Software Test

Lesson 2
Test Basics
v1.2

Uwe Gühl, Jittat Fakcharoenphol

Fall 2007/ 2008

Contents (1/3)

- Test Basics
  - Definitions of terms in Testing Environment
    - Software quality
    - Verification and Validation
    - White Box Test
    - Black Box Test
    - Defect / Bugs
    - Prioritization
    - Test Case (functional / non-functional)
    - Test Scenario
    - Test Data
    - Release Management
    - Reporting – Key figure
    - Test automation

Contents (2/3)

- Test Basics
  - Test Philosophy
  - Test in Software Development Processes

Contents (3/3)

- Test Basics
  - Test Stages
    - Unit Test
    - System Test
    - System Integration Test
    - NFR (Non Functional Requirements) Test
    - UAT (User Acceptance Test)
  - Roles and specialization
  - Input / Output parameter
  - Tools

Test Basics
Definitions of Terms

- Software quality [Wik07]
- Discussion: What means (software) quality?
  - "everyone feels they understand it" (Scott Pressman)
  - Software quality characteristics (Steve McConnell)
    - external - those parts of a product that face its users,
    - Internal - those that do not
  - "a product's quality is a function of how much it changes the world for the better” (Tom DeMarco)
  - "Quality is value to some person" (Gerald Weinberg)

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- ISO 9126 is an international standard for the evaluation of software quality – focusses on the product.

- **Conformance** applies to all following characteristics to evaluate in a specific degree, how much of the agreements got fulfilled

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Functionality - A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.
  - Suitability: Does the software the specified tasks?
  - Accuracy: e.g. the needed precision of results
  - Interoperability: cooperates with specified systems
  - Compliance: ...with conditions / regulations
  - Security: No unauthorized access possible

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Reliability - A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
  - Maturity: Minor breakdowns because of defects
  - Recoverability: If there is a breakdown, how long does it need to recover – how much time / effort is needed (including data!)?
  - Fault Tolerance: Can the system handle unexpected inputs?

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Usability - A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.
  - Learnability: Effort to use, for Input and Output
  - Understandability
  - Operability

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Efficiency - A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.
  - Time Behaviour: Response time, processing time, throughput
  - Resource Behaviour: Usage of RAM, disk space, energy

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Maintainability - A set of attributes that bear on the effort needed to make specified modifications.
  - Stability: What happens after a power cut?
  - Analyzability: Monitoring the system
  - Changeability: Changes at runtime possible?
  - Testability: E. g. possible to reproduce activities?

Test Basics
Definitions of Terms

Software quality – ISO/IEC9126 [Wik07]

- Portability - A set of attributes that bear on the ability of software to be transferred from one environment to another.
  - Installability: Effort to install a system in a specific environment
  - Replaceability: With a specific different software (compatibility of data)
  - Adaptability: E. g. move on another operating system

Test Basics
Definitions of Terms

- Verification & Validation
  - Verification
    Did we build the right product?
    Verification is defined as the "demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle." *
  - Validation
    Did we build the product right?
    Validation is the "determination of the correctness of the final program or software produced from a development project with respect to the user needs and requirements. Validation is usually accomplished by verifying each stage of the software development life cycle." *

* Adrion, W. Richards, Martha A. Branstad, and John C. Cherniavsky. "Validation, Verification, and Testing of Computer Software," Computing Surveys, June 1982, pp. 159-192.

Test Basics
Definitions of Terms

- White Box Test
  - Testing with knowledge of the internals of the program [KBP01]
  - Test Cases are out of the program, not out of the Specification
  - Several methods could be tested after creation – without any relation to the utilization
  - Example: Activity driven testing with test cases , which check sufficiency criteria
    - Line coverage: Execution of all source code lines
    - Command coverage: Execution of all commands

Test Basics
Definitions of Terms

- Black Box Test
  - Testing external behaviour of a program based on specification / requirements
  - Idea: Feeding specific input, expecting specific output
  - Tester has no knowledge of the internals
  - Implementation not considerable, only outside behaviour is important

Test Basics
Definitions of Terms

- White Box Testing
  - Less organizational effort
  - Automation easy
  - Higher code quality

Suggestion: Study http://tynerblain.com/blog/2006/01/13/software-testing-series-black-box-vs-white-box-testing/

- Black Box Testing
  - Good Testing of the complete software
  - Review of specification
  - Independent from implementation
  - Test focus only on specification
    Less quality of specification -> less quality of test results

Test Basics
Definitions of Terms

- Gray Box Testing / Behavioural Testing [KBP01]
  - Testing external behaviour like Black Box Testing
  - Test Strategy based partly on internals of a software
  - Idea: If you know something about the inside, you can test it better from outside
  - Important with Web and Internet applications

Test Basics
Definitions of Terms

- Defect
  - Non-Compliance of a specified requirement – Difference between target and actual [Sol07]
  - Something is definitely wrong with the product [KBP01]
  - Distinguish:
    - Specification fault
    - Software defect
    - Environment failure
    - Interface defect
    - Error in the Test Case (Test Scenario)
    - Error in test data

Test Basics
Definitions of Terms

- Defect
  - Severity-Level
    - 1 – very high: Data loss, not usable
    - ...
    - n – very low: disfigurement
  - Priority-Level (concerning fixing)
    - 1 – very high: Fastest fixing necessary
    - ...
    - n – very low: subordinated handling: Acceptance in „open points / Proposals“
    - Special status: Defect must not be fixed

Test Basics
Definitions of Terms

- Prioritization
  - ... is the basic of testing!
    - Why?
      - Time problems
      - Focusing on critical areas
    - Which criteria are important for prioritization?
      - Complexity
      - Importance
      - Specification coverage
    - How should I prioritize?
      - Identify most important business processes
      - Identify most important use cases

Test Basics
Definitions of Terms

- Test Case
  - Sequence of steps consisting of actions to be performed on the system under test [Bla04]
  - is the “basic unit” in Testing ⚠
  - serves to validate the functionality and to confirm the realization of a requirement
    - functional
    - non functional (quality criteria)
  - originates typically out of an Use Case

→Usually NFR-Test Cases are taken from regular Test Cases, if so simplification

Test Basics
Definitions of Terms

- Test Case
  - describes the role who should execute it
  - contents Test Steps with
    - Activities of the tester
    - Input values
    - Expected output values
  - describes preconditions and postconditions

Test Basics
Definitions of Terms

- Test Case – Example

Test Case name	„IU22_Create-Object“
Test Case ID	7
Priority	1
Test classification	Standard
Preparation Hours	1
Execution Hours	1
Description	Creation of an Object. The user must select an object He has to decide which specific kind ...
Risk	Without Creation of objects Software can't be used
Version	01
is Test Case Chain	[ ]

Test Basics
Definitions of Terms

- Test Case – Example (cont.) Condition
  - Goal Creation of a new object
  - Prerequisites Following objects must be available in database to execute this test case:
    * object A
    * object B
  - Remarks Function „select module“ is described in Test Case „IU21_Display-Object“

Test Basics
Definitions of Terms

- Test Case – Example (cont.) Test Steps

StepNo.	Description	Comments	Expected Result
10	Select an object in the tree structure		Selected Item will be highlighted
20	Choose „create“		Dialog box opens
30	Choose radio button		
40	Enter Obj ID		

Test Basics
Definitions of Terms

- Test Scenario
  - Synonym: Test Case Chain, Test Suite [Bla04]
  - Collection of logically related test cases [Bla04]
  - Test Scenarios are used to test processes – were process requirements implemented completely and correct?
  - A Test Scenario is a combination of – possibly modified (as a rule simplified) – Test Cases
  - A Test Scenario arises typically from a Business Scenario (Business Use Case)

