5. QualityManagement

5. 1. Introduction

Quality

= all the characteristics of an entity which ensures its ability of fulfilling imposed and implicit requirements [PMBOK]

= all the characteristics which allow fulfilling expected implicit functional and performance requirements + fulfilling standards for correct development /building [Software]

= the degree of excellence for something [Webster Dictionary]

≠

Grade

= rank associated to an entity with a specific functional role, yet different technical characteristics

e.g: photo paper, writing paper, Xerox paper

Difficulties:

- PM should ensure the quality of management and the quality of deliverables
- Quality level can be evaluated by means of criteria unanimously accepted

The quality of project management results from

The quality of results, the delivering of expected results

The satisfaction of the team

The organization experience achieved throughout the project

>> Key of success: communication, cooperation with all stakeholders

The quality of software

There is no solid theoretical background → subjectivism: important factors outlined without complete definitions; clearer definitions for low level quality factors

Mc Call scheme:

Quality factor (high level)

1. product operation
   - correctness (the result is correct?)
   - fiability (the result is always good?)
   - efficiency (the hardware is exploited as well as possible?)
   - usability (easy usage? user needs served?)
   - integrity (guaranteed data/functionalities?)
2. product revision
   - maintenance (easy to repair?)
   - testability (easy to test?)
   - flexibility (easy to modify?)
3. product transitions
   - portability (can be used on different architectures?)
   - re-usage (can be re-use?)
   - interoperability (can be interfaced with other systems?)

Criteria (low level): audit access, control access, accuracy, communicability, consistency, concision, expandability, completeness, tolerance to errors, efficiency in execution, independency to hardware, modularity, simplicity, traceability, complete documentation, etc

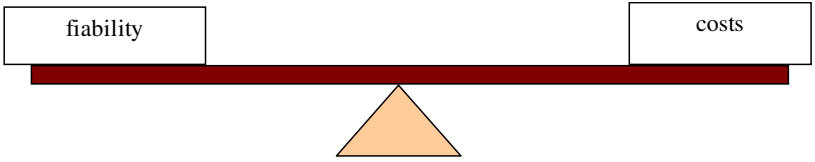
The quality & the grade of a product are related to

- The value of the product + costs requested for product elaboration
- Usability
- Fiability
- Longevity (in IT: longevity is reduced!!!)

The quality & the grade of a service are related to

- Implementation costs
- Value obtained by means of service (unitary price * no. of units)
- User satisfaction
- Fiability
- Longevity

- set a convenient trade-off between conflicting requirements



Recommendations for an efficient quality management:

- **client** must be satisfied by your product/service

  >> the project should deliver what expected

  >> pay attention to usability

  !!! low quality is dangerous, but low grade is sometimes called

  !!! pay attention to implicit requirements

- **proactive actions** are better than inspection + correction
- use **development cycles** that allow “conceptual” model and early testing
- **organizational context** is important: culture + experience + infrastructure

- **main responsibility assigned to PM**, the team only participates!!

  PM is responsible with assuring the necessary quality level + the desired grade for **deliverables** and **project**

  >> sometimes these objectives are conflicting:

  - e.g.: in order to meet the deadlines, the testing phase must be shortened; in order to meet additional client requirements, the team must work more than planned

  >> be a good example!!!

- **high quality is the key of long term success**

Quality standards

- the client gains the guarantee that the product is obtained at an acceptable quality level

- employing a quality standard could be hard and expensive

  E.g.: ISO 9000 (ISO 9000-3 for software) better documenting, availability of clear working procedures, continuous use of available working procedures

