

Software quality and standards Testing / V&V arrangement

Jaak Tepandi

Tallinn University of Technology

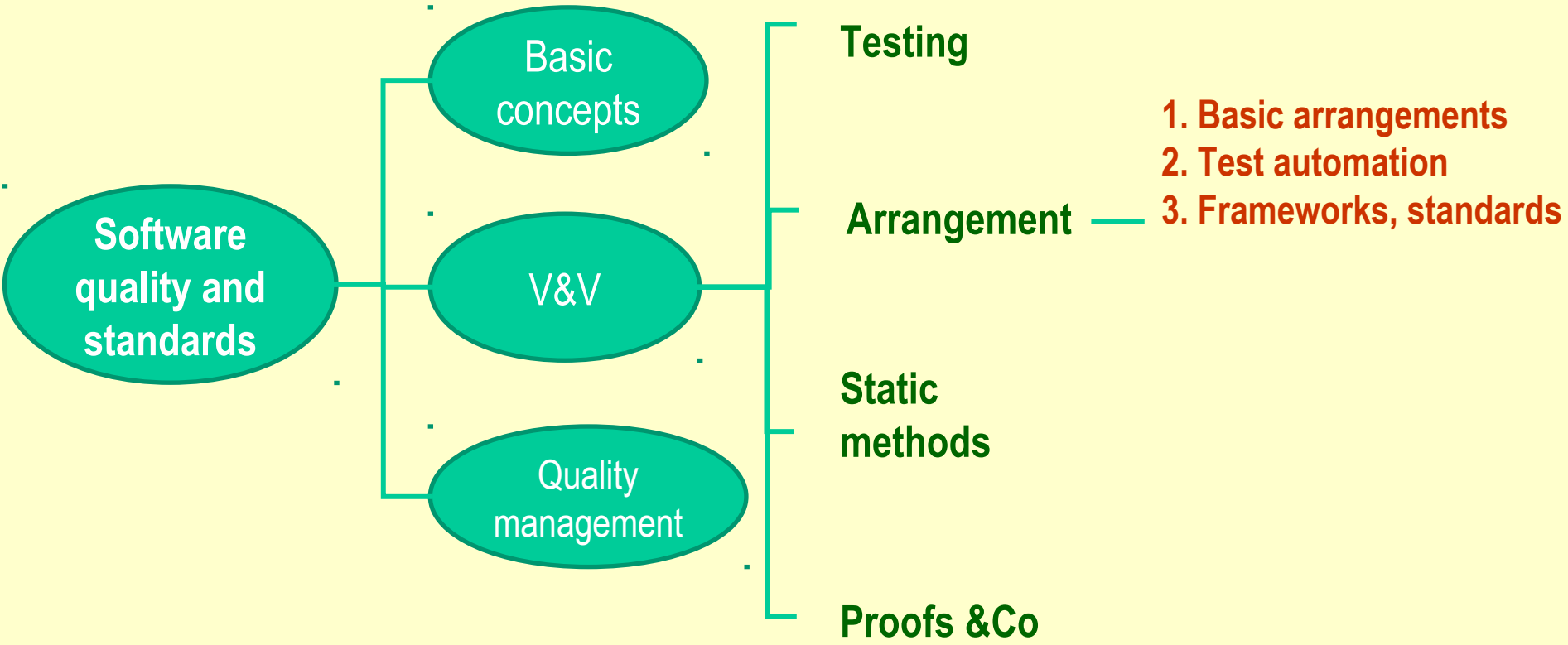
Institute of Informatics

Moodle: „Software Quality (Tarkvara kvaliteet)”

Alternate download: tepandi.ee

Version 19.10.2016

Context and content



Part 1. Basic arrangements

- Simple
- Need for more complicated management
- Module, integration, validation, system testing



Simple arrangement (1)

- **user = developer**
- **develop yourself, test yourself (use yourself)**



Simple arrangement (2)

- few requirements
- low criticality
- customer gives an order
- programmer delivers the required software
- customer tests the software and returns description of errors found
- programmer corrects the errors and delivers the next version



Simple arrangement (3)

- **more significant software**
- **many users**
- **testing by the developer**
- **testing by the application and testing experts (for example from a support group)**
- **testing by the users**
- **possibility to go from each level to previous level if an error was found**



Need for more complicated management

- when software is built into a system, the system test is usually expensive – there is a need for separate testing of software and the system
- the product is complicated - start the control in the beginning of development and divide it into manageable parts
- product with critical reliability – divide into stages, use special methods
- organizationally separated development stages – cooperation of analysts, designers and programmers
- organizationally complicated user - cooperation of different user groups
- the product is used in different ICT, language, legal environments - needs to be tested for these environments
- frequent releases - development and operations need to be integrated
- etc