Test Basics
Definitions of Terms

- Test Scenario
  - Test Scenarios typically test the data flow in the system
  - Tests usually don't end with testing the system itself only
    - Interface test (e.g. with MQSeries from IBM)
    - System simulator

Suggestion: Study
“Mocks Aren't Stubs” at
http://www.martinfowler.com/articles/mocksArentStubs.html

Test Basics
Definitions of Terms

- Test Scenario Example “User logs in a vocabulary training system and does 1st lecture”
  - Test case 17 „First login“
  - Test case 33 „Choose Language“
    - Source language „English“, Target language „Thai“
    - Level „Starter“
    - Lesson „Vacancy“
    - Learning strategy 1
  - Test case 46 „First lesson“
  - Test case 103 „Follow-up lesson“
  - Test case 132 „Score“
    - Choose Bar Chart

Test Basics
Definitions of Terms

- Test data
  - All data needed for testing
  - Discussion
    - Based on Business Object Data Model (BuOM) or Physical Data Model (PhDM)
    - Artificial data or based on real business data, e. g. out of legacy systems
    - Which test data are included with delivery?
    - Feed of Test data
    - Remove of Test data („nacked system“)

Test Basics
Definitions of Terms

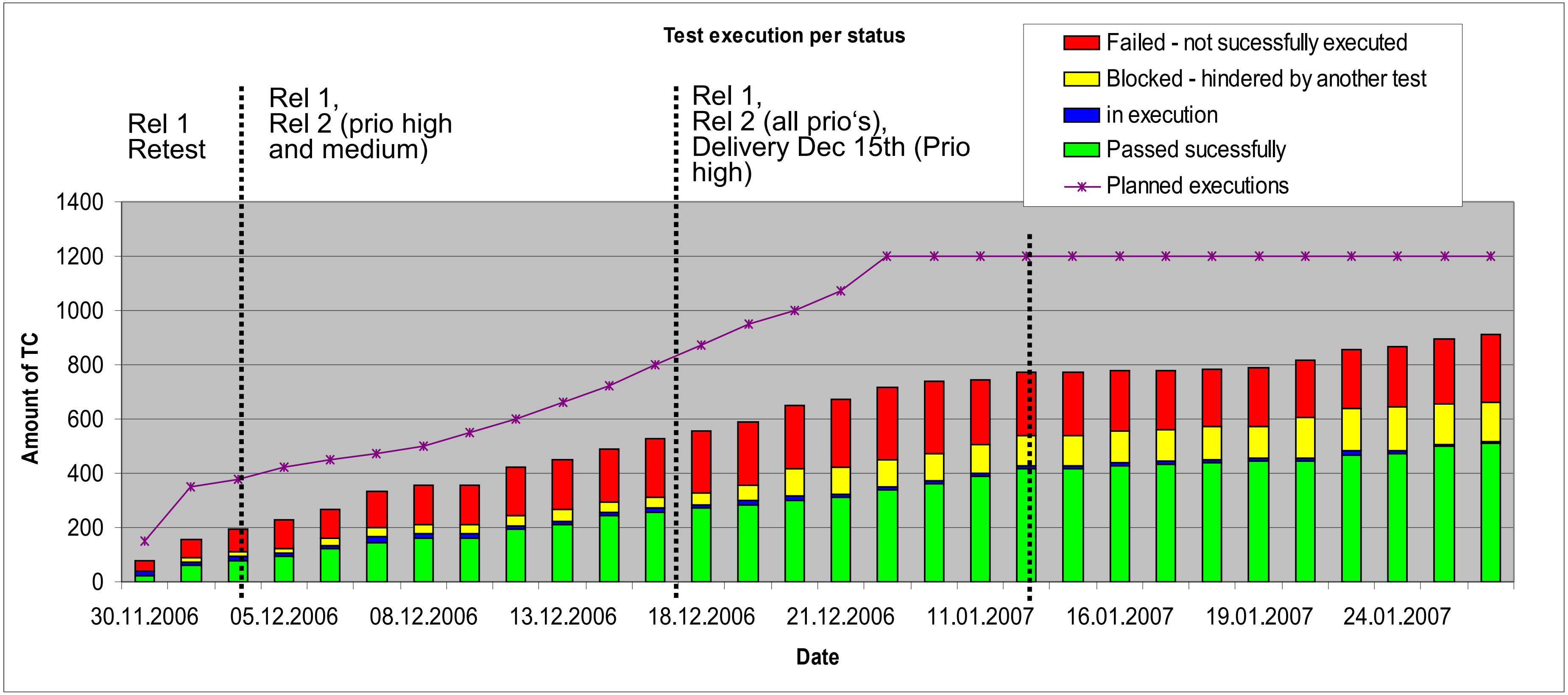
- Release Management
  - Agreement, when which version / release / patch gets delivered
  - Scope of a release
  - Release note
    - Description of contents / new functionality
    - Fixed defects
  - Name convention

Test Basics
Definitions of Terms

- Reporting
  - Expected: Periodical statements concerning
    - Quality (of software, specification, test cases)
    - Test progress
    - Test coverage
    - Status concerning critical areas
  - Hint: Discuss reporting criteria **in advance** with
    - Customer
    - Software Developer
    - Specification Team
    - Operation

Test Basics
Definitions of Terms

- Reporting – Example (1/2)



Remark: According to our plan (1200 TC) we have executed 966 Test Cases. The gap is approx. 20 %. Reason of less increase in the amount of test execution is mainly the necessary retesting of fixed and delivered defects.

Test Basics
Definitions of Terms

- Reporting – Example (2/2)
  - Coverage
    - Delivered SR1 covers 123 out of 124 Use Cases
    - 966 of 1200 Test Cases executed
  - Defects
    - 303 open defects (32 Severity Level 1, 164 Severity Level 2, 107 Severity Level 3)
    - 642 final defect status (60 Change Requests, 427 Closed, 110 Cancelled, 35 Duplicated, 10 Deferred)
  - Most important statements
  - Risks

Test Basics
Definitions of Terms

- Test automation
  - An executable program or a script executes automatically Test Cases with
    - Execution of defined Test steps
    - Corresponding data input and
    - Control of the results
  - The results get logged and analysed
  - Special Test tools support the test automation

Test Basics
Definitions of Terms

- Discussion: Test automation
  ... for the management
    - No time for testing?
    - Test Capacity is too expansive?
    - Tests are too complicated?
  - Test automation is **THE** solution, cause ...
    - quick
    - low cost
    - simple