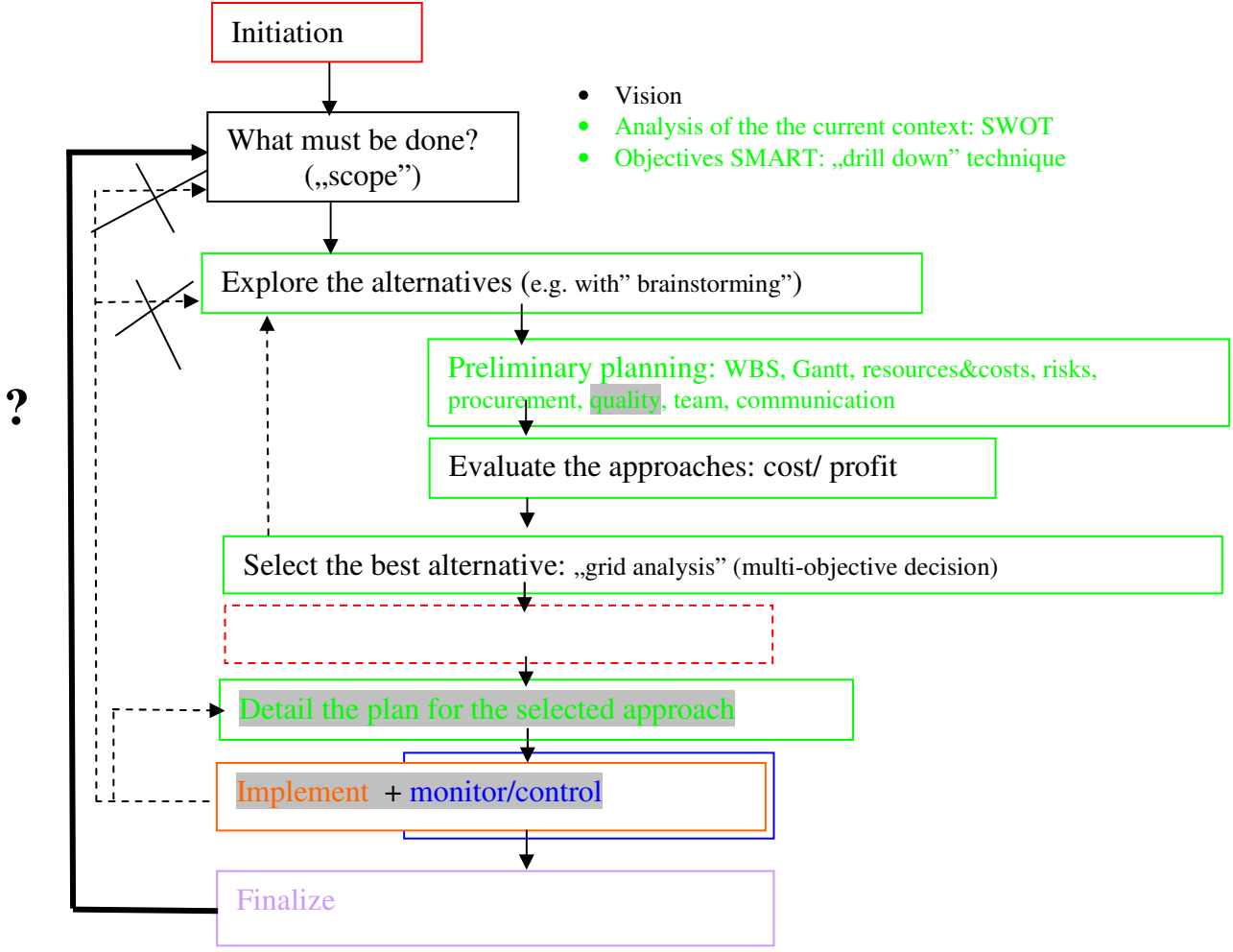
  >> it does not directly ensure the desired quality level

  << periodic re-validation (every 3 years), periodic auditing (every 6 months)

