On Approach to Training in Methods of Software Working Out

Authors:
V.A. Kurylev

Financial University Under The Government Of The Russian Federation, Smolensky branch Smolensk

M.V. Raskatova
A.S. Anisimov
P.V. Shchegolev

Department of Computers, Systems and Networks Moscow
On Approach to Training in Methods of Software Working Out

INTRODUCTION

Training in methods of software working out ranks high in many basic educational programs in data systems designing and working out.

This discipline attends in a relative new specialty “Business-IT” as well to train not the software professional designers only but the specialists capable of rendering IT-decisions in various areas of plants and organizations activities.

Here, the ability of grasping the software development problem on the side of end user prevails as well as knowing those basic requirements to software quality maintenance, meeting of which will guarantee the success in project implementation.
DEVELOPMENT OF SPECS FOR SOFTWARE IN CUSTOMER EYESIGHT

The demand originates for a truncated and in-depth approach to software quality description, on one hand, understandable for end user and, on the other hand, allowing a technically complete description of software development problem statement.

Analysis of software development due allowance for reliability is proposed to be used as said method. Requirements to reliability are urgent at all software development stages, therefore it may be pointed out that the enumeration of such requirements will cover the entire process of software production and be easy-to-understand for every end user.

The major software development stage consists in working out the Specification to formulate the requirements to software quality and requires business-IT specialist skills of distinction the problem from the end user side the interests of which this specialist represents.

Ipso facto, the Specification is the document containing the Customer requirements to the Contractor.

In practice the Specification can be the result of Customer, Contractor work or their combined activities.
Control over implementation of preset requirements of the specification does not, usually, makes problems for the Customer which can normally formulate the list of operating capabilities of the software.

While control over software quality requires determination of some properties of software, on one hand, defining the software quality and, on the other hand, being clear from intimate impulse and technically the user.

The user considers one of the main software quality criterion to be the fact that software runs in compliance with preset functions without faults to respond adequately to hardware and software critical situations, possible defects and errors of applied nature including fool proofing. These properties are what provide the requirements to software reliability.

Ipso facto, reliable software should feature the resistance to errors, serviceability and no-failure operation.
REQUIREMENTS TO SOFTWARE RELIABILITY

• selection of software development model (life cycle model) and execution of established order;
• reliability analysis at every stage;
• planning;
• application of the standards;
• formulation of quality requirements;
• tracing;
• verification;
• control over configuration;
• independent verification of the results;
• adequate processing of critical situation;
• testing;
• abnormal tests;
• test coverage;
• error correction procedure;
• documenting
Selection of the software development model (life cycle model) and execution of established order

- The purpose of life cycle model consists in that it defines the sequences of software development actions from the designing to industrial operation.

- The basic thing here is that the model sets up the regulated process divided into stages.

- Objectives and procedures of their implementation are set up for every stage.

- Attaining these goals is a guarantee of reliability provision and successful implementation of the project.
Analysis of reliability at every stage.

- Reliability requirements should be examined at a time with the requirements to functionality and modularity.

- Every development stage and every functional requirement should allow for reliability factors.

- It should be defined in advance which requirements to functionality and modularity are strictly regulates and which are not.

- At development of software it is necessary to, first, implement the defined earlier rigid requirements to functionality and modularity, while other requirements may be implemented on the fly.
Planning

✓ Planning of the software development model implementation is the key factor for reliable software creation.

✓ It is impossible to create the reliable software without detailed plan of efforts and actions aimed at its implementation.

✓ For each software life-cycle model, detailed composition standards and content of development stages exist.

✓ Every stage should incorporate an additional analysis tool to define possible occurrence of errors, failures, accidents and ways of their troubleshooting.
Application of standards

• Reliable software development should meet the requirement of the maximum application of standards.

• Observance of commonly known standards ISO and GOST guarantees automatically a definite level of software reliability and quality.

• Standardization covers in fact all processes of the software life-cycle.

• The standard of program code writing is important for reliability and further maintenance that exploits such phenomena as using the comments, desirable or adverse structures of programming language, programming style, rules of composing the names of variables and procedures, and the other.
Formulation of qualitative requirements

- Clear and full requirements to software functionality and modularity are the basis for development process on the whole and reliability provision in particular.

- These requirements should be correct and specific, sequential and self-consistent.

- They may be divided into two classes, that is, high- and low-level classes.

- High-level requirements are constructed on the basis of system requirements, requirements to software functionality and modularity.

- Low-level requirements relate to the process of implementation in constructed architecture to make the on their basis.
Tracing

- Tracing is designed for tracking and analyzing the links between software various modules.
- Tracing is very important for modularity verification especially if specific nature of software being developed uses a parallel or dependent-chance programming.

Verification

- Reliable software concept includes verifying not only program code but the results of life cycle other process as well.
- Detection of errors, defects, incompleteness in design solutions, in requirements, in documentation. In program code
- Detection of bottlenecks in the system being created, that is, the points most subjected to errors and critical in terms of system reliability maintenance on the whole
Control over configuration

- Control over configuration is the control over everything created at all software development stages.

- Uncontrolled change of worked-out plan parameters results in output software product reliability and notable deviation from planned requirements.

Independent verification of results

- Software development processes verification procedure should conducted by personnel not involved directly in said processes.

- Employment of technical support service personnel at testing and pre-prod will efficient
Adequate processing of critical situation

- Requirements to software reliability necessitate an adequate behavior of the program at various failures.
- This means that given the failure in software, the computer may not inform about OS error, freeze up and stop “dumbly”.

Testing

- Internal and external testing.
- Inner testing means testing performed by the designers of for code writing verification.
- External testing stage comes up at system testing in the package, jointly with the Customer representatives. Test procedures should be worked out in advance, described in the plan of verification and adhered to unavoidably.
Abnormal tests

- It is necessary to foresee the tests for program stability under various abnormal conditions:
  - improper program initialization;
  - erroneous structure (format) of input data etc.

Test coverage

Planning of test coverage (quantity and complexity of tests) makes one of the tools designed to make sure that the developed code complies completely with the preset requirements.

Error correction procedure

- It is necessary to work out the error correction mechanism and mention it in the verification plan.
- Detected errors may not be corrected without control.
CONCLUSION

Execution of cited requirements will be a guarantee of the developed software quality.

Clear and well-founded account of software quality guarantee by the Realizer makes a basic premise for signing a contract and successful execution of the project.

Approach to software development problems described in this article in terms of reliability has been used in teaching the specialized training business-IT courses and at execution of practical software design work.
Thank you for attention!

Speaker’s contacts:

M.V. Raskatova
Department of Computers, Systems and Networks Moscow

e-mail RaskatovaMV@mpei.ru
web-site