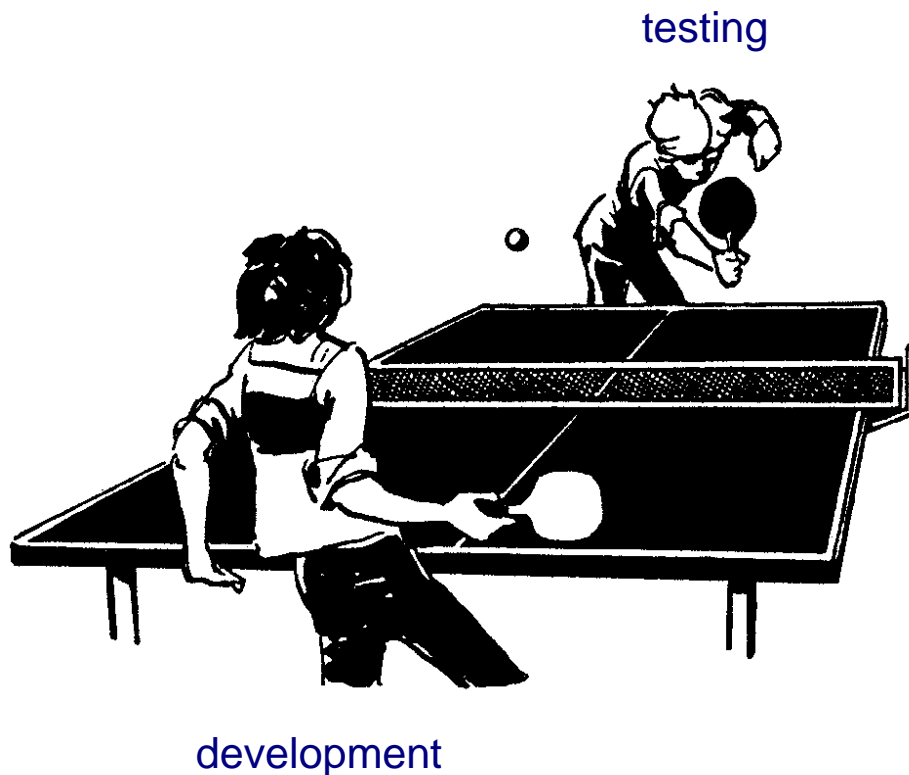


⇒ don't throw defects over the wall to the developer

⇒ every car has only one driver, every project too

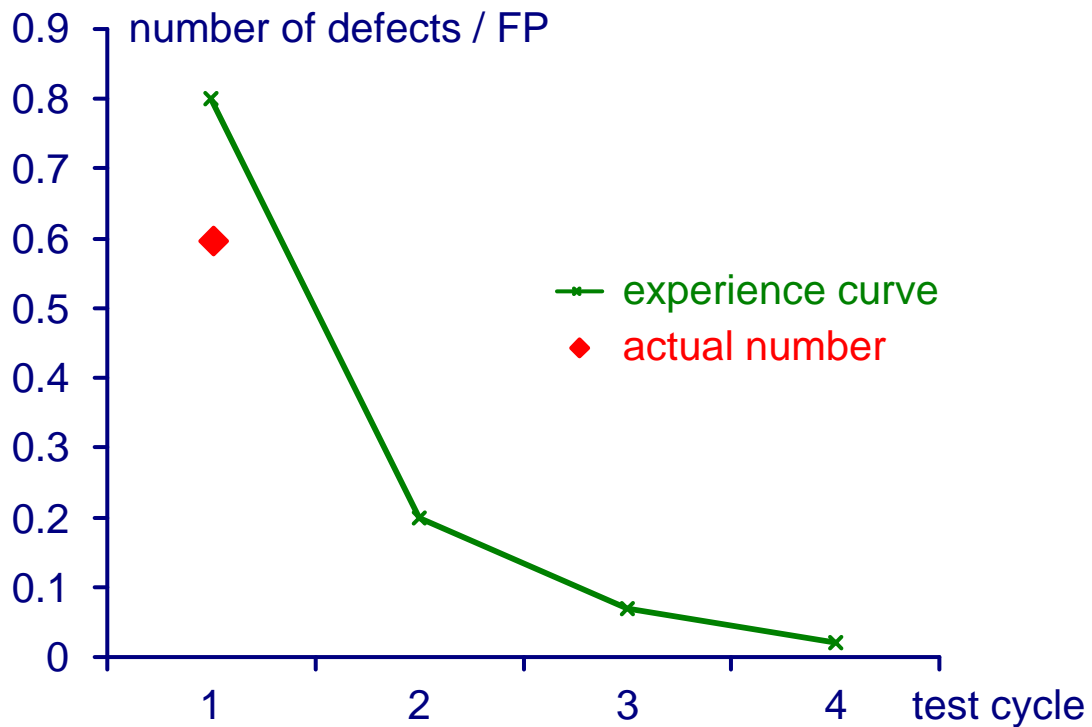
Copyright 2010 INFOGEM AG

## II The quickest way to release is ping-pong testing



Copyright 2010 INFOGEM AG

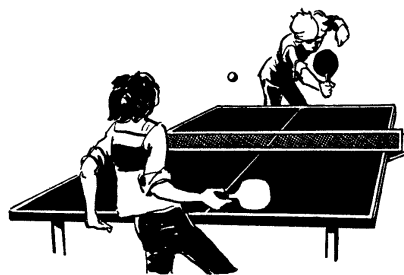
## II The quickest way to release is ping-pong testing