5. 2. Quality Management Processes

= the processes which ensure that the project delivers the desired quality level

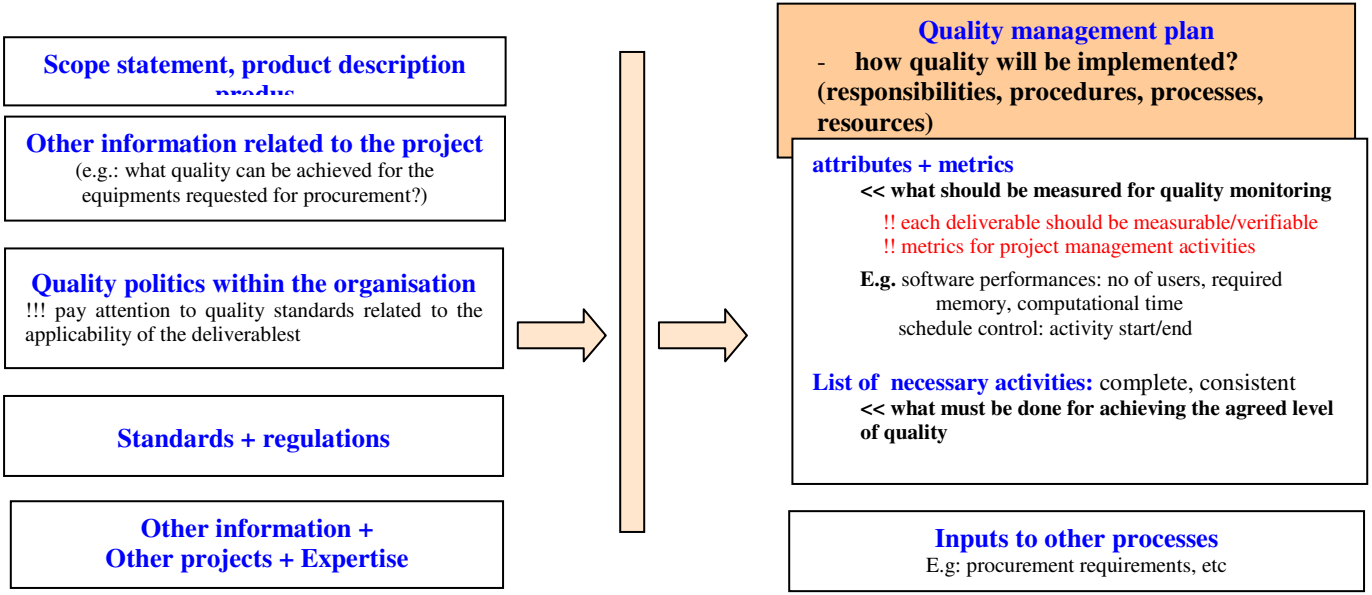
quality planning (PA)	identify the appropriate quality standards and indicate how they can be met
quality assurance (E)	evaluate project performances in regular manner and indicate if the imposed quality standards are met
quality control (C)	monitor certain project results and determine d if the quality standards are met + identify necessary changes which can ensure the fulfillment of imposed quality standards


Initiation
What must be done? („scope")
Explore the alternatives (e.g. with" brainstorming")
Preliminary planning: WBS, Gantt, resources&costs, risks, procurement, quality, team, communication
Evaluate the approaches: cost/ profit
Select the best alternative: „grid analysis" (multi-objective decision)
Detail the plan for the selected approach
Implement + monitor/control
Finalize
?
Vision
Analysis of the the current context: SWOT
Objectives SMART: „drill down" technique

5. 2. 1. Quality Planning (PA)

= identify the relevant quality standards and the procedures needed for their fulfillment

HOW QUALITY CAN BE OBTAINED?



Quality management plan, list of activities aiming quality assurance

List of major deliverables

Requirements: set by clients, internal, standard

Standards + quality criteria (for each deliverable)

List of related documents: who is in charge with elaboration/ verification/ execution

(testing and integrating plans, testing specifications, testing reports)

Testing frameworks

List of activities meant to quality assurance:

objective, person in charge (execution, verification), effort estimation, the method used for quality verification, milestones, how performances are reported (see the list recommended at 5.3)

Attributes and metrics used for quality verification, values corresponding to normal state (green), alert (yellow), problem (red)

Verification and validation

Other related documents

Taxonomy, definitions, abbreviations

Recommendations:

- is a **cost analysis** needed?

  costs necessary for quality assurance =

  = costs for preventing + verification

  + costs for corrections (whenever the detected quality level is lower)

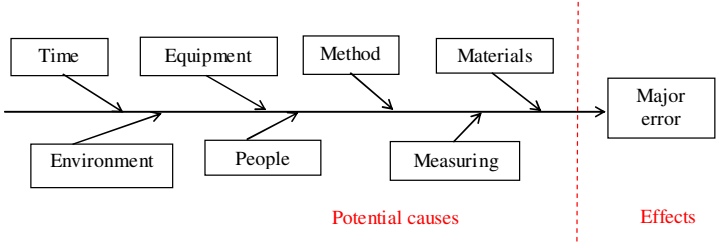
  income is dependent to clients satisfaction, productivity, etc

  !!!! obtain income > costs

- **build diagrams to illustrate the main causes** which can lead to unsatisfactory quality level

E.g: **flowchart;**

Ishikawa cause-effect diagrams („fishbone”) = diagrams which illustrate the cause-effect dependency