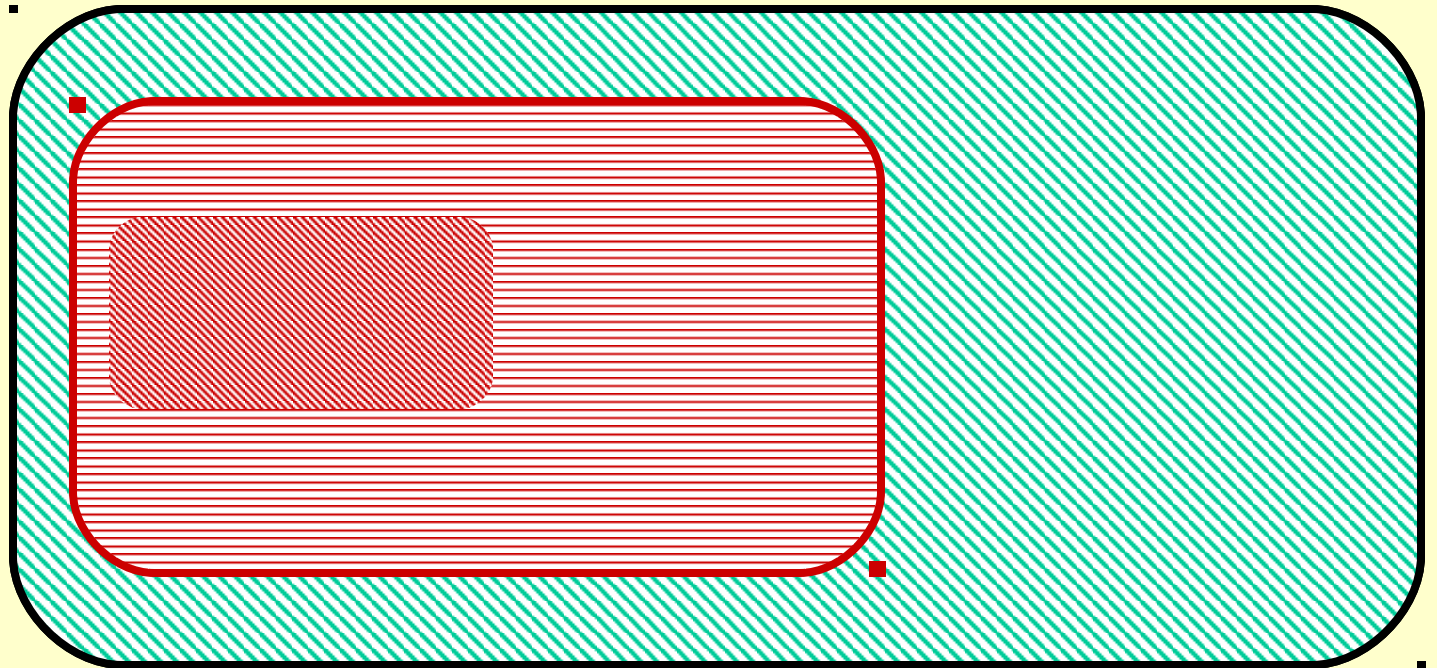


Price effectiveness of some testing approaches

- developer testing
- smoke testing
- risk based testing
- walkthroughs, reviews
- testing with user data
- exploratory testing
- expert based error search
- testing automation
- boundary situations
- equivalence partitioning
- program based testing
- random data
- reliability oriented methods



System and software testing



Testing on software / system levels

- Module: white box, formal verification, tools; consider cost-effectiveness
- Integration testing
- Validation testing
- System testing



When? When no?

Testing levels

Costar

System requirem. analysis

30...