Copyright 2010 INFOGEM AG

## II The quickest way to release is ping-pong testing

.. as soon as the tester detects a defect

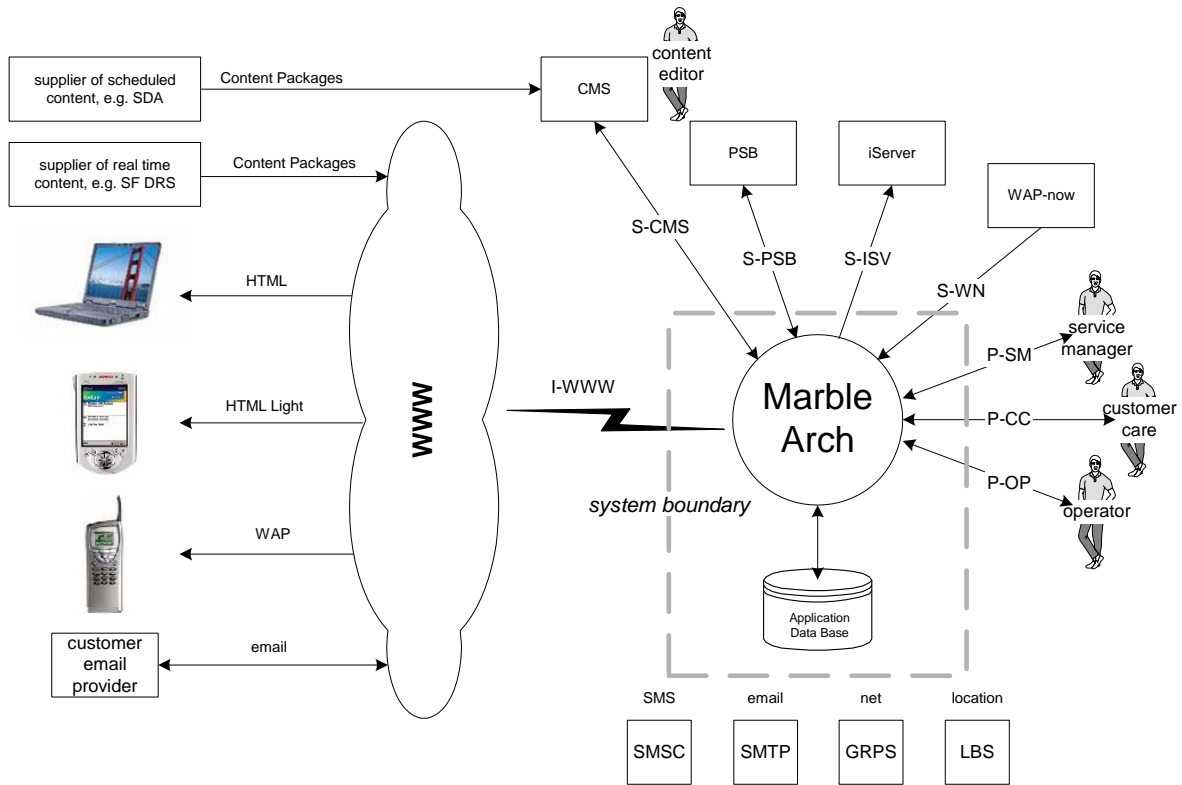


he returns the software to the developer

- ⇒ **we have only one defect to fix ...**
- ⇒ **expensive regression tests**
- ⇒ **if special condition then rucksack;**
- ⇒ **execute all specified test cases, then switch to repair mode**
- ⇒ **state of affairs identified, forecast possible, much cheaper**