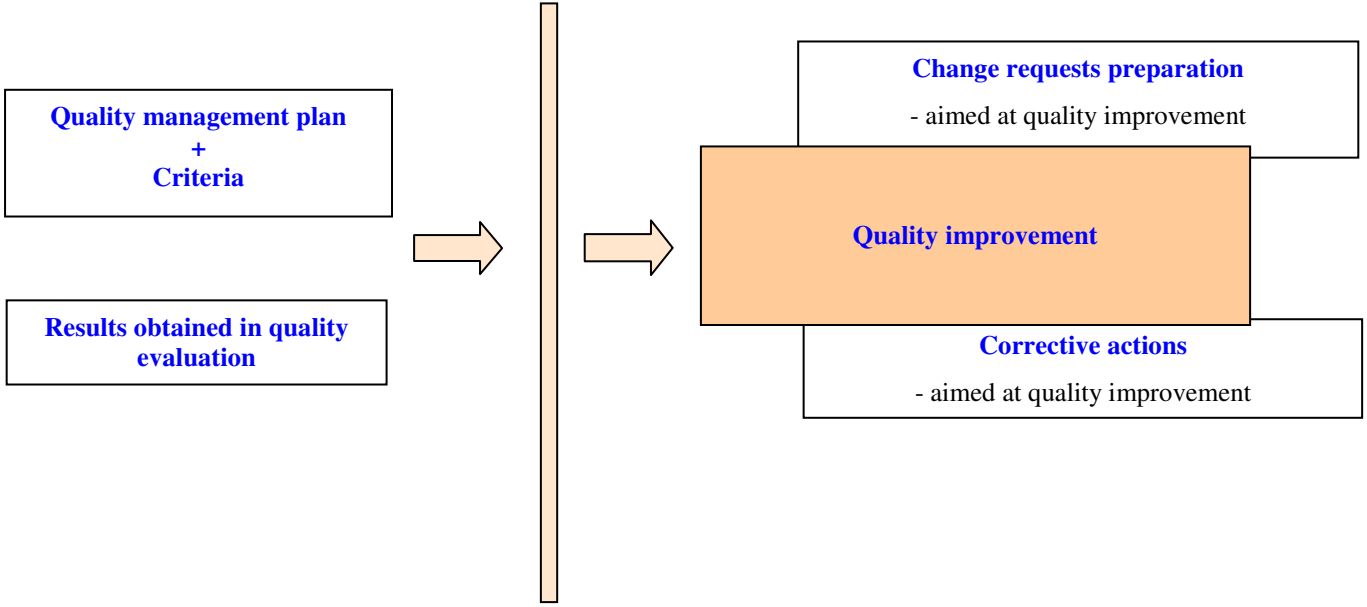
- choose appropriate **benchmarks** – able to provide a relevant portray of quality level

- choose **the experiment sequence** needed for product quality verification (some data can be stochastically generated)
    - the method can be also applied for verifying the plan in different potential contexts (risks, different management strategies, etc.)

5. 2. 2. Quality Assurance (E)

= systematic, planned activities (carried out regularly), aiming the delivering of desired quality

DOES THE PROJECT MEET THE DESIRED QUALITY LEVEL?



Recommendations + remarks:

- the organization can ask for **supplementary audits** - external/internal, planned/unplanned
  - the audit illustrates how quality management is fulfilled within the project,
  - what lessons can be learned within the organization

- some organizations have a **specialized quality department** which can give proficient recommendations