Softw. requirem. analysis

9..100

Software design

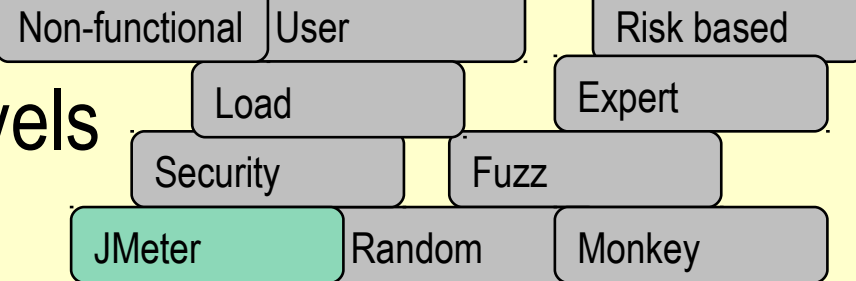
3...10

Softw. programming

Error here = 1

Tests

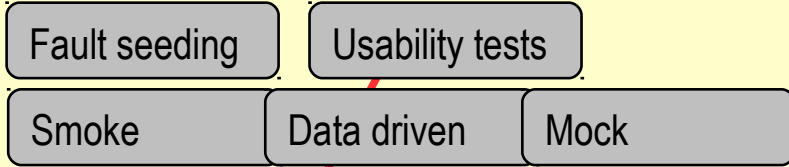
Changes



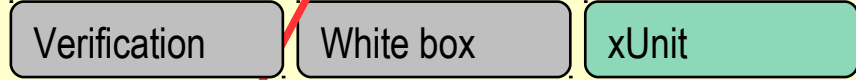
System testing



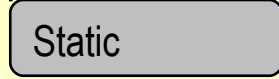
Validation testing