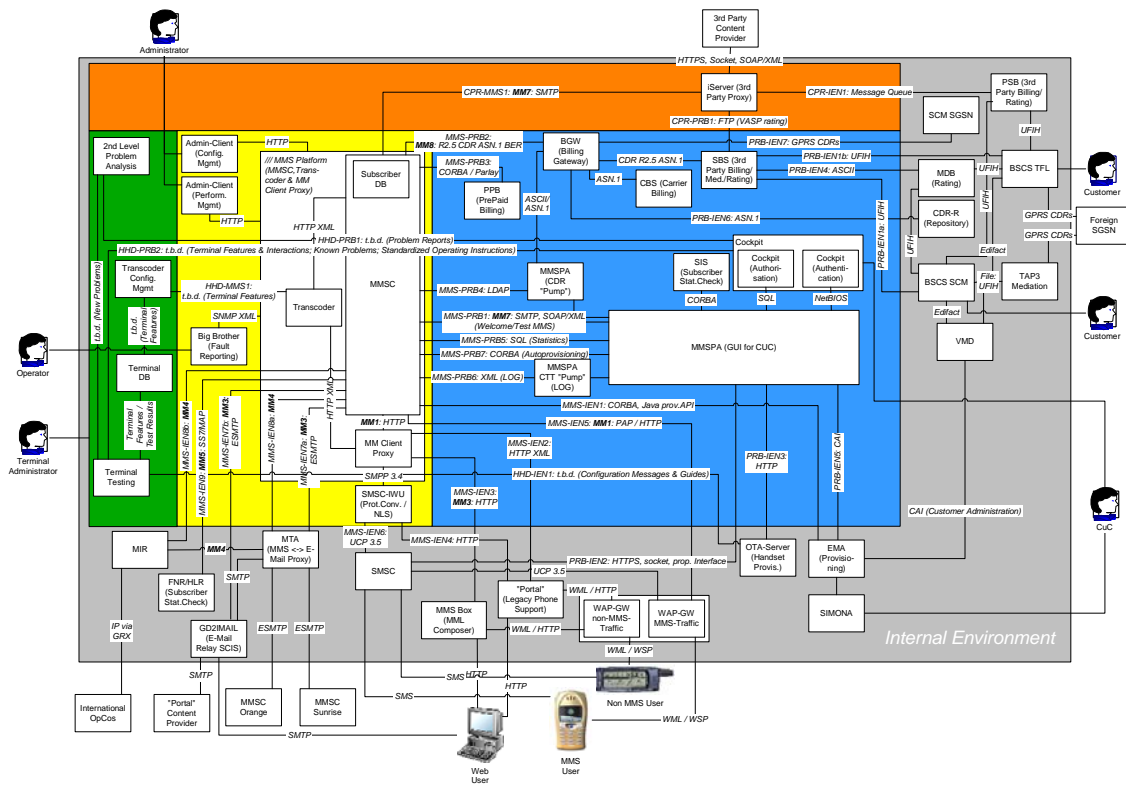
Copyright 2010 INFOGEM AG

# III Tester don't need to know the context



Copyright 2010 INFOGEM AG

# III Tester don't need to know the context



Copyright 2010 INFOGEM AG

## IV Integration testing is interface testing

Integration testing: Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them.

[IEEE 610.12]

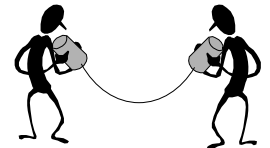
Integration testing: Testing performed to expose faults in the interfaces and in the interaction between integrated components.

Interface testing: Integration testing where the interfaces between system components are tested.

[BS7925-1]

Integration testing is the process of verifying the interaction between system components (possibly and hopefully tested already in isolation).

[SWEBOK 1.0]