- **proactive actions** are preferred to corrective ones

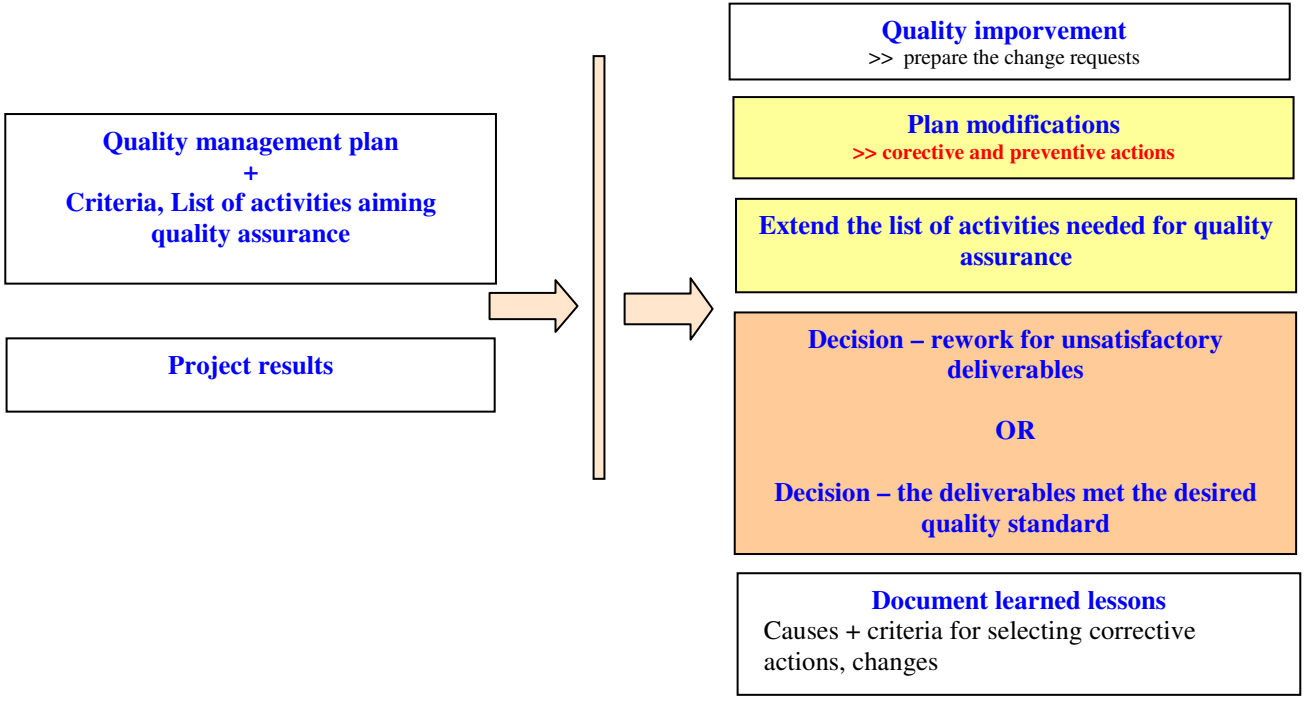
5. 2. 3. Quality Control (C)

= monitor certain results of the project for determining if the adopted quality standards are fulfilled
+ identify corrective alternatives
+ manage necessary changes

DID YOU ACHIEVE DELIVERABLES COMPLIANT WITH THE AGREED QUALITY STANDARDS?

WHAT MISTAKES ARE YOU DOING?

HOW CAN YOU CORRECT?


Quality management plan
+
Criteria, List of activities aiming quality assurance
Project results
Quality imporvement
>> prepare the change requests
Plan modifications
>> corective and preventive actions
Extend the list of activities needed for quality assurance
Decision – rework for unsatisfactory deliverables
OR
Decision – the deliverables met the desired quality standard
Document learned lessons
Causes + criteria for selecting corrective actions, changes

Recommendations:

- Careful evaluation of results is needed:

  Inspection = measuring + examination + testing, in order to see if the requirements are met

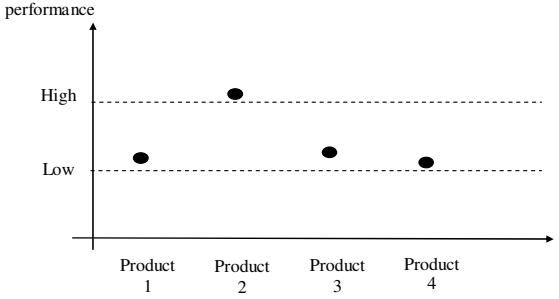
Inspection	≠	**Prevention**
(the error is not visible to the client)		(the error is avoided)

Types of inspections:

- Peer review – possible when confidence and expertise are available;
  advantages: team members communicate, learn one from another, reduced impact if someone leaves the project
- External – no internal expertise available
  advantages: objectivity, accuracy
- Walk around (= done by PM)
- Statistical

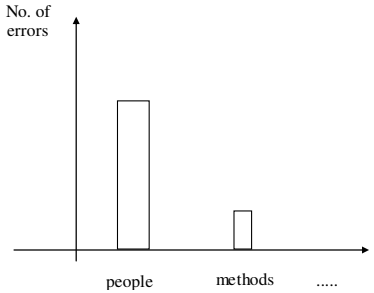
- If possible, determine **variable sampling** useful for prediction