Test Basics
Definitions of Terms

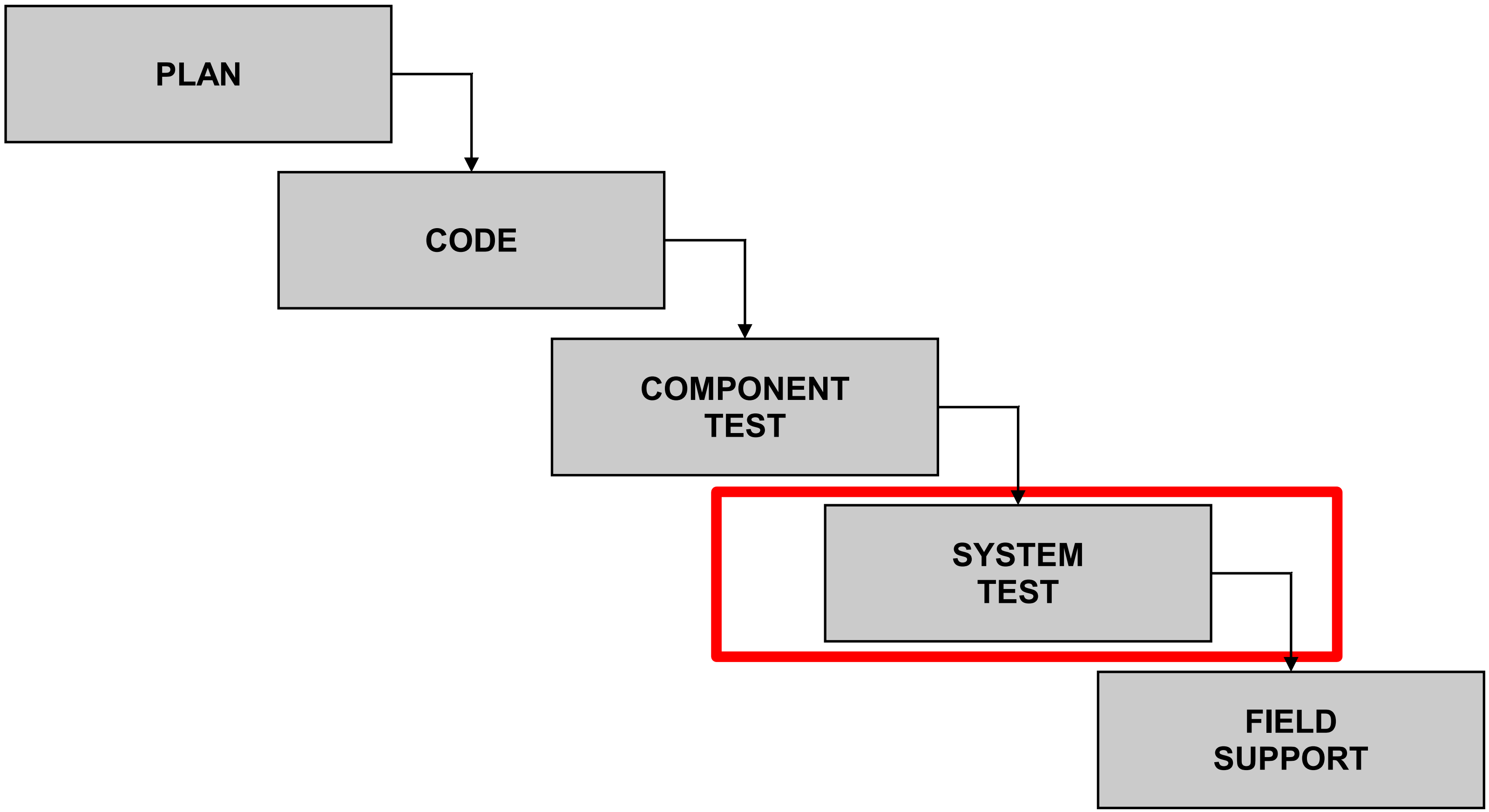
- Discussion: Test automation
  ... for the management
  - **WRONG!** The contrary is right!
    Test automation in general is
    - Time consuming
    - Expensive (in the beginning)
    - Complex (compare to software development)

Test Basics
Test Philosophy

- Reference – Test refers (esp. Black Box Testing)
  - always on the specification and
  - not on the delivered product or code
- Prioritization – Important in Testing to
  - test the important scope first
  - achieve as early as possible a high coverage
  - detect critical defects as soon as possible
  - minimize the risk of not detected critical defects at the end of testing
  - support the defect fixing in the best way

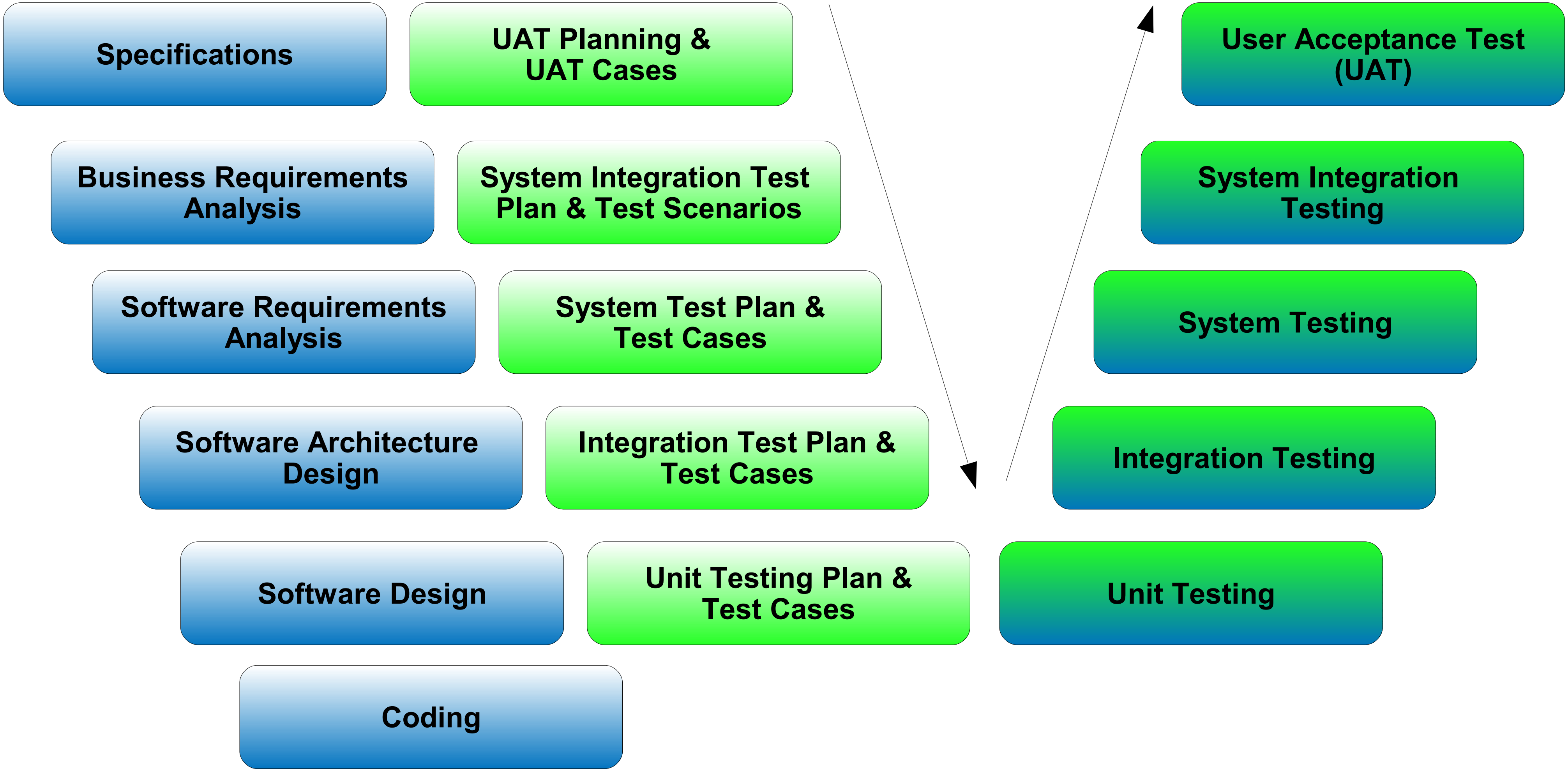
Test Basics
Test in Software Development Processes

- Waterfall model [Sol07]



Test Basics
Test in Software Development Processes

- V-Model



Specifications

UAT Planning & UAT Cases

User Acceptance Test (UAT)