Integr./testing



Module testing

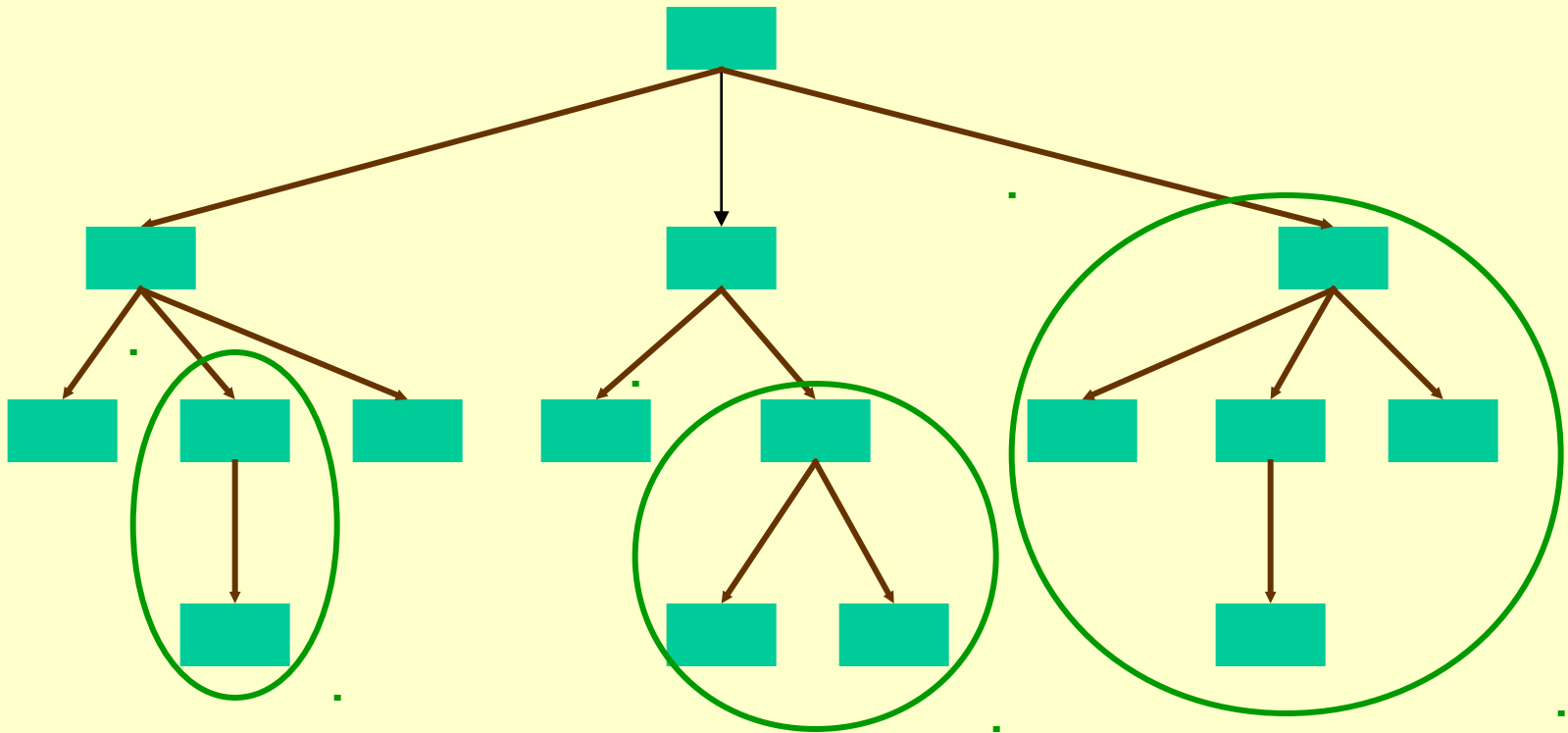


Integration testing

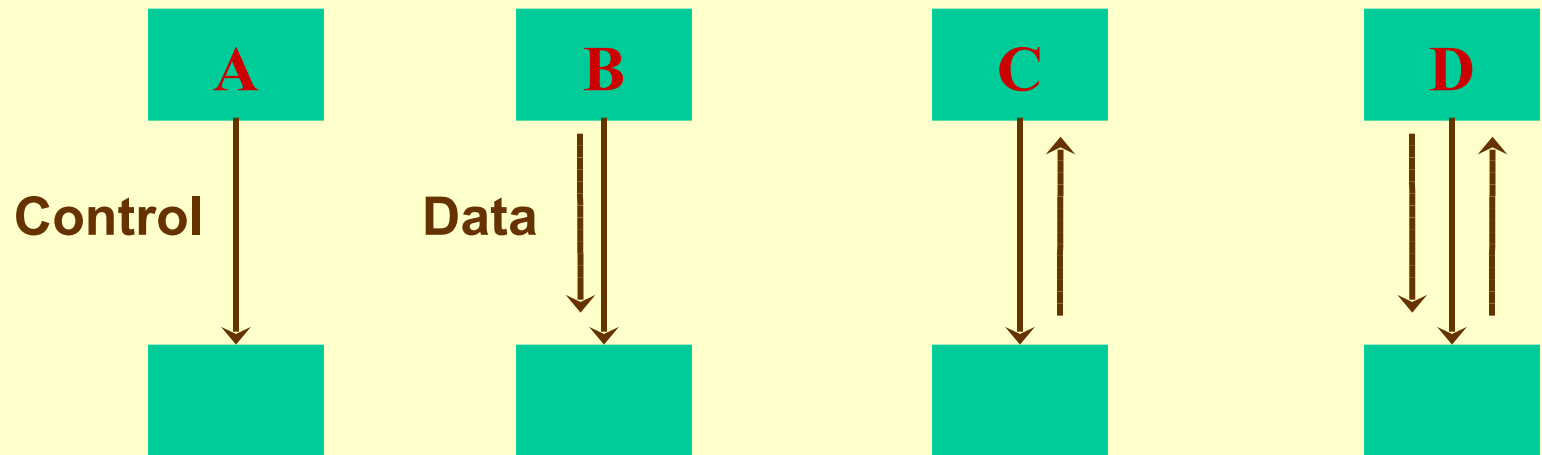
- Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems [ISTQB]
- „Integrate and test” for simpler systems
- Bottom up
- Top-down
- Mixed
- Mock objects