## IV Integration testing is interface testing

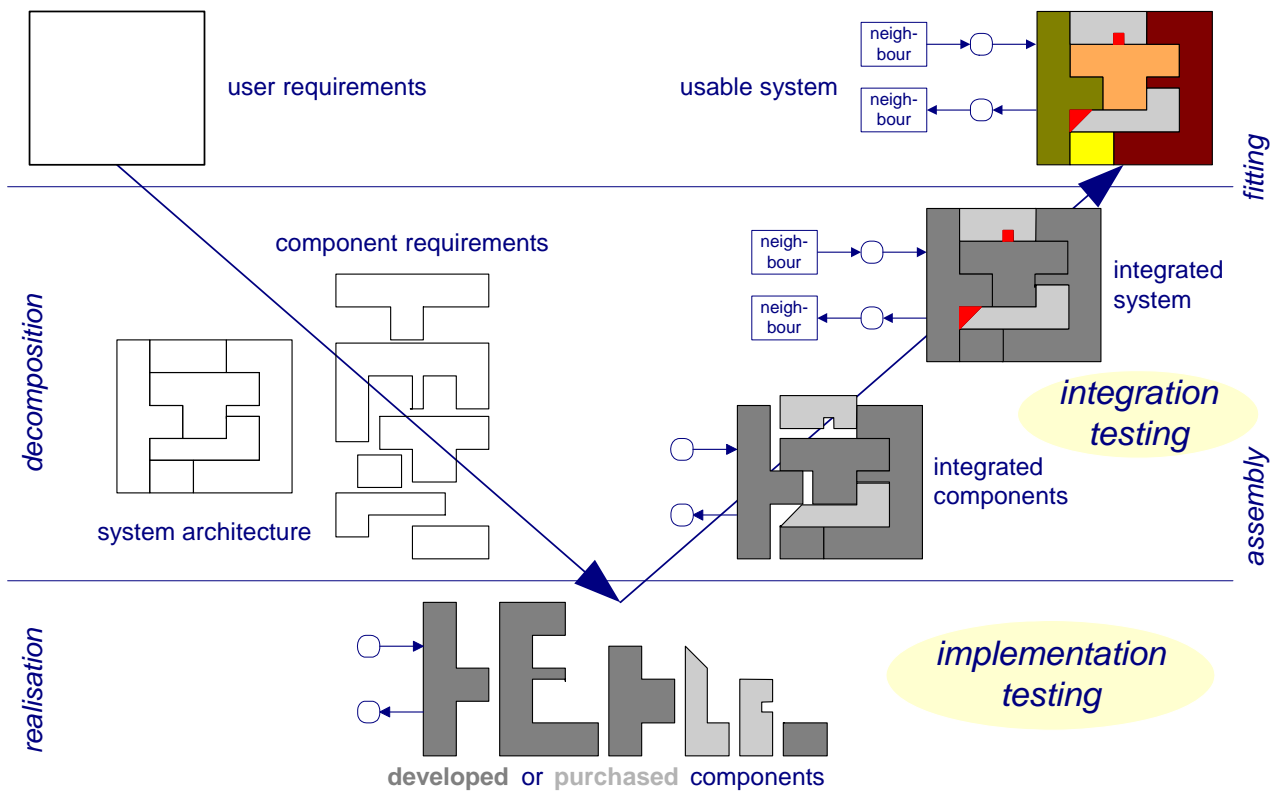
### implementation testing

- testing in which aggregates are tested with the aim to detect defects caused by errors made during **implementation**
- concern is the functionality of the aggregate (unit testing) or the interaction of its parts (interface testing)

### integration testing

- testing in which aggregates are tested with the aim to detect defects caused by errors made during **integration**, e.g.
  - building
  - writing scripts (function test of scripts)
  - integration of components to tiers and these to system
  - integration of components to subsystems and these to system
  - configuration of the system
  - installation of the system in the target environment

## IV Integration testing is interface testing

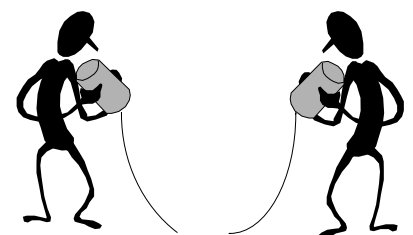


Copyright 2010 INFOGEM AG

## IV Integration testing is interface testing

type of errors integration testing is looking for

- wrong address
- wrong name used
- queue is not set-up
- queue is too small
- file is missing or is in wrong location
- processes are started in a wrong sequence
- a process is not started at all
- wrong setting of configuration parameters or no setting at all
- etc.