For repetitive actions, variable sampling plot corresponding to similar products:



7S Rule: if 7 consecutive products have accepted quality level, but close to the lower boundary, than there is a detectable cause (are corrective actions necessary)

- Any modification/correction must be motivated by a profound analysis:
  - **Re-analyze** the flowcharts + fish-bones diagrams
  - Elaborate the histogram of errors subject to potential causes



Pareto law: 80% problems are produced by 20% causes (law 80/20)

- **Analyze the tendencies** of the project and make prediction for the next time interval
  - technical performances analysis: predict the number of errors in relation to the errors already met
  - project management performances analysis (time, cost, etc): predict duration extension, in relation to current delays

- Make correct and complete reports

  + cumulative reports: from the beginning of the project

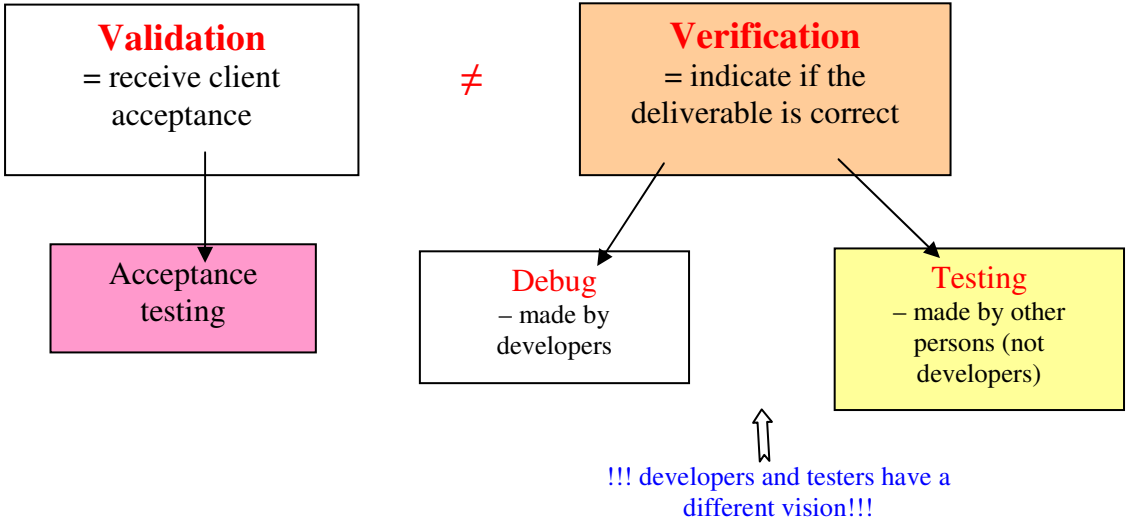
  + summary reports – for managers

  + reports focused on plan modifications

- Use **only** agreed procedures for change requests management
- **Notify** the changes to all the stakeholders

  >> + integrating

5. 3. Software Testing



Remarks:

- The costs for correcting the errors in different project phases

  implementation: testing: deployment = 1: 10:100

- Testing involves about **30% total project cost**

- Main causes of errors: **requirements 60%, design 30%, coding 10%**

Testing styles:

- top - down (rapid detection of major errors; it requests stubs)
- bottom - up (easier generation of testing scenarios)

- white-box – rarely used at testing, often used at debug
- black box – partitioning the input space in clusters, tests at the boundaries of the input space, etc.

