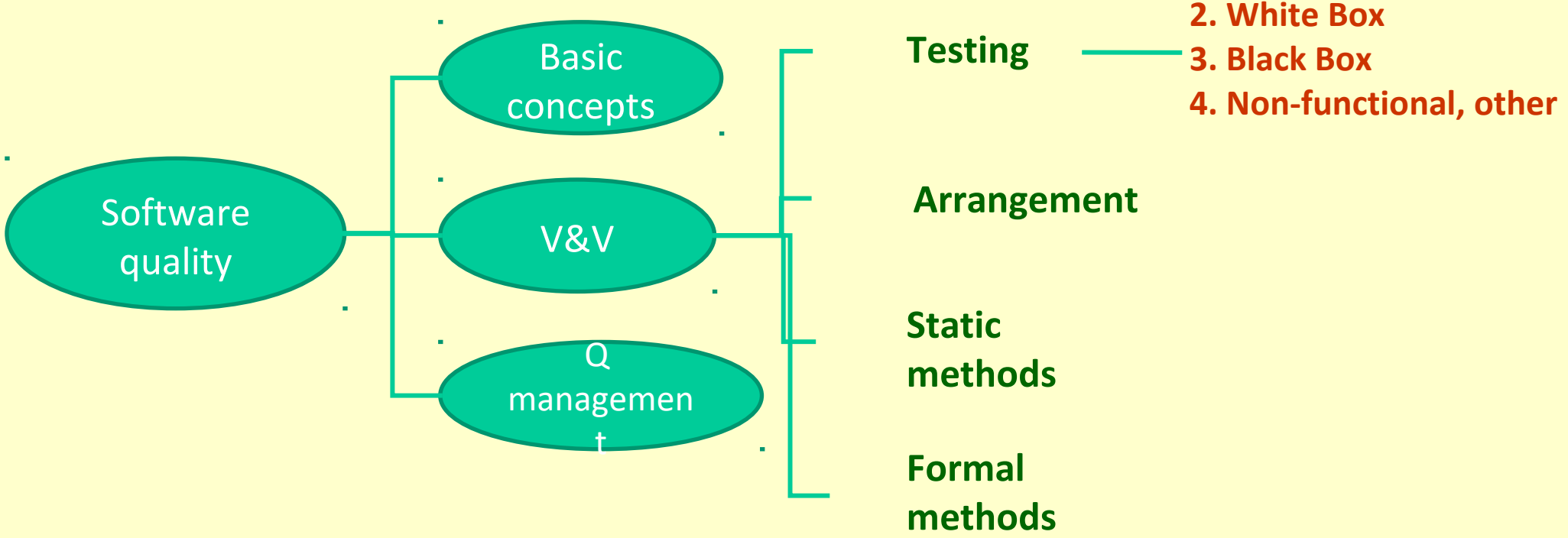
Software processes, quality, and standards
VTV, fast methods, automation

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
Version 4.10.2017

Context and content



Key points to know

- Testing. Test. Verification. Validation. Certification. Debugging. Error, fault, failure.
- Goals of testing? Successful test? Testing in main SDLCs.
- Ad hoc, smoke, experience based, exploratory testing
- Risk based testing. Risk parameters. Finding risks. 4T for risk control.
- Price effectiveness of testing methods
- How much to test?
- Difference between systematic and non-systematic testing
- Test automation - idea, possibilities, tools, when to use, recommendations, advantages and disadvantages.

Verification, Testing, Validation

- Testing?
- Test?
- Good test?
- Testing questions
- Error, fault, failure
- V&V, certification
- Testing in software life cycle

Testing

Testing - many definitions, both broad and narrow, examples:

Software testing - the dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite executions domain, against the expected behavior. ISO/IEC TR 19759:2005, Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK).

The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects (International Software Testing Qualifications Board, ISTQB, http://www.istqb.org) .

Test

- Test and associated notions - also many definitions, example:
  - A set of one or more test cases (IEEE Std 829, ISTQB).
  - Test case: A set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement. [ISTQB, After IEEE 610]

Success in testing

- System or software is not enough for efficient testing – compare first lesson(s) and the project
- If errors were found, then the test was successful?
- A good test has a high probability of detecting errors?
- Can testing prove that there are no errors in the program?