Business Requirements Analysis

System Integration Test Plan & Test Scenarios

System Integration Testing

Software Requirements Analysis

System Test Plan & Test Cases

System Testing

Software Architecture Design

Integration Test Plan & Test Cases

Integration Testing

Software Design

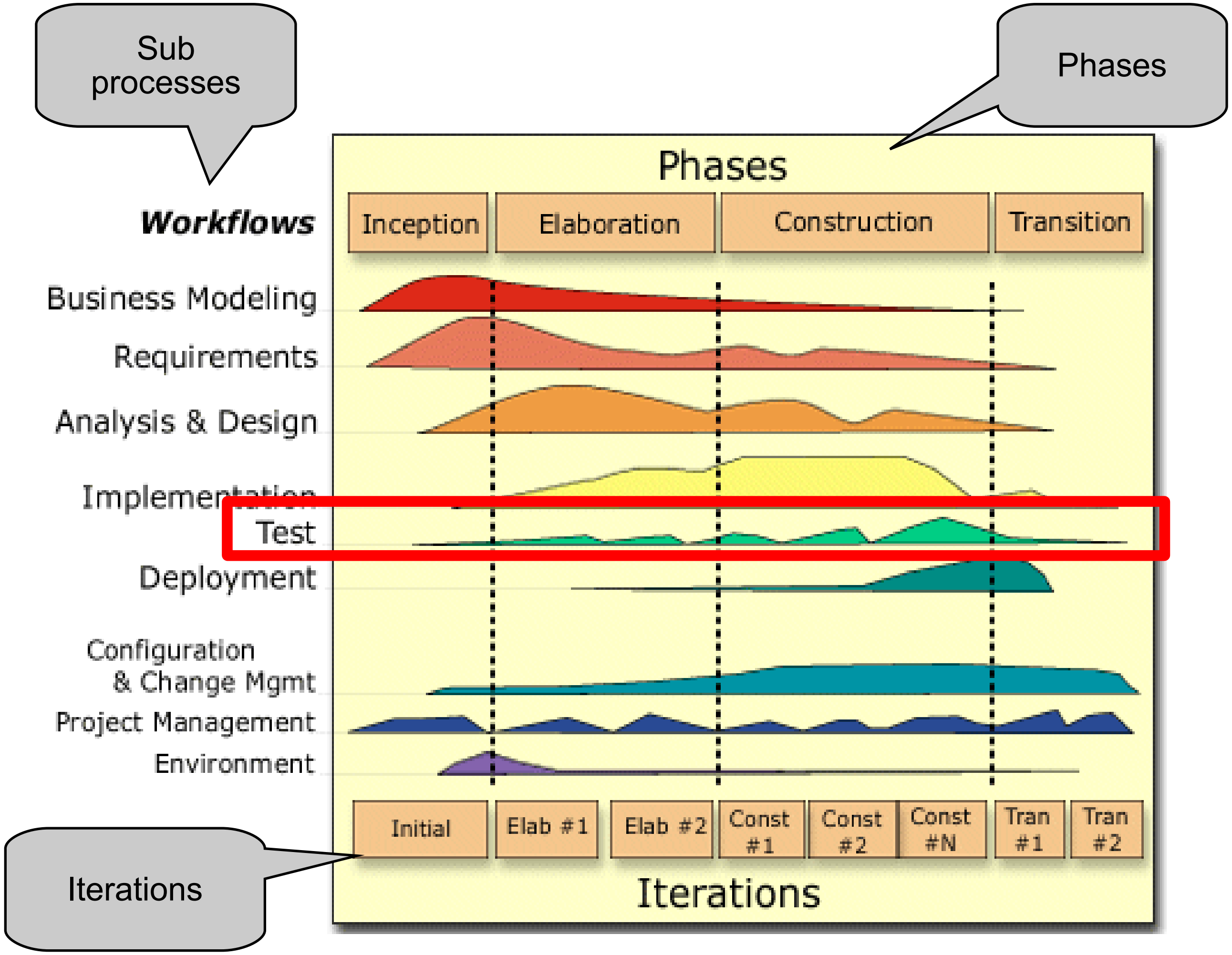
Unit Testing Plan & Test Cases

Unit Testing

Coding

Test Basics
Test in Software Development Processes

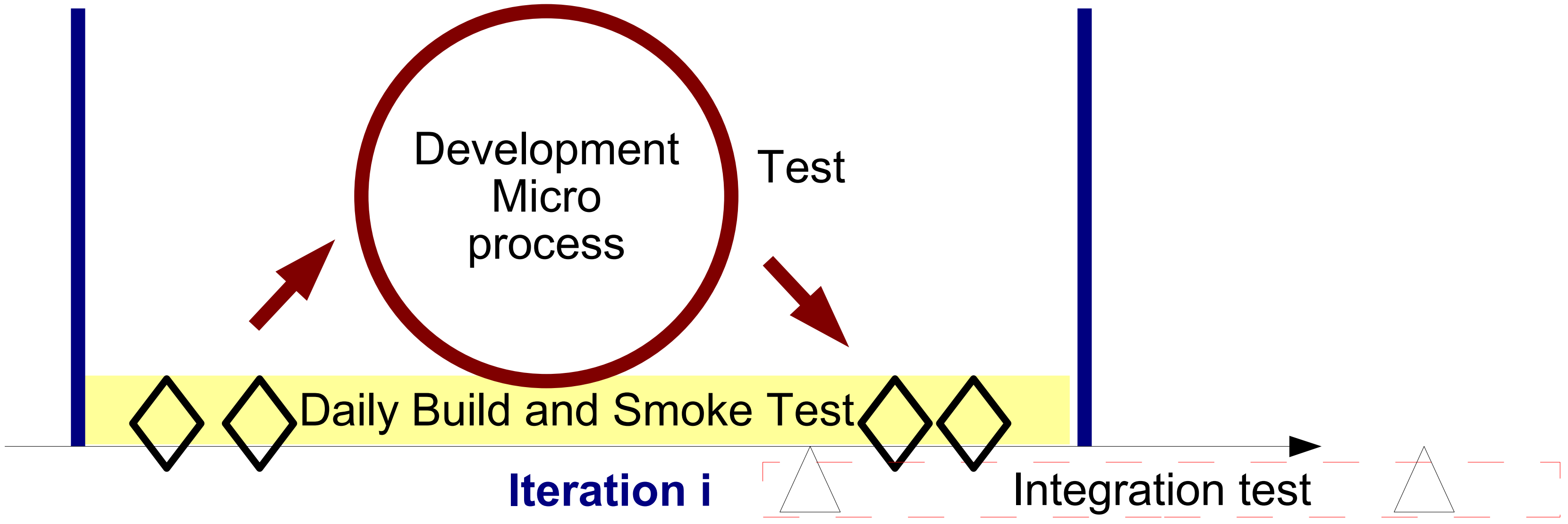
- Rational Unified Process [Wik07a]



Test Basics

Test in Software Development Processes

- Example Agile Software Development [OW06] Here we find testing in following parts:
  - Micro process circle - Testing to measure success
  - Daily Build and Smoke Test
  - Ongoing Integration Test



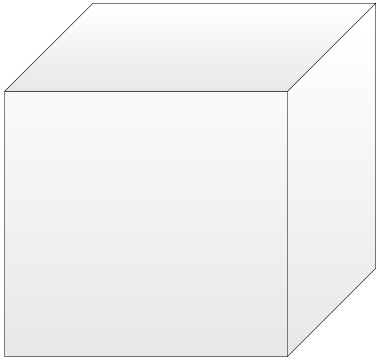
Test Basics

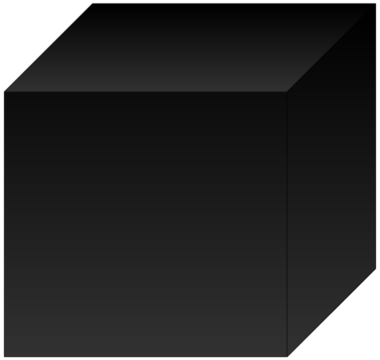
Test in Software Development Processes

- Summary
  - Test follows the software development process chosen in the project
  - We follow in this lecture a(n idealized) derived V-Model
  - Principally is early testing helpful for the project: The earlier one tests, the earlier one gives feedback, if the realized software follows the requirements including the possibility to react early