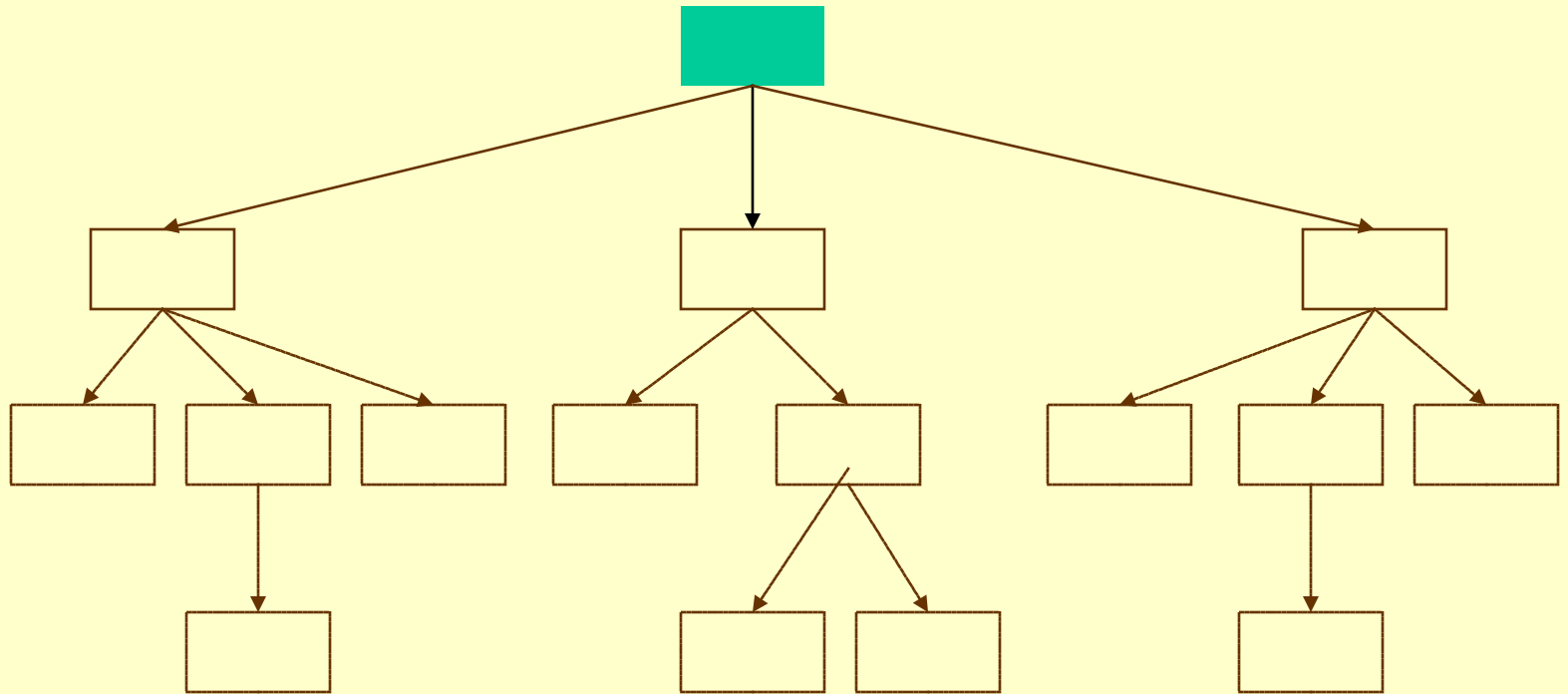
Bottom-up



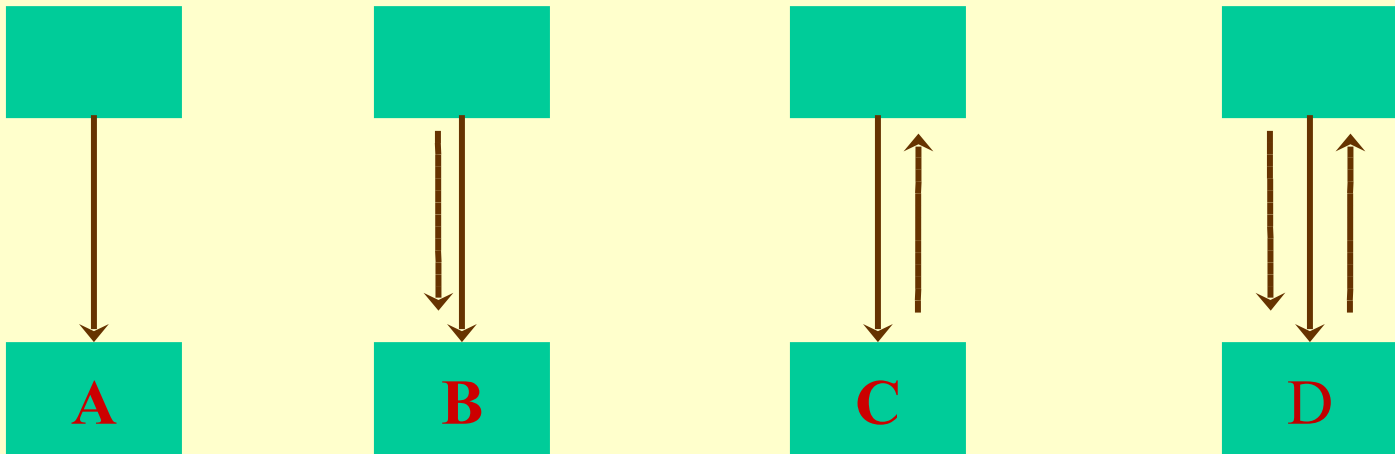
Driver (mock object) examples



Top-down



Shortcut (stub) examples



Validation testing

Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled [ISO 9000, ISTQB]

Activity trying to demonstrate that what was required has been achieved

Did we build the right software?

May involve requirements, software, user/project documentation

May involve Alpha (customer tests, software on developer's site) / Beta (customer tests, software on customer's site) testing



System testing

The process of testing an integrated system to verify that it meets specified requirements [ISTQB].

May involve: black box; usability, performance, load, security, reliability, recovery etc – see requirements



Questions

- **Simple V&V arrangements. The best one?**
- **When there is an need for more complicated arrangements?**
- **Cost-effectiveness of V&V methods**
- **V&V integrated into development**
- **Module, integration, software validation, system testing**
- **What to do when testing is not possible?**