Testing questions

- Developers, users, testers separately
- In designing tests, the following problems have to be solved:
  - How to choose inputs?
  - How to value outputs?
  - When to finish testing?
  - Who are the testers?
  - When, whether and how to perform regression testing?
- Risk-based, experience based, exploratory...
- Functional /specification and program / structure based (“black and white box”) testing

Error, fault, and others (ISO/IEC 2382 / ISTQB)

- Error - A discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition / A human action that produces an incorrect result
- Fault - An abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function / =Defect: A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system
- Failure - The termination of the ability of a functional unit to perform a required function / Deviation of the component or system from its expected delivery, service or result
- To debug - To detect, locate, and eliminate errors in programs /debugging: The process of finding, analyzing and removing the causes of failures in software

Verification, validation, certification

- Verification - activity trying to demonstrate that the result of the next stage of development complies with the specification provided in the previous stage
- Validation - activity trying to demonstrate that what was required has been achieved
- Certification - activity of the third part trying to demonstrate that a product or service complies with relevant legislation, standards and norms

Activity?

- Executing software with a series of inputs and comparing outcomes
- Checking whether requirements conform to stakeholder needs
- Documentation review
- Expected output: 0. Actual: 1
- Must be: "T:=0" instead of "T:=1"
- The Ambulance Dispatch System cannot find the nearest ambulances
- Finding out why the Ambulance Dispatch System cannot find the nearest ambulances

...can testing of a system result in its failure?

V & V & testing in these software life cycle models?

- Waterfall - http://www.nouveautech.co.uk/images/developmentprocess.gif
- V-model - http://en.wikipedia.org/wiki/V-Model_(software_development)
- Waterfall – iterative - http://kanemar.files.wordpress.com/2005/12/Staggered-Iterative-Waterfall.jpg
- Spiral - http://en.wikipedia.org/wiki/File:Software_Development_Spiral.svg
- Extreme Programming - http://www.extremeprogramming.org/map/project.html
- Scrum - http://www.scrum.org/Scrum-Guides
- TDD - http://en.wikipedia.org/wiki/Test-driven_development
- Also: http://agilemanifesto.org/
- Also: http://en.wikipedia.org/wiki/Software_development_methodology
- Also: http://www.youtube.com/watch?v=OfgfnZZdMlI

Fast methods

- Ad hoc
- Smoke
- Experience
- Exploratory
- Risk based
- Effectiveness?
- How much to test?

Ad hoc/smoke/expert/exploratory/risk based

- Fast to learn, cost-effective
- Often not very systematic and / or deep
- Main or additional components in some development frameworks
- Before or in addition to more systematic methods
- Insufficient time / resources for testing

Ad hoc and smoke testing

Ad hoc testing: Testing carried out informally; no formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity (ISTQB)

Smoke testing: Testing the main functionality of software to check the most important features

...Useful? Sufficient? When? When not?

Experience-based test design technique

Test cases are derived / selected based on the tester's experience, knowledge and intuition (ISTQB)

- General knowledge
- Knowledge about the application domain
- Knowledge about typical defects (defect based techniques)
- Knowledge about development, developer, customer
- Etc

Example: Fine for delay

- Given:
  - deadline (from 04.10.2000 incl.)
  - payment (from 04.10.2000 incl.)
  - amount (S, up to 100M incl.)
  - Fine % per day P, 0<=P<=100
- Find: fine
- To calculate the number of days, a tested function “Days from 01.01.1990”, D (D,M,Y) is used. D (1,1,1990) = 0

...expert knowledge?

Exploratory testing

- An informal test design technique
- The tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests.

- (www.istqb.org).