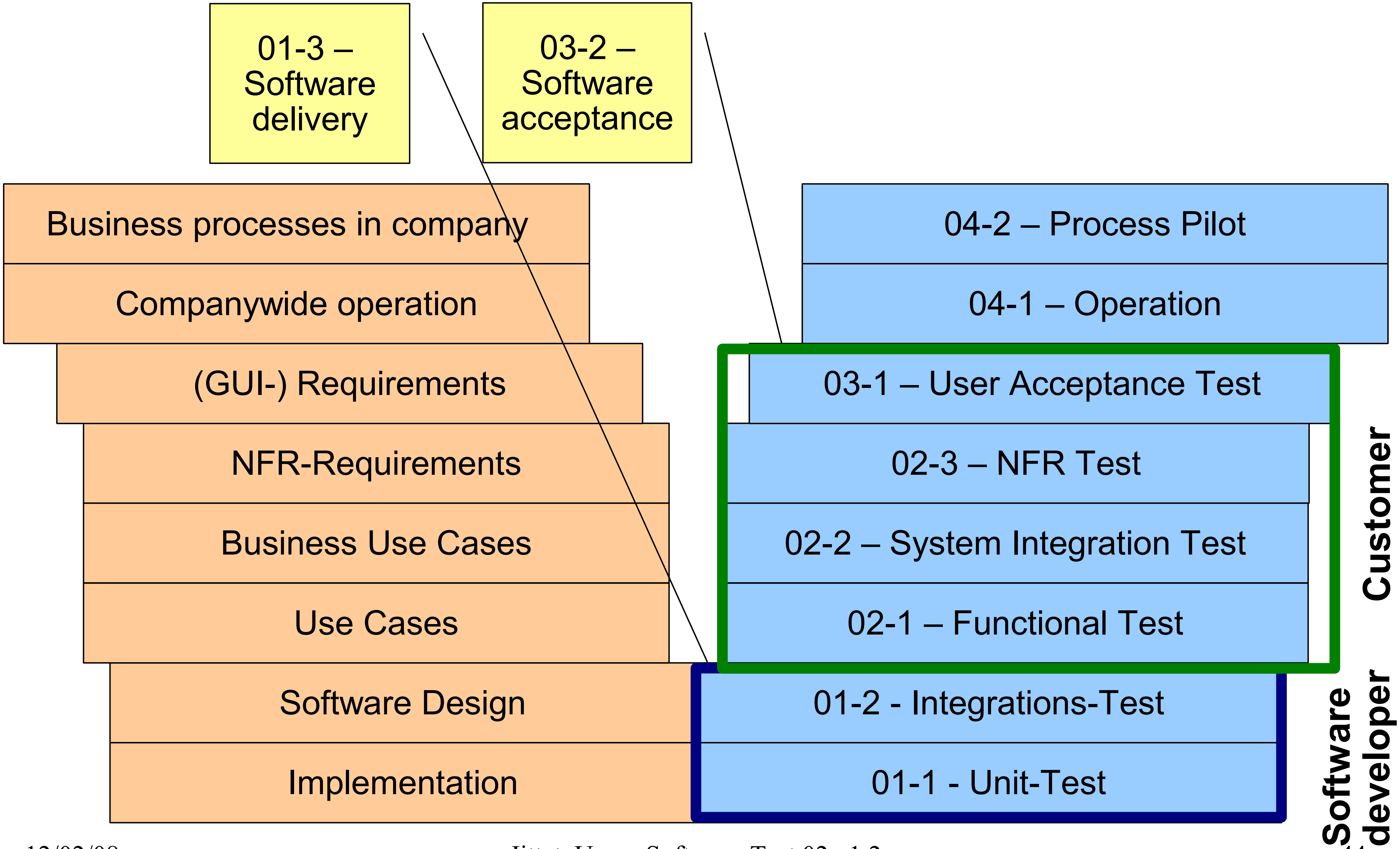
Test Basics
Test Stages

- Concerning Test we distinguish between
  - Software Developer Tests
    - before delivery of a software
    - including Unit tests and Integration tests
  - Customer Tests
    - after delivery of a software
    - to verify the coverage of
      - procedural requirements
      - functional requirements
      - non functional requirements





Test Basics
Test Stages



01-3 – Software delivery

03-2 – Software acceptance

Business processes in company

Companywide operation

(GUI-) Requirements

NFR-Requirements

Business Use Cases

Use Cases

Software Design

Implementation

04-2 – Process Pilot

04-1 – Operation

03-1 – User Acceptance Test

02-3 – NFR Test

02-2 – System Integration Test

02-1 – Functional Test

01-2 - Integrations-Test

01-1 - Unit-Test

Customer

Software developer

Test Basics
Test Stages

- 01-1 Unit Test
  - Goal: Testing of components (e.g. classes or packages in Java) concerning functionality, robustness and efficiency, based on the software code
  - Scope: Usually the developer of a component creates and executes the corresponding test cases
  - There are special test frameworks (e.g. for Java JUnit)

... following [Sol07]

Test Basics
Test Stages

- 01-2 Integration Test
  - Goal: Identification of defects in the interaction of components; focus is interface formats and data exchange [Sol07]

Test Basics
Test Stages

- 01-3 Software delivery
  - First after software delivery to the customer we test, if the software could be installed
  - Usually a smoke test follows

Not the most sophisticated way to tell if something is working properly, but it's a sure guarantee that it isn't.

Test Basics
Test Stages

- 02-1 Functional Test
  - Synonyms Black Box or behavioural testing [KBP01]
  - Goal: Verification of functional requirements
  - Basics: Specification, requirements, Use Cases
  - Scope: Execution of Test Cases

Test Basics
Test Stages

- 02-2 System Integration Test
  - Goal: Verification of Business processes
  - Basics: Business-Specification
  - Scope: Execution of Test Scenarios

Test Basics
Test Stages

- 02-3 NFR-Test
  - Goal: Validation of NFR-requirements
  - Basics: NFR requirements
  - Scope
    - Operability Test
    - Security Tests
    - Load tests, performance tests
      - Response time
      - Possible number of users
    - Failure tests
      - Availability?

Test Basics
Test Stages

- 03-1 User Acceptance Test (UAT)
  - Goal: Make acceptance of software possible
  - Basics: Specification, especially concerning usability
  - Scope: Testing of Usability

Test Basics
Test Stages

- 03-2 Software acceptance
  - Goal: Acceptance following specified criteria
    - Test coverage
    - Test execution
      - Test Cases following prioritization
      - Test Scenarios following prioritization
    - UAT-results (Usability)
    - Number of detected defects with weighting (Severity level)
  - A report contents the test results and is basics for acceptance

Test Basics
Test Stages

- 04-1 Operation
  - Goal: Regularization of transition
  - Operation must control the introduction of the software, e.g. a step-by-step concept

Test Basics
Test Stages

- 04-2 Process Pilot
  - Goal: Verification of new software / new processes, before everybody uses
  - Basics: Management decision
  - Consider migration!

