FACET CLASSIFICATION OF SOFTWARE QUALITY MEASURES

G. P. Kozhevnikova and A. A. Stognii

The paper considers a multiaspect classification of software quality measures covering the main directions in software science. The conceptual tools of facet description of software quality measures are introduced, classes of evaluation functions are synthesized, and the usage frequency of measures of various categories in software science is considered.

Software quality problems that first arose in the late 1970s moved to the forefront of programming science in less than a decade. An independent branch of programming technology has crystallized, focusing on the development of theoretical principles, methods, and tools for the development of correct, reliable, efficient, and easily maintainable software. One of the central issues in laying the theoretical foundation for the research in this area was the development of a suitable apparatus for metric analysis of programs, i.e., objective measurement of quality characteristics of the software product.

The advent of the science of metric properties of programs (program metrology or software science) was anticipated by Academician A. A. Lyapunov back in 1959 [1], but serious and constructive work in this direction was done only in the last decade (see, e.g., the monographs [2-7], the surveys [8-15], and the bibliographic reviews [16-18].

Seven main directions can be distinguished in the metric theory of programs today:
- software reliability measurements, intended for predicting program failures and fault situations during execution;
- structural complexity measurements of software modules and intermodule connections, intended for testing and maintaining the modules in operational state as well as minimizing the intermodule data flows;
- program efficiency analysis, intended to increase the software productivity and cost effectiveness by improving resource utilization characteristics;
- measurement of readability and understandability of program code, intended to assess various psychological factors that affect modification and maintenance of programs by persons who are not the original developers;
- analysis of usage frequency of language constructs in programs, intended to improve compiler quality, refine the programming language tools, and standardize the programming style;
- studies of program development effort and programmer productivity, intended to ensure more accurate prediction of software development deadlines, work scheduling, and financial budgeting of software production;
- measurement of generalized software quality characteristics by a collection of criteria.

The following aspects are of key importance for each of these directions:
1) establishment of measures for effective ranking of programs by a given quality criterion from measurement of certain quantitative characteristics;
2) development of quality evaluation methods;
3) development of support tools and technologies for software quality measurements.

Software quality research suffers from absence of a complete objective diagram of relations on the set of program properties, fuzziness in the definition of various properties, inaccessibility of fuzzily defined properties to direct measurements in programs, existence of conflicting properties, and ambiguity of quality criteria and measures, which are usually chosen from subjective considerations of what is relevant for evaluating specific program properties and how it should be measured. All this has led to a variety of evaluation measures, methods, tools, and procedures in each of the areas of software science listed.
Thus, more than twenty measures have been proposed by now for evaluating structural complexity of programs [8, 9, 17], but none of these measures orders programs by this criterion in a way that is consistent with programmers' notions of relative complexity of the same programs.

The validity of the choice of measures and measurement methods for specific program properties depends primarily on the completeness of description of the available alternatives. It is accordingly necessary to develop a systematic classification of measures, measurement methods, and measurement tools in software science. The aim of this paper is to construct a multiaspect classification of software quality measures that encompasses all the main directions of software science listed above.

1. CONCEPTUAL TOOLS OF FACET DESCRIPTION OF SOFTWARE QUALITY MEASURES

Our classification of software quality measures is based on Ranganatan's facet classification systems (also known as synthetic or analytic-synthetic classification systems), which are much more powerful for expressing meaning than the enumerative, in particular hierarchical, classifications [19].