Risk based testing (simplest form)

From the inital stages:

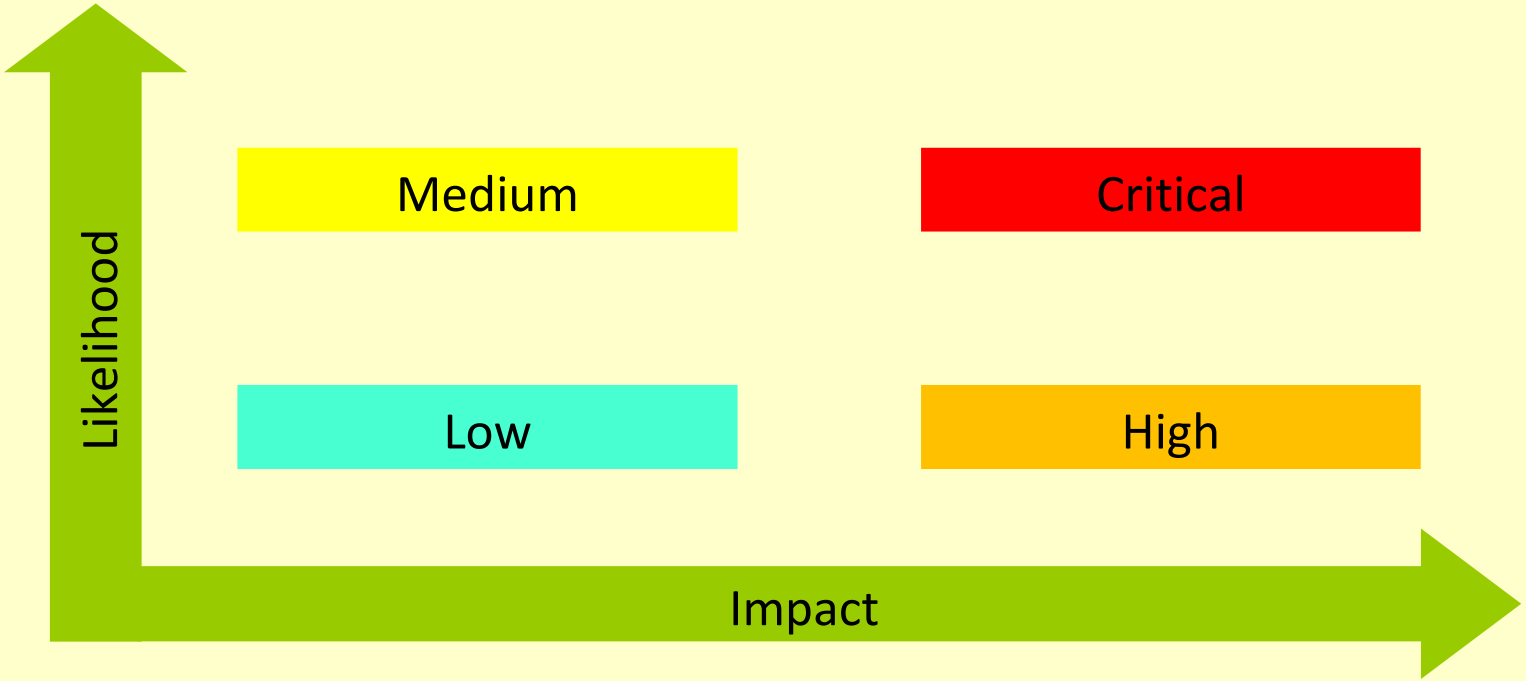
- Identify risks / Assign priorities
- Test the most critical risks
- Inform the stakeholders

Finding risks

- Most valuable functionality (where is the risk?)
- Non-functional requirements
- Safety (how related to functionality? Where is the risk?)
- Technical: complexity, novelty, data volume, etc
- Development: deadlines, resources, team, etc

Not always possible to test

Risk (a possible matrix)



Risk based testing => 4T for risk control

- Treat
- Transfer
- Tolerate
  - Accept
  - Manage impact (contingency plans etc)
- Terminate

Summary about the method

- concept
- preconditions
- advantages
- disadvantages
- results
- relationship to other methods
- evaluation
- tools

Price effectiveness*

- Developer testing
- Smoke
- Risk based
- Walkthrough etc
- Testing with user data
- Exploratory
- Experience
- Automation
- Boundary situations
- Equivalence partition
- Program based testing (branch adequacy)
- Random
- Seeded errors
- Proofs

* Subject to specific situation

How much to test?