Test Basics
Roles and Specialization

- Test Manager
  - Qualification
    - Test Know-how
    - Experience as project manager
  - Tasks
    - To aid Test Team members, so they could do a good job

Test Basics
Roles and Specialization

- Test Manager
  - Tasks
    Responsible for [Sol07]
    - Identification of all testing actions to be done
    - Estimation of the effort
    - Planning of all necessary activities
    - Management of all people involved in testing
    - Reporting
    - Communication of Testing results

Test Basics
Roles and Specialization

- Test Planer
  - Qualification
    - Deep Test Know-how, long years experience
  - Tasks
    - Test planning, initial and follow-up
    - Controlling

Test Basics
Roles and Specialization

- Tester
  Synonyms Test Engineers, Test Designer
  - Discussion
    - Best people should test!
    - Software Testers are real experts after finishing the tests
      - They know the software: Strengths and weaknesses
      - They could work as multiplier, introducing, and train

Test Basics
Roles and Specialization

- Tester
  - Qualification
    - IT Know-how (Use Cases, UML)
    - Test Know-how
    - Expertise about the subject to be tested
  - Tasks
    - Test preparation
      - Review of specification, point out faults / open issues
      - Generation of Test Cases, and Test Scenarios
      - Combine Test data with Test Cases / Test Scenarios
    - Test execution
      - of Test Cases, Test Scenarios; reporting of defects

Test Basics
Roles and Specialization

- Test Data Manager
  - Qualification
    - Data base expert (Data modelling know-how)
    - Test Know-how
  - Tasks
    - Test data strategy / concept
    - Test data research
    - Test data generation
    - Mapping of Test data to Test cases / Test scenarios
    - During Test execution supporting with test data

Test Basics
Roles and Specialization

- NFR-Test Manager
  - Tasks: Defining of a strategy, planning, organizing, execution of performance test, load tests, security tests, breakdown tests
- Test automation expert
  - Tasks: Test automation strategy, choice of tool, preparation and execution (scripting, delivering reports)

Test Basics
Roles and Specialization

- Defect Manager
  - Tasks: Choice of tool, defect collection, defect tracking, moderation of defect meetings, control of release management
- Environment Manager
  - Tasks: Providing Test environment – at a time for corresponding tests, accept software, installing it, running smoke test, keep the software „ run capable“

Test Basics
Roles and Specialization

- Qualification
  - QAI (Quality Assurance Institute Worldwide, USA) [QAI07]
    - CSTE – Certified Software Tester
    - CSQA – Certified Software Quality Assurance
  - ISTQB (International Software Testing Qualification Board, Germany) [IST07]
    - “Foundation” and “Advanced” Certificates

Test Basics
Roles and Specialization

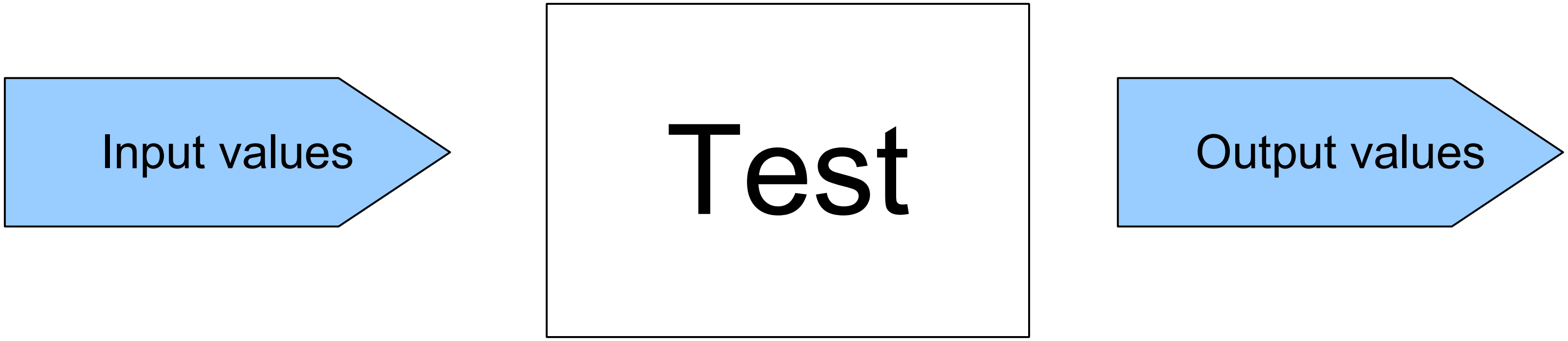
- Stakeholder
  - Customer
  - Creators of the specification (they know the requirements best)
  - Software developer
  - Users („old stager“ are very valuable! Processes)
  - Operation (Architectural requirements)

- A helping tool: RACI-Matrix*

* Responsible, Accountable, Consult, Inform
To identify and to define roles of people involved in the project