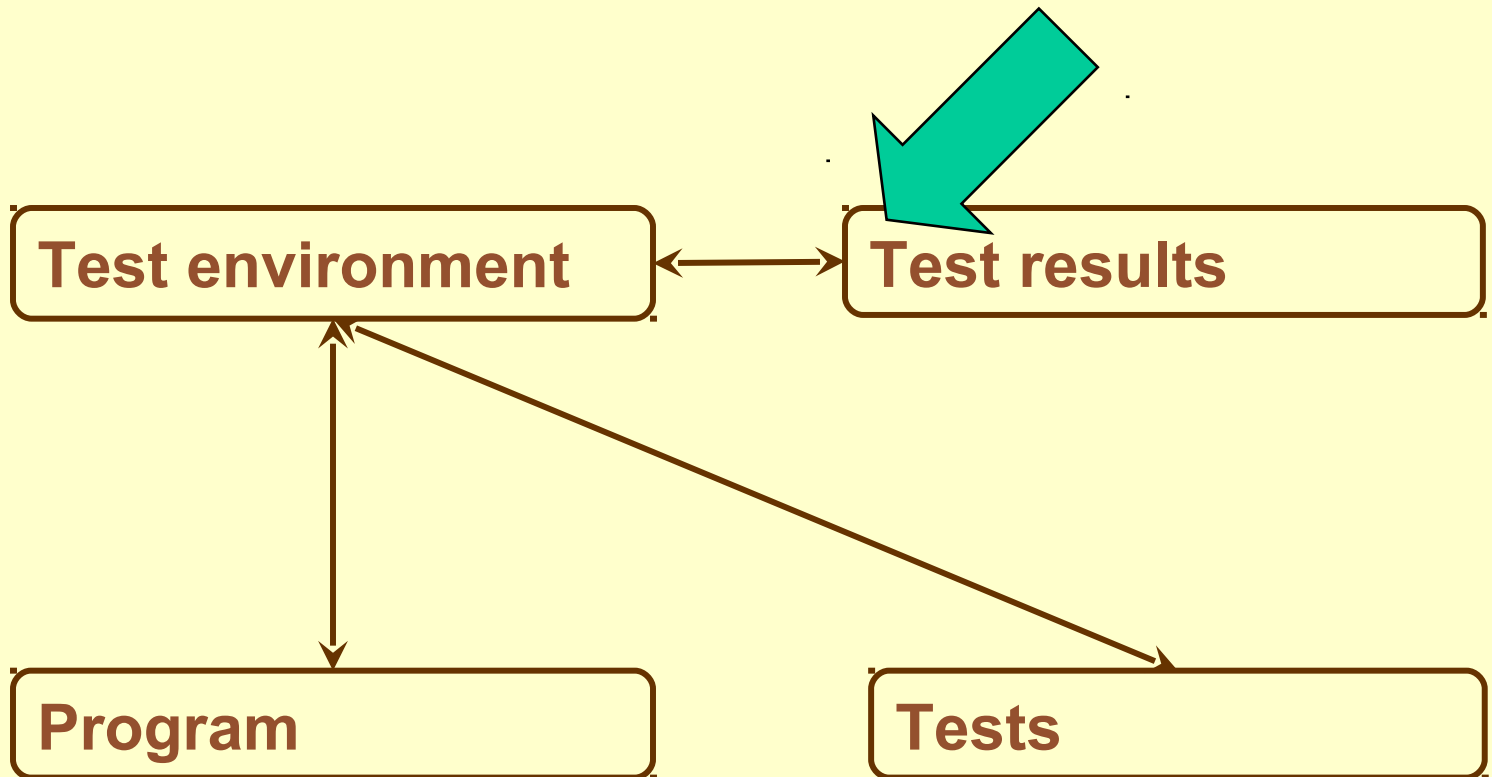
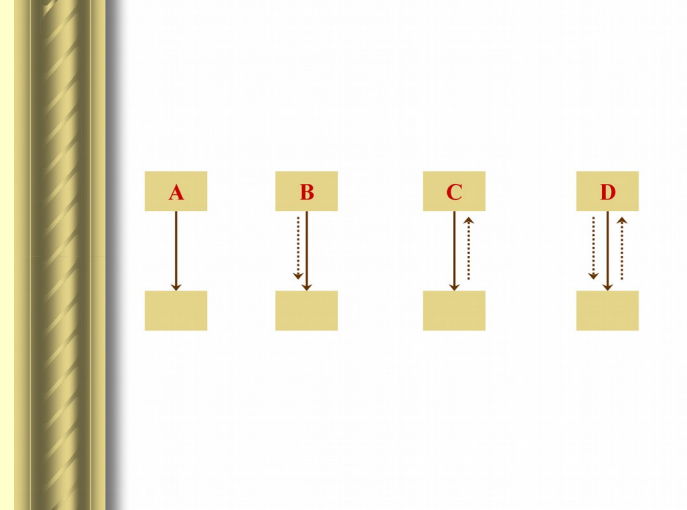


Part 2. Test automation

- **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

Apache 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**



Part 3. Testing frameworks

- RUP
- TPI
- Standards



The RUP Testing Philosophy

- Iterative development + mission
- Low up-front documentation. Detailed Test planning is defined iteration by iteration, based on a governing master Test plan
- Holistic approach - tests from the requirements *and* from other sources.
- Automation. Testing continues throughout, the RUP lifecycle. Tools help generate test data conditions, run tests, and analyze results