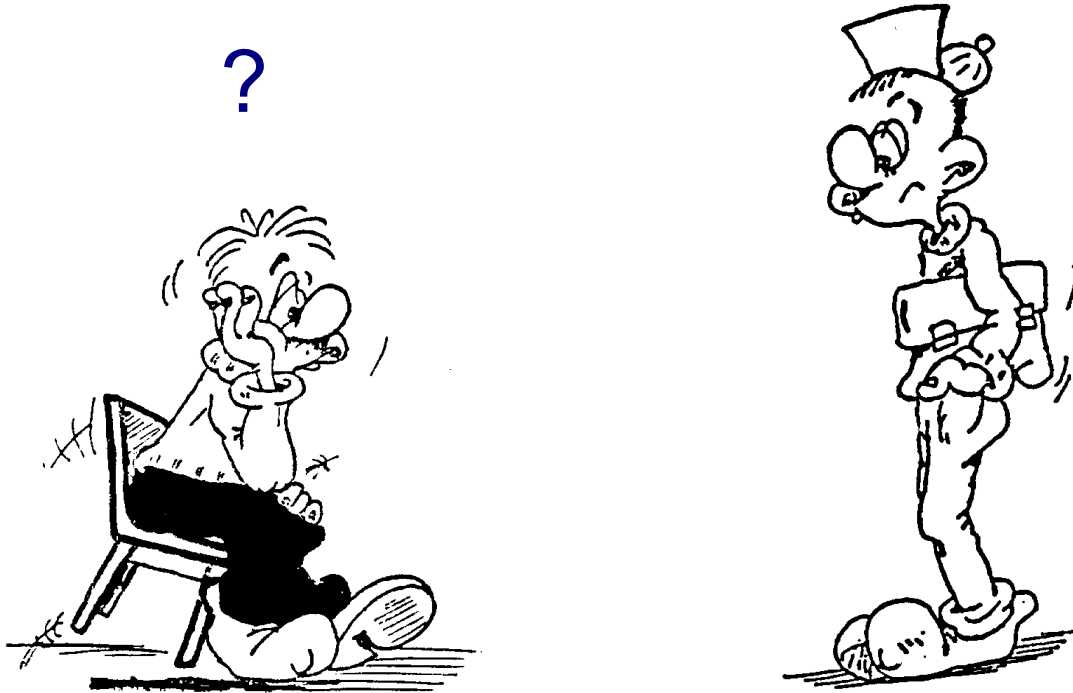


Copyright 2010 INFOGEM AG



## V Test coverage is a glass box test concept

a quite usual conversation ...



Copyright 2010 INFOGEM AG

## Example: Black-box test of the Windows clock



Copyright 2010 INFOGEM AG

## Example: A complete set of test cases (1)

output	test cases		
	1	2	3
analogue time display		X	
digital time display			X
font (28 types)		Arial	TnR
display of the Greenwich time		X	
display of the system time			X
display of the title bar		X	
no display of the title bar			X
display of seconds		X	
no display of seconds			X
display of the date			X
no display of the date		X	
display of information	X		

Copyright 2010 INFOGEM AG

## Example: A complete set of test cases (2)

analogue display of time: 8 test cases

	1	2	3	4	5	6	7	8
time display	gch	gch	gch	gch	sys	sys	sys	sys
title bar display	yes	yes	no	no	yes	yes	no	no
seconds display	yes	no	yes	no	yes	no	yes	no
date display	no	no	no	no	no	no	no	no

digital display of time: 448 test cases

date display is possible: doubles the analogue test cases = 16

28 font types available:  $16 \times 28 = 448$

**total:** analogue display + digital display + info =  
 $8 + 448 + 1 = 457$  test cases

Copyright 2010 INFOGEM AG

## V Test coverage is a glass box test concept

first criterion (3 test cases)

⇒ for all possible types of display at least one of the possible outputs is produced by at least one test case

second criterion (457 test cases)

⇒ all possible combinations of outputs are produced by at least one test case

a possible criterion in between (30 test cases)

⇒ all possible outputs are produced by at least one test case



Copyright 2010 INFOGEM AG

## V Test coverage is a glass box test concept

- testing is a sampling procedure
  - the sample content depends on risks
    - the sample size is defined by the envisaged "confidence level" of the test
- ⇒ coverage defines the sample
  - ⇒ coverage is a target for the test designer
    - ⇒ coverage makes systematic test case selection possible
      - ⇒ coverage determines the extent, thus also the cost of testing
        - ⇒ coverage enables the project leader / software manager to (better) assess the quality of the test