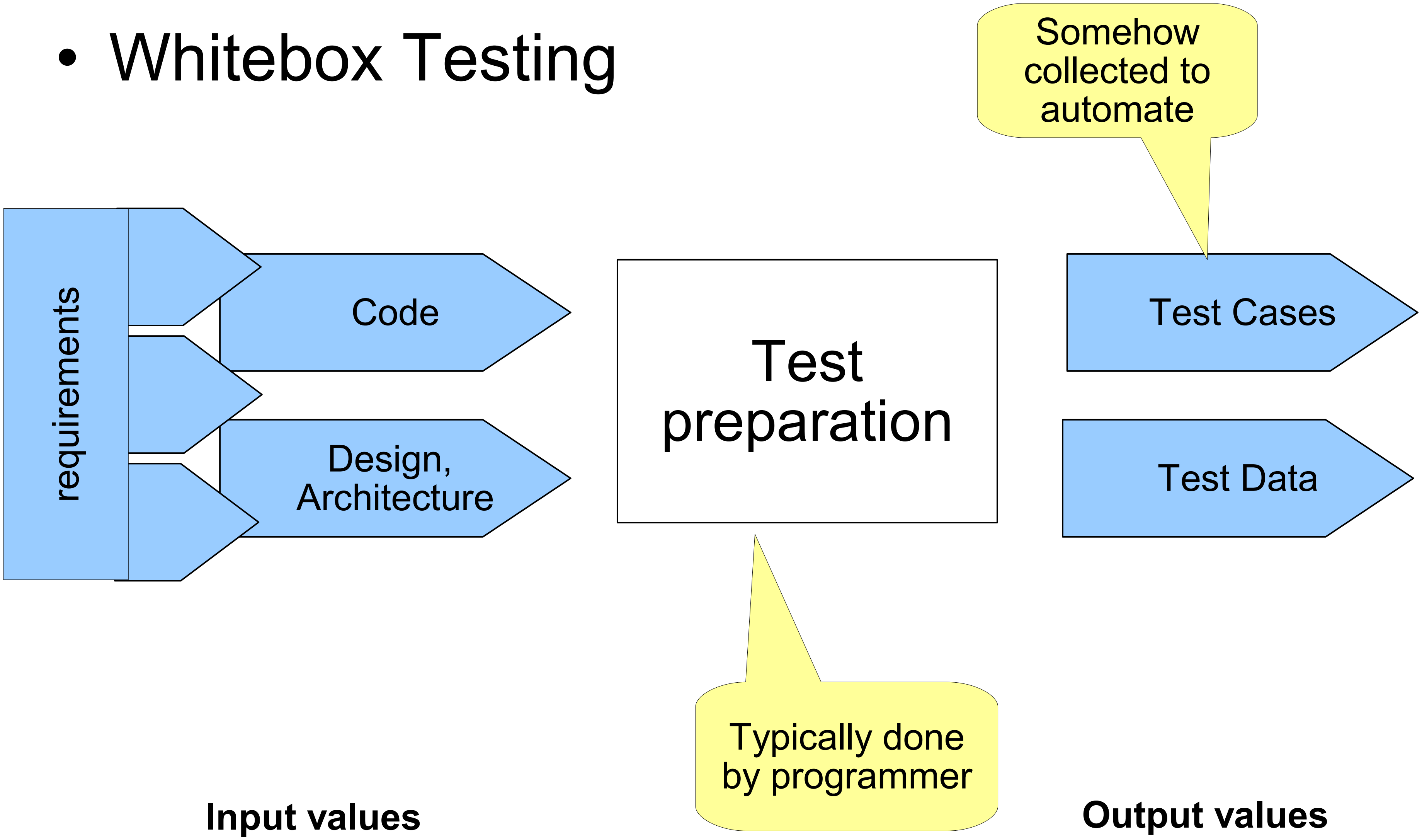
Test Basics
Input and Output values

- Whitebox Testing



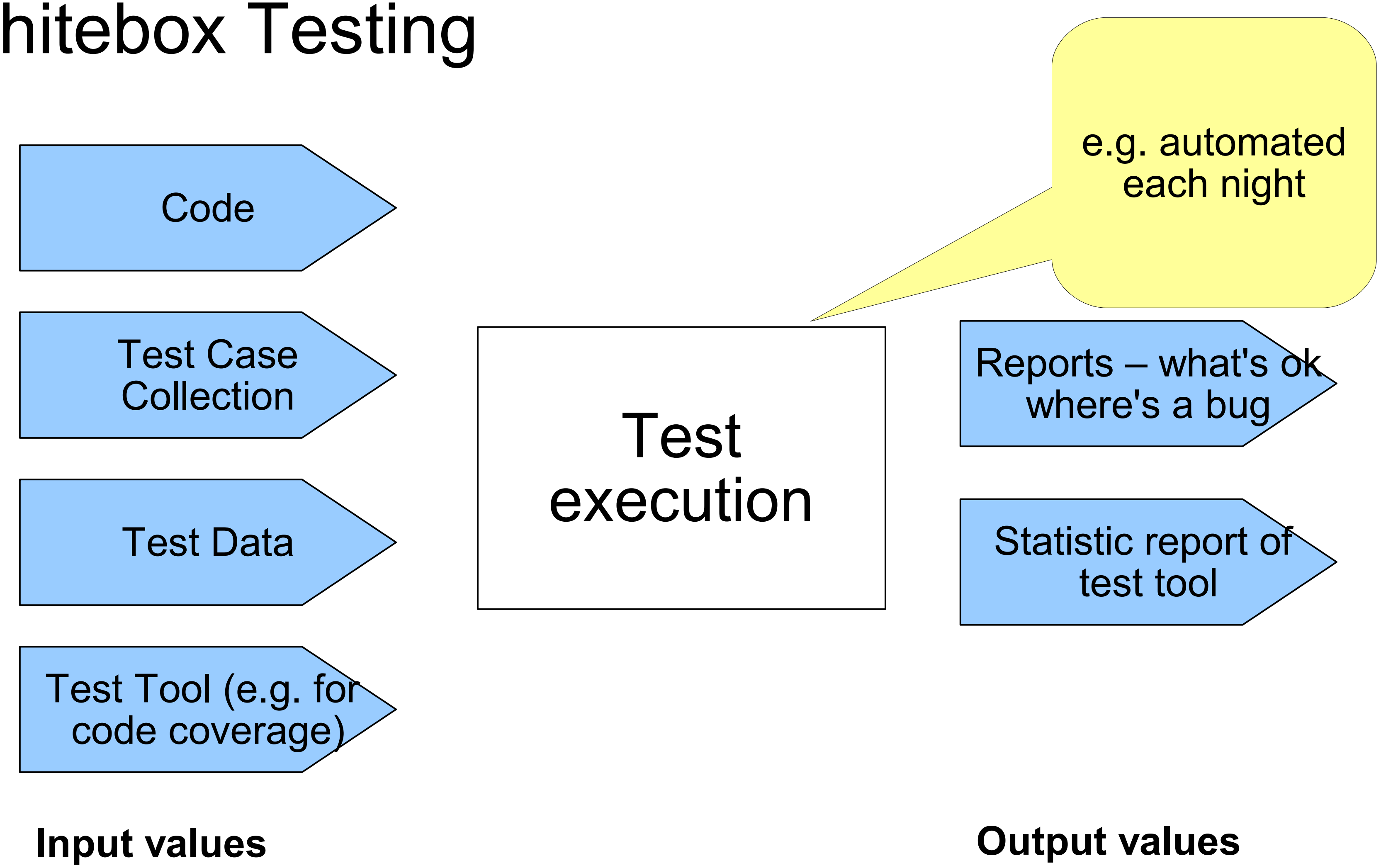
Test Basics
Input and Output values

- Whitebox Testing



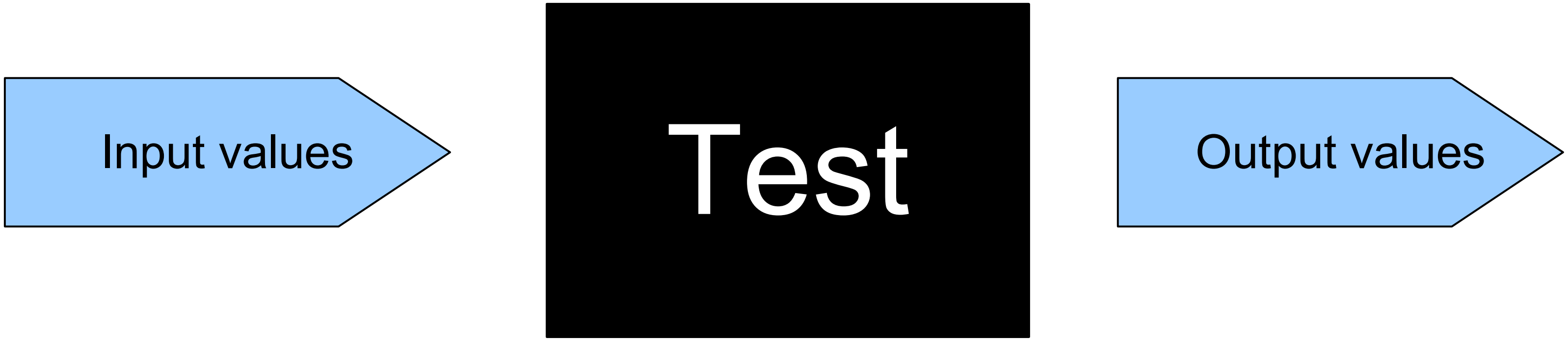
Test Basics
Input and Output values

- Whitebox Testing


e.g. automated each night
Code
Test Case Collection
Test Data
Test Tool (e.g. for code coverage)
Test execution
Reports – what's ok where's a bug
Statistic report of test tool
Input values
Output values