Iteration mission examples

- Find as many defects as possible
- Find important problems fast
- Assess perceived quality risks
- Advise about perceived project risks
- Advise about perceived quality
- Certify to a given standard
- Assess conformance to a specification (requirements, design, or product claims)



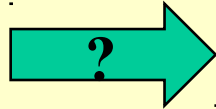
Artifacts

- Test Evaluation Summary
- Test Plan (and sometimes a Master Test Plan)
- Test Ideas, and Test-Idea List
- Test Suites and Test Cases
- Defect and Defect List
- Workload Model



Roles and artifacts

- Test Manager
- Test Analyst
- Test Designer
- Tester



- Test Evaluation Summary
- Test Plan (and sometimes a Master Test Plan)
- Test Ideas, and Test-Idea List
- Test Suites and Test Cases
- Defect and Defect List
- Workload Model



Test Process Improvement, TPI / Tmap NEXT/ TPI NEXT

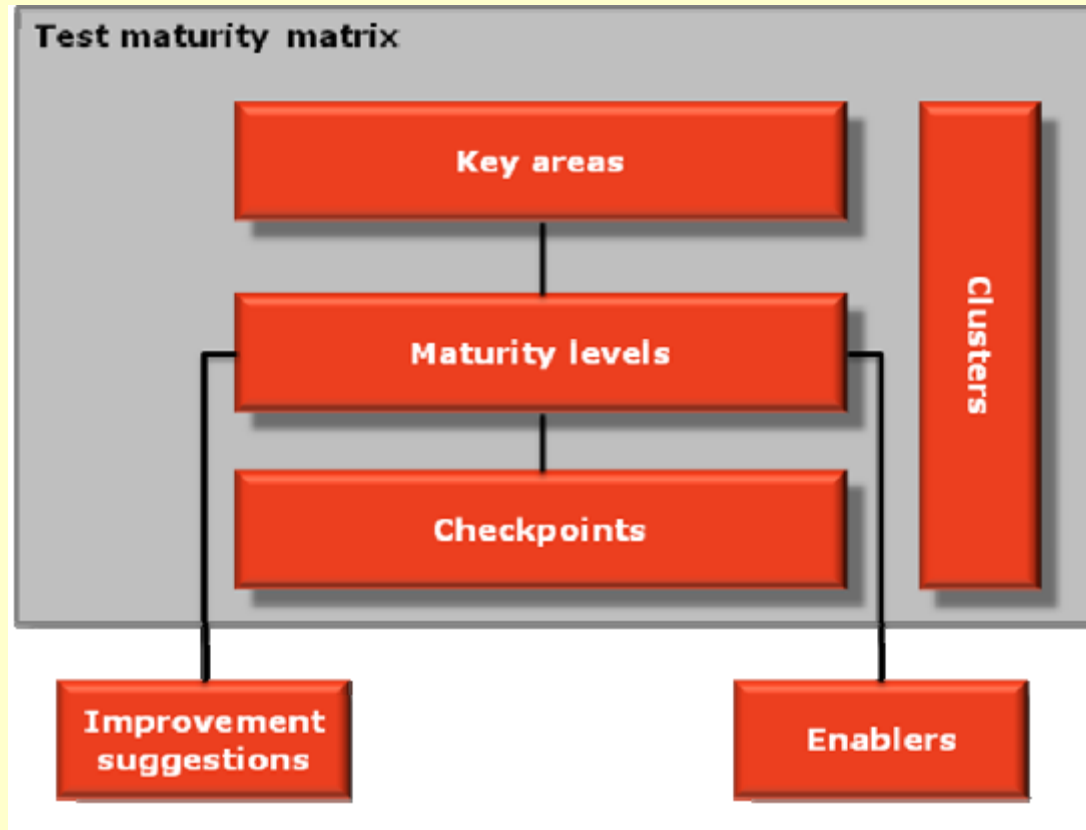
- Model, key areas, methodology
- Test Maturity Matrix
- TPI - Test Process Improvement (the earliest)
- TMAP NEXT - methodology for structured testing of software products
- TPI NEXT - improving test processes

....A sound product risk analysis is the basis for all testing (No Risk No Test)