- Ideally
  - requirements => testing volume
- Practically (examples)
  - it is difficult to evaluate requirements and the extent to which they are satisfied
  - work must be done fast
  - experience determines testing volume
  - application is not critical

Possibilities to determine testing capacity

- user / risk / experience / etc
- essential data combinations must be tested,
- all equivalence classes (boundary situations) must be tested
- testing must satisfy the branch coverage criteria
- data based testing boundary situations must be tested
- software reliability must be P%

There can be still be errors, but

- Testing methods
  - improve testing efficiency
  - enable test design
  - enable systematic testing
  - enable outside evaluation

Testing standards

- Voluntary
- A useful list of possible non-mandatory activities
- Depends on organisation etc
- Sometimes more for the procurer

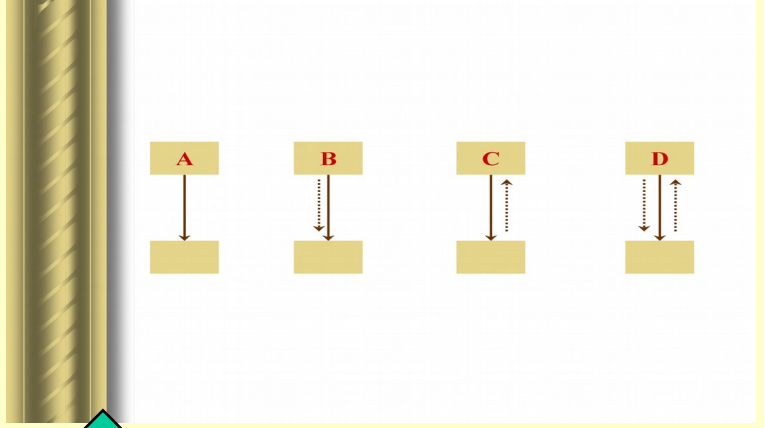
"ANSI/IEE Std 829 Standard for Software Test Documentation"

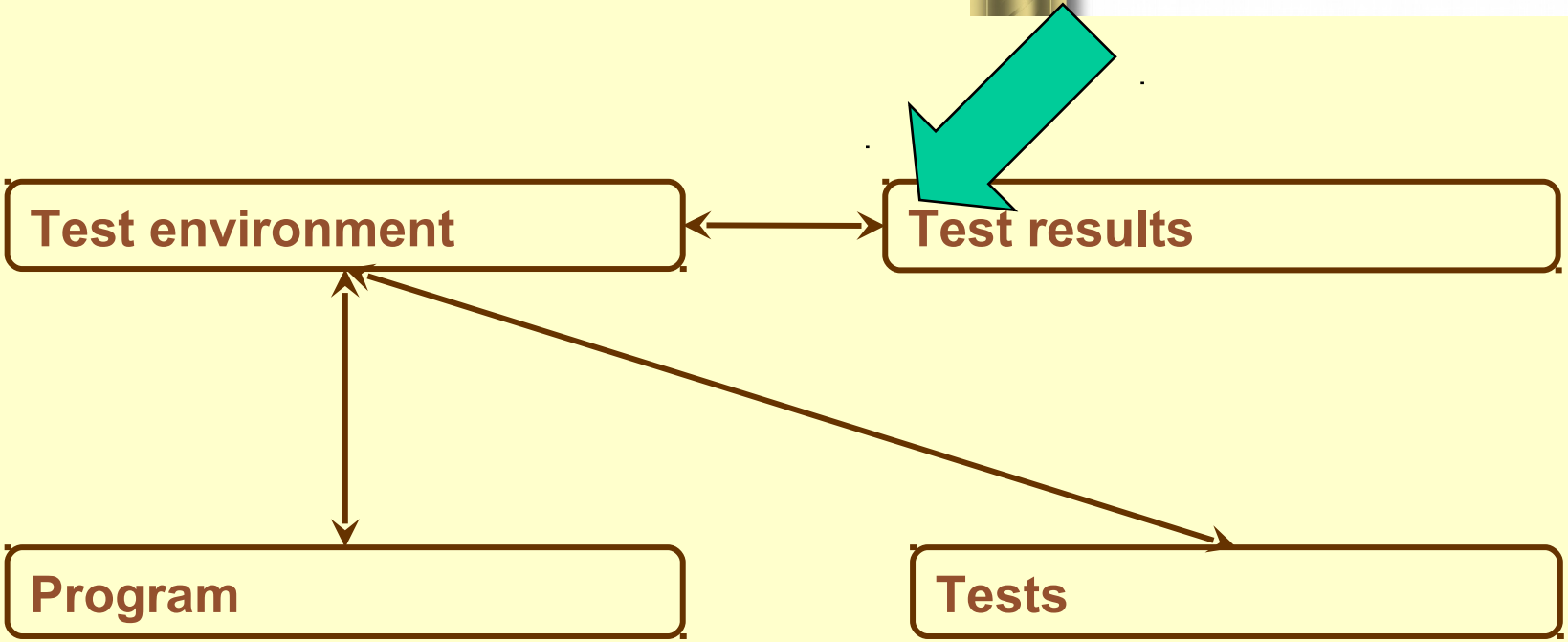
1. * TEST PLAN
2. * TEST-DESIGN SPECIFICATION
3. * TEST-CASE SPECIFICATION
4. TEST-PROCEDURE SPECIFICATION
5. * TEST-ITEM TRANSMITTAL REPORT
6. TEST LOG
7. * TEST-INCIDENT REPORT
8. * TEST-SUMMARY REPORT