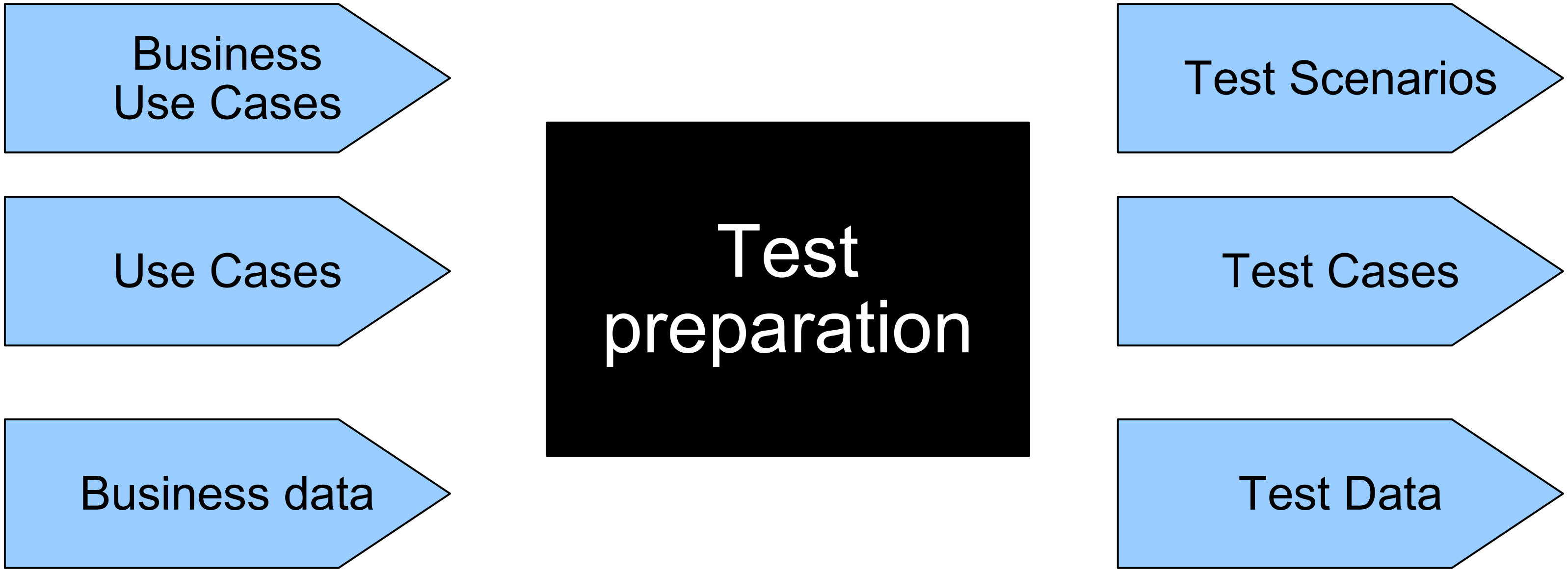
Test Basics
Input and Output values

- Blackbox Testing



Test Basics
Input and Output values

- Blackbox Testing

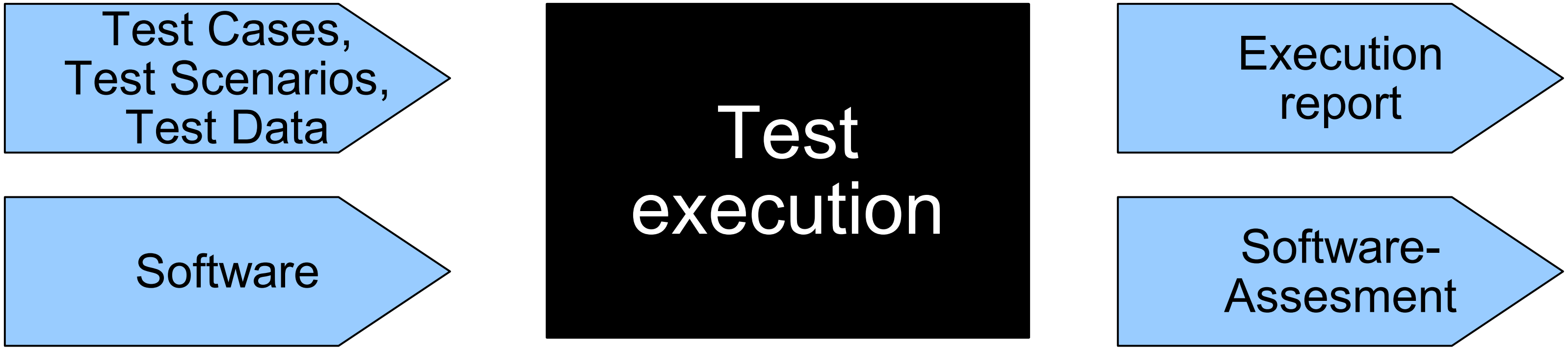


Input values

Output values

Test Basics
Input and Output values

- Blackbox Testing



Input values

Output values

Test Basics
Input and Output values

- Whitebox Testing – What do I need from others?
  - General goals (e.g. code coverage, guide lines)
  - Not something special, somehow requirements
  - Additionally some nice test tools

- Whitebox Testing – What do I create?
  - Test cases and test data for “internal use”
  - Better programs!

Test Basics
Input and Output values

- Blackbox Testing – What do I need from others?
  - Specification
    - Business Scenarios (Business Use Cases)
    - Functional Requirements, Use Cases
    - Non functional Requirements
    - Data model (Business Data Model, Physical Data Model)
    - where applicable GUI-Prototype, if so informal documents contenting requirements
  - Business data, application data (artificial, original)
  - Software

Test Basics
Input and Output values

- Blackbox Testing – What do I create?
  - Test Suite
    Test Cases, Test Scenarios, and Test Data as basics for the test execution
  - Reports
    The results documented in reports – based on Test Cases, Test Scenarios, and Test Data – are basis for a decision if the software could be accepted or not.

Test Basics
Tools

- As a basic principle:
  Tools should help and be adequate
  - No / cheap tools could be expensive!
  - A fool with a tool is still a fool!

Test Basics
Tools

- How to differentiate between tools?
  - Commercial / Open Source
  - Programming level (Unit-Test-Tools) / Requirements level
  - Test Management Tools
    - Test Case Creation and Update
    - Test Case Execution
    - Reporting
  - Test Automation Tools
    - GUI Regression Tests
  - Load Test Tools
    - to verify performance

Test Basics
Tools

- Which tool to use? Carey Schwaber, Forrester, May 31, 2006 * wrote:

  “Forrester evaluated leading functional testing solutions — tool suites with support for manual testing, test automation, and test management — across 87 criteria.
  Our research revealed **Mercury Interactive** to be the sole Leader in this market,...
  IBM follows Mercury as a Strong Performer, with especially notable manual testing capabilities and the best test automation tool for users with programming skills.
  Borland Software and **Compuware** are both Strong Performers — but just barely. Our evaluation also included **Empirix**, ...”

* Source: http://www.forrester.com/Research/Document/Excerpt/0,7211,37587,00.html

Test Basics
Tools

- Which tool to use?
  Recommended proceeding:
  - Notice requirements
  - Collection of information, play around with tools
  - Evaluation of tools following requirements, recommendation of one tool
  - Decision
- Inquiries (Sources in German)
  - Comparison of Test Management tools [IPR+06]
  - Comparison of load test tools [Wei05]

Test Basics
Tools

- Unit-Test-Tools [Tre07]
  - Junit
  - nunit
  - jwebunit
  - Htmlunit
  - TestNG
- Development tools [Tre07]
  - Visual Studio Team test
  - perl
  - ruby

Test Basics
Tools

- HP Mercury [HPM08]
  - TestDirector, Quality Center (Test Management Tool)
  - QTP (Quick Test Professional), Winrunner (Automation Tool)
  - Loadrunner (Load test tool)
- IBM Rational [Rat07]
  - Rational Test Manager (Test Management Tool)
  - Rational Robot (Automation Tool)
  - Rational Performance Tester (Load test tool)

Test Basics
Tools

- Borland [Bor07]
  - SilkCentral Test Manager (Test Management Tool)
  - Silk Test (Automation Tool)
  - Silk Performer (Load test tool)
- Compuware [Com07]
  - QA Director (Test Management Tool)
  - QARun, TestPartner (Automation Tool)
  - QALoad (Load test tool)

Test Basics
Tools

- TEQneers [TEQ07]
  - TEQdit (Test Management Tool)
- SQS [SQS0]
  - SQS-TEST/Professional

Test Basics
Tools

- Open Source Products [OST07]
  - IeUnit (Unit Test Tool)
    ...a simple framework to test logical behaviours of web pages
  - Bugzilla Test Runner (Test Management Tool)
  - Fitnesse (Test Management Tool)
  - rth (Test Management Tool)
  - Selenium (Automation Tool for Web applications)
  - WATIR - Web Application Testing in Ruby (Automation Tool for Web applications)
  - Software Testing Automation Framework (STAF)

Test Basics
Tools

- Open Source Products [OST07]
  - Eclipse Test & Performance Tools Platform Project ... for Java applications.
  - Apache JMeter
    is a Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

Test Basics
Tools

- Open Source Products [OST07]
  - Abbot
    is a framework for testing Java GUIs. It lets you launch an application or GUI component, play back user actions on it, and examine its state. Tests may be coded or scripted where test scripts are JUnit extensions
  - Siege
    Load Test Tool for web applications. Siege is an http regression testing and benchmarking utility. It was written on GNU/Linux and does not run under Microsoft Windows.