Copyright 2010 INFOGEM AG

## VI Test planning is an easy task

– we do unit testing, integration testing, system testing

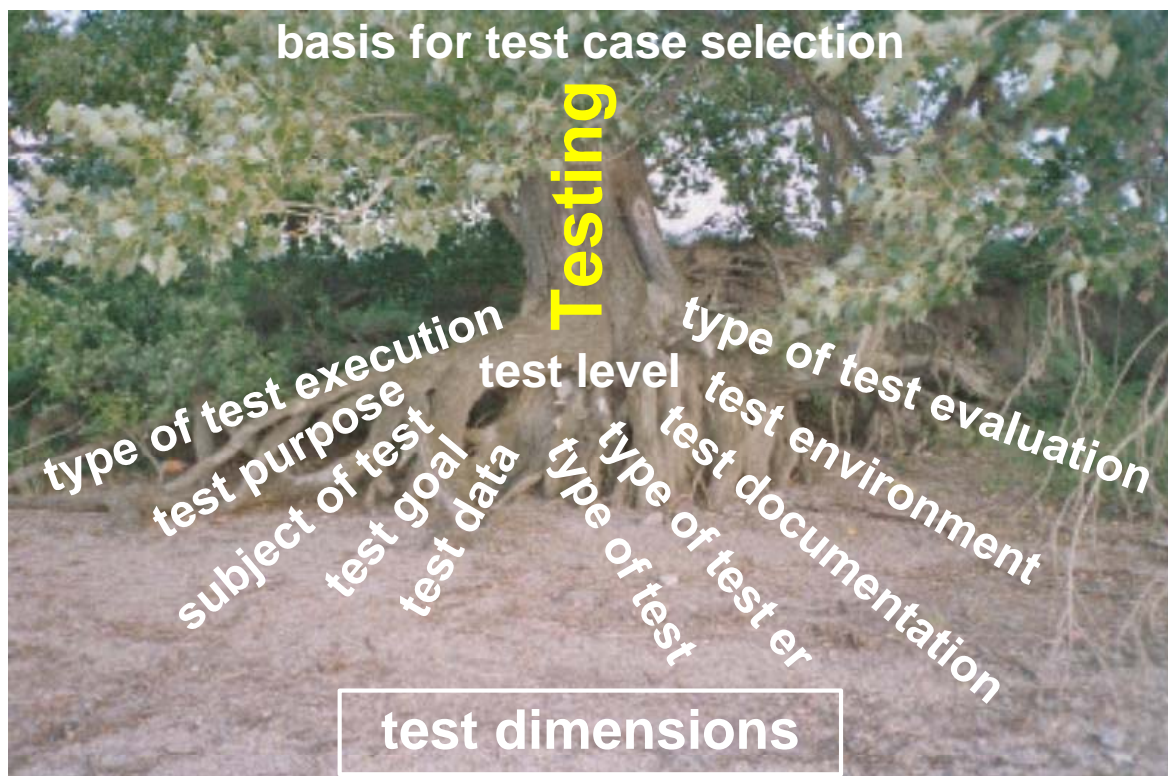
test planning involves

- identify system boundaries and system structure
- define strategy for reviewing, integration, and testing
- analyse risks
- define test objects
- for all test objects define the **test dimensions**
- design the test infrastructure and specify the test harness
- identify all testing activities and estimate the effort
- trade cost and benefit of the tests
- schedule test activities and assign resources

can't be all done at the beginning and not all of what can be done,  
can be defined with the same level of detail

Copyright 2010 INFOGEM AG

## VI Test planning is an easy task



Copyright 2010 INFOGEM AG

## Dimensions of a single test

subject under test	an executable unit (or many)
test level	unit or component or system or an aggregate in between
type of test (error types to look for)	logic, data entry, navigation, fault tolerance, connection, communication, response time, size, etc.
basis for test case specification	artefact used to gather information about possible test inputs and expected output
basis for test case selection	artefact used to define test coverage criteria used to assess the completeness of the selected test case set
test environment	development or integration or test or production
tested configuration	configuration of the subject under test
test goal	extent of error type and tested configuration coverage
test execution	manually using a checklist, using test procedures, with automatic test logging, completely automated, etc.
tester	user, test engineer, ignorant, expert, etc.
test evaluation	compare with specification (basis), compare with assured results, etc.
test record	completed checklist, manual test log, automatic test log, etc.

Copyright 2010 INFOGEM AG

## Example: System test planning with variations

WEB configuration	possible values				cardinality
OS	Mac	Vista	Win 7		3
browser	Safari	IE	Firefox		3
registered user	no	yes			2
locked user	no	yes			2
user language	German	French	Italian	English	4
WAP configuration	possible values				cardinality
operator	we	foreign			2
device brands	5 new	15 legacy			20
registered user	no	yes			2
locked user	no	yes			2
user language	German	French	Italian	English	4
				<b>WEB</b>	<b>WAP</b>
	minimal number of variations			4	20
	theoretically maximal number of variations			144	640

Copyright 2010 INFOGEM AG