Seealso: <http://www.tmap.net/tmap-building-blocks>



TPI® Next



16 Key areas

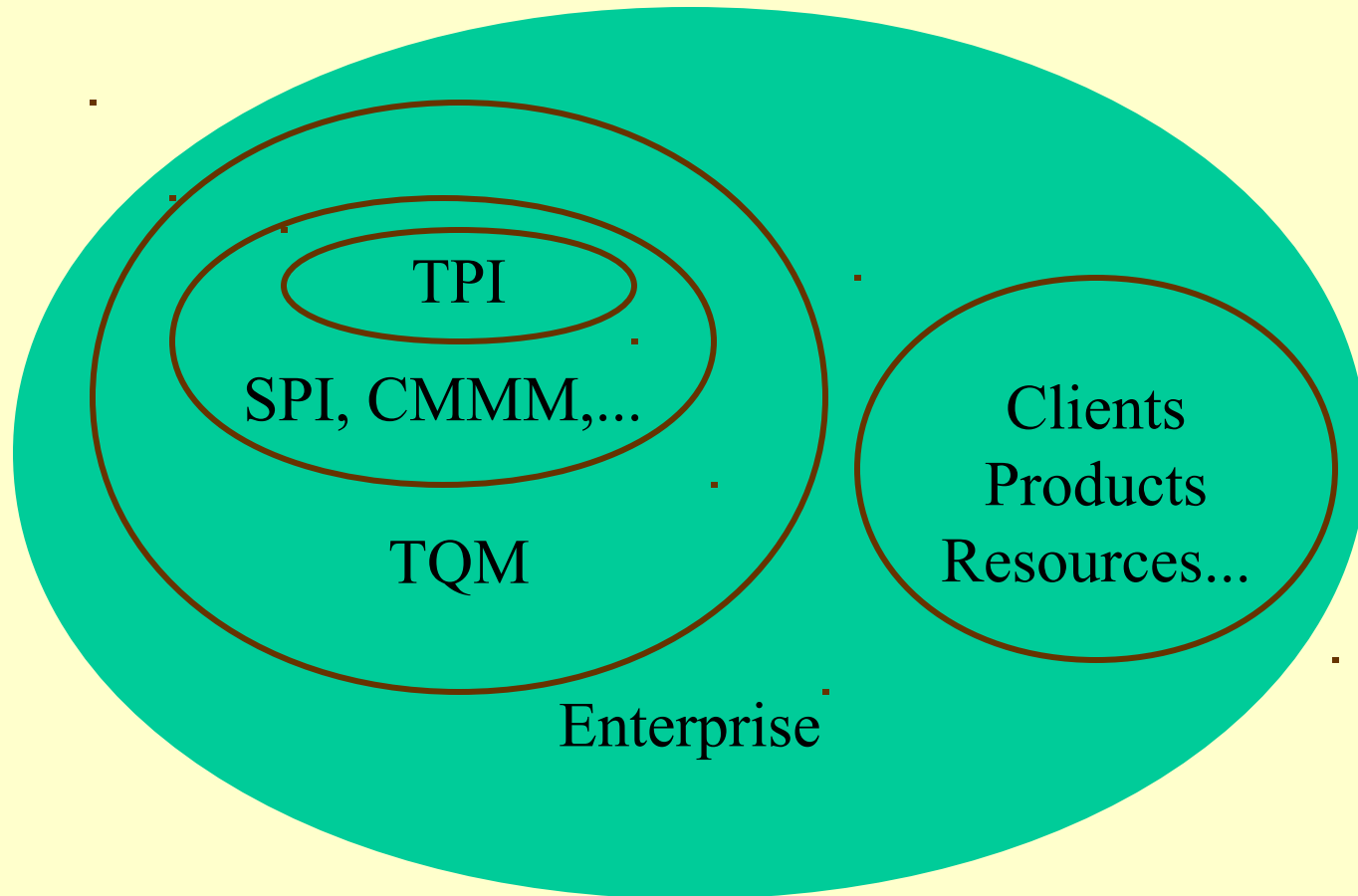
4 Maturity levels: Initial, Controlled, Efficient and Optimizing

Related: CMMI® or SPICE

<http://www.slideshare.net/InfinITnetvaerk/tpi-en-metode-til-forbedring-af-testproces-af-elisabeth-andresen-sogeti>



TPI and others



IEEE Std 1012-2012

1012-2012 - IEEE Standard for System and Software Verification and Validation

The scope of V&V processes encompasses systems, software, and hardware, and it includes their interfaces.

V&V processes include the analysis, evaluation, review, inspection, assessment, and testing of products.

Control has been integrated into life cycle – faults found at early step are more cheaper to repair

Tests at each development level

<https://standards.ieee.org/findstds/standard/1012-2012.html>



Points to remember

Basic testing management schemes

Module, integration, validation, system testing

Testing automation - types, advantages, disadvantages

Testing with XP and TDD (other lessons), RUP, V-model

Test Process Improvement (TPI)



Additional reading - testing / V&V arrangement (examples)

Daniel Galin, Software Quality assurance from theory to implementation, Pearson - Addison-Wesley. Chapter 10.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 10 Sections 2 and 3.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 5 and 6.

Test Process Improvement - <https://www.ict-books.com/topics/testing>.