Test automation

- What for?
- Possibilities and examples
- The cost of automation
- Recommendations
- Advantages and disadvantages

A naive start





Why?

- Improve quality (both actual and perceived!)
- Faster integration / feedback (depends)
- Agile methods impossible without automation
- Simplify routine work
- Regression testing
- Seems simple

Automation (1)

- Saving and executing tests (eg Selenium, JUnit)
- Tests scripts - generating test scripts automatically / manually
- GUI test drivers
- Automation of test generation – automatically generate statement or branch coverage tests, tests based on functional description, discover branches not covered etc

Automation (2)

- load testing tools
- tools for simplifying the testing process: for example software that simplifies the generation of drivers during integration testing, random fault generators and so on
- Performance analysis
- Coverage analysis

Automation (3)

- tools for estimating the testing quality, for example to find program components not covered during testing
- static analysis tools, for example software for evaluating program metrics
- bug management tools
- testing process and resources management tools
- website testing tools

Website testing tools (examples)

W3C Markup Validation Service
W3C Link Checker
CSS Validation Service
WDG HTML Validator
A Real Validator
CynthiaSays
Canoo WebTest
Rational Robot jt
Parasoft WebKing
Selenium
Appache Jmeter
...

Automation costs

Cost-effective and simple?

Resources needed:

- Tools
- Training
- Development procedures
- Creating automated tests
- Test execution
- Updating tests
- Porting to new environments
- Other resources

Recommendations (1)

- Understand the difference between the test process and the automation process (tools). In case of automation testing must still remain transparent
- Automated testing is a supplement to the whole testing process – not as a replacement
- Choose the tools carefully, get acquainted with them first, collect the information
- Choose carefully the structure of test sets, let it be understandable and supportive to the testing process

Recommendations (2)

- Each automated testing session must deliver a report describing tests performed and errors found
- Ensure that the product is sufficiently developed – then there is hope that benefits from the test automation exceed the expenses of test transformation
- Automation pays off if the object(s) of testing are (1) large enough, (2) sufficiently variable (regression testing is needed), (3) sufficiently stable (tests are not redesigned too often)

Advantages and disadvantages of testing tools

- The testing tools enable to get free from a great deal of routine handwork
- Many agile development methods are possible only with testing tools (why?)
- The testing tools require a lot of extra work, especially in case of fast changing requirements, software, and environment

Key points to know

- Testing. Test. Verification. Validation. Certification. Debugging. Error, fault, failure.
- Goals of testing? Successful test? Testing in main SDLCs.
- Ad hoc, smoke, experience based, exploratory testing
- Risk based testing. Risk parameters. Finding risks. 4T for risk control.
- Price effectiveness of testing methods
- How much to test?
- Difference between systematic and non-systematic testing
- Test automation - idea, possibilities, tools, when to use, recommendations, advantages and disadvantages.

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.

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

Christian Johansen, Test-Driven JavaScript Development. Addison-Wesley, 2010,

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 2.

Moodle: „Software Quality (Tarkvara kvaliteet)”. Alternate download: tepandi.ee