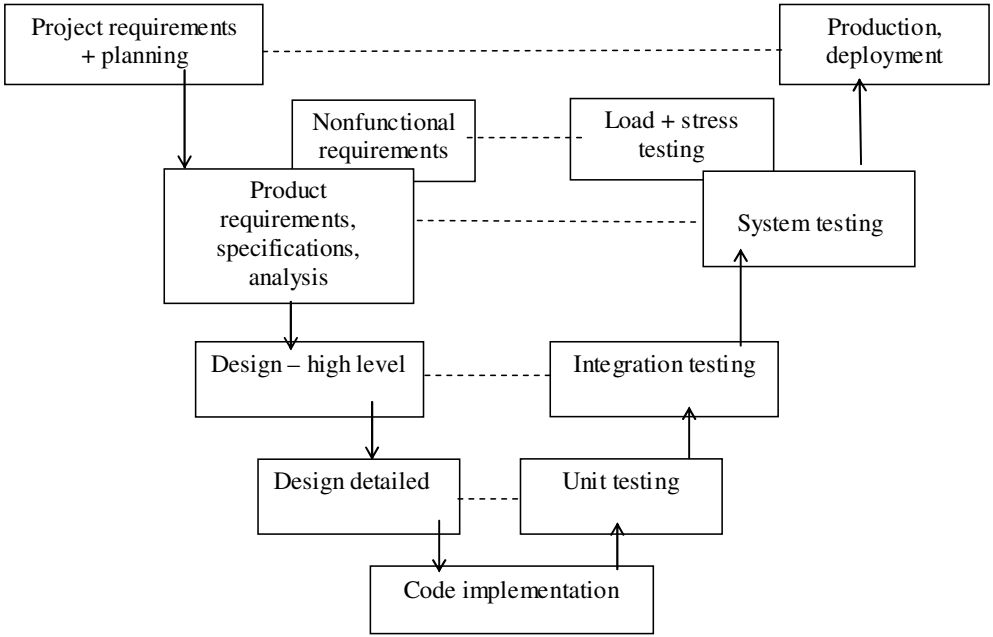
What is tested?

Testing type:

- unit – test the functionalities (black box)
- integration – test the interfacing between the components (black box)
- system - test run-time, stress, load, recovery and security performances
- beta testing for validation (using the acceptance criteria of the client)

!!!! Automatic testing is preferred: after any code change, the whole set of tests should be repeated (regressive testing)

V & V Model (verification and validation)



Is the software sufficiently tested?

- Testing cannot guaranty complete elimination of errors, it cannot be exhaustive!!!

Subject to its efficiency, testing can be

- **„Toy test”** = non-systematic, on few data
- **Professional testing**

  = with systematic cases selection, on large data set

  >> good coverage

  + automatic testing (scripts + tools)

Indicators for evaluating the efficiency of testing

- **Test coverage =** how much was tested (%) - branches visited, instructions executed at least once [<100%]
  - automatically computed by the testing software

- **Fault seed effect**

  Step 1: seed the program with *NS* errors

  Step 2: run the testing scenario and detect *NF* errors, from which *NFS* inserted at step 1

  Step 3: NF-NFS ..... ?
  NFS ..... NS

  >>> $q = \frac{NF - NFS}{NFS} \cdot NS$ - as small as possible

Recommendations for project plan – activities related to testing

Activities specific to testing

Set the objectives of the testing phase (in Requirements phase)

Design the testing scenarios

Elaborate the testing scripts

Test the testing scripts

Execute the scripts + record the errors

Evaluate the results

Correct the errors – with the developers

Repeat the tests on the corrected code

Testing specifications – suggested structure

Project identifier, PM

Name, the position of the person who proposes the scenario test

Objective of testing, tested module/component

Testing framework - description

Detailed description of the testing cases manual and automatic

For each testing case:

Identifier,

Priority

Preconditions: if other preliminary tests are necessary, initial state

Exact sequence of steps, with expected parameter values, etc.

Description of the sequence of tests– e.g. by means of a sequence diagram

What could be detected: for every testing case, indicate tested requirements/characteristics (in a separate table)

Teams: for test design, test executions, report elaboration and verification
History of changes
Other related documents – product specifications, etc
Terminology, abbreviations, definitions

Acceptance plan – recommended structure

List of major deliverables

Acceptance tests for every deliverable

Performance indicators agreed for acceptance

Delivery deadlines/delivery schedule

Testing report – recommended structure

Project identifier, PM
Name, position of test coordinator
Type of testing, tested module/component – testing sequence identifier
Testing framework
Test cases run and their results - summary
    No. of tests done/planned
    Nr. of tests failed/successful
    Degree (%) of success
Important unsolved problems: description, priority, etc
Test cases run and their results – detailed description
Remarks: e.g. – execution time bigger than expected
Testing team, persons who verified the testing report
Time requested for testing
Other related documents – testing specifications, etc
Terminology, abbreviations, definitions

Revision

Definitions, taxonomy:

quality / grade, quality standards

testing (categories), verification/validation, model V&V

flow / Ishikawa diagrams, rules 7S, 80/20, etc

Quality management processes: quality planning (PA), quality assurance (E), quality control (C)

Documents

Quality management plan, list of activities devoted to quality assurance, indicators

Testing specifications

Acceptance plan

Testing report