Testing a new approach to the analysis of projects development using generalization parameters offered by software science

A. V. Gahov
V.N. Karazin Kharkov National University, Kharkov (Ukraine)
gahov@univer.kharkov.ua

Abstract

The paper is dedicated to the testing of the Mishchenko's new approach to the metric software description that has been suggested by M. Halstead. The measurement of an author's project is carried out. We succeeded in the interpretation of the measurement results.

1. Introduction

Article [1] suggested using the relation between different assessments of the programming work introduced in [2] as a specific feature of software tools belonging to the same class. When discussing software quality problems, we are going to use concepts and definitions given in [3, 4]. Developing the approach described in [1], characteristics making it possible to assess the implementation efficiency of a complex software system were introduced in [3]. In this case the introduced metrics serve as software metrics. However, it was mentioned that it was possible to use them as metrics of the development process. To be more exact, the above-mentioned work specifies how to apply these metrics as well as ones derived from them to the analysis of software development quality in the field of computer-based information systems. The latter would be of practical use since it would enable us to forecast software quality and keep track of suspect situations during the development stage. This is of much importance for computational systems that often have well-encapsulated errors [5].

2. Objective and aims of the study

The objective of the work is to test the new approach [3] to the indication of software quality by means of software science offered lately. The algorithm of monitoring the software development process based on metrical characteristics observations and their versions [3] was the object of the study. As a matter of fact, we can call these characteristics (metrics) 'energy' ones.

The following observations are the matter of the study:

1) Testing [3] as a manual for calculating energy metrics of the software that has an extensive modular and nested block structure.

2) Testing lexical aspects of the above-mentioned technique regarding the language of Maple 8 system.

3) Making a comparison with results of [3] where simple software of this type realizing computational experiments on the basis of discrete singularities methods is given.

4) Comparing the data obtained on the information heat behavior when implementing a real project with informal idea of this project development process.

3. Research methods and results

For to test the above-mentioned approach, we have chosen four sequential versions of the *Diffraction on Rectangle* program [7]. Our choice was stipulated by quite a complex structure as compared with small size thereof. The program comprised 10 Ada modules and one Maple 8 module at the final stages of its development.

It was not difficult for us to calculate the specification energy E in accordance with [3] for each of the versions.

Article [3] uses the idea of potential volume for such modules that include different status blocks. However, there is just a formal rule for a simple case. We had to restore the full definition. By potential volume we mean the sum of potential volumes of all blocks. This definition does not conflict with the special case for which there is an appropriate definition in [3].

Some rules of [3] use the concept of the sequence of modules development. By this sequence we must mean the logical order of the development. We would be wrong if we observe it on the basis of dates of files past modifications.

Article [3] only suggests the rules for determining the amount of programming efforts spent on interface modules developed later than the series of modules within a context of which they are. The programming effort will be understated substantially if we take this hint literally. We realized the hint given in [3] in the form of a precise definition and used it for calculations.

For example, the *diffraction_on_rectangle* module uses specialized output functions described in the *io* module. We know from the development process information that the *io* module was developed later than *diffraction_on_rectangle*. Thus, if we use the precise rule, we take into account *io* modules twice - at the scheduling stage (in *diffraction_on_rectangle*) and directly at the implementation stage (in the *io* module).

Let us proceed with making comparisons to results of work [1]. For the system under examination, it turned out that the specification energy E was approximately equal to the programming work A for complete versions. The value of E made up approximately to 2*A for initial versions. This value is close to the conclusion provided by [1] where E is approximately equal to 3*A. Let us keep in mind that [1] analyzed such program versions that we must consider as initial by the development terms (students' diplomas!).

We have studied parameters ρ and κ [3] as development process characteristics. Two new parameters (σ and τ) to characterize the software development process were introduced:

$$\sigma = \Delta A / E \text{ and } \tau = \Delta Q / E \qquad (1)$$

These parameters have the advantages as compared to those offered in [3]:

1) When $\Delta E = 0$ (that is when the number of parameters is the same for two adjacent realizations), $\sigma = \infty$ and $\tau = \infty$, that provides no data on ΔA or ΔQ. This is not the case for σ and τ since $E \neq 0$.

2) It is impossible to restore the signs of ρ and κ (which are of certain interest) by those of ΔA and ΔQ (in [3]). Parameters σ and τ preserve the data on signs of ΔA and ΔQ.

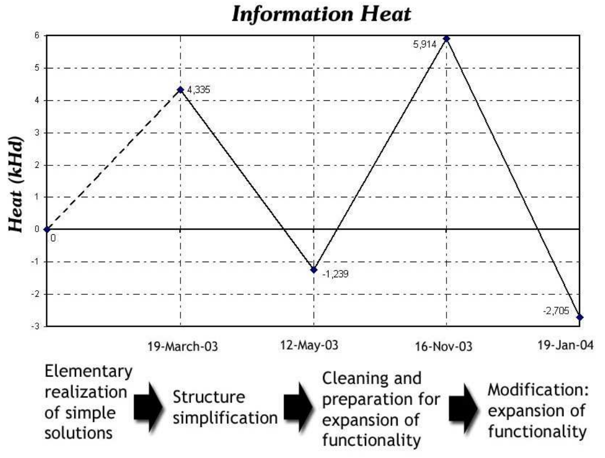

Figure 1. Information Heat

Analysis and interpretation of the results makes the next stage of our research.

Actually, the development process consisted of two cycles. Each of them included the initial version and then the release one.

The initial version suggested and reconstructed the structure and the other one improved and extended the functionality.

As follows from Fig. 1, Fig. 2 and Table 1 for these stages, initial versions always had the increase in the heat ($\Delta Q > 0$, $E > A$, $\tau > 0$) and the other ones had the experience accumulation ($\Delta Q < 0$, $E < A$, $\tau < 0$).

Therefore, results of the calculations and their interpretation in accordance with [3] are in agreement with the qualitative evolution pattern of the program under examination during the project implementation.

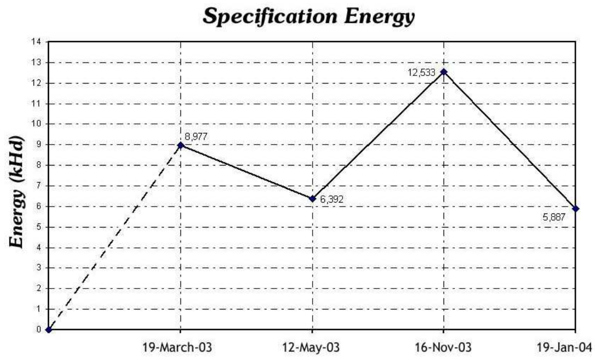

Figure 2. Specification Energy

Table 1. Some parameters of the versions

	body	spec	E	A	Q	σ	τ
19-March-2003	6	5	8.98	4.64	4.3	**0.48**	**0.52**
12-May-2003	12	7	6.39	7.63	-1.2	**-0.86**	**0.47**
16-Nov-2003	12	7	12.53	6.62	5.9	**0.57**	**-0.08**
19-Jan-2004	11	7	5.89	8.59	-2.7	**0.34**	**-1.46**

3. Conclusion

The examination of the software system model (that has been suggested in [3]) by the real example is carried out. From our investigation we can see as follows:

1) It is necessary to add a more accurate definition given in this paper to the potential and development volumes definition (see [3]).

2) The direction of the information heat flow was calculated for 4 versions with information about the development progress. The comparison of these values proves the correctness of the interpretation suggested in [3].

3) With an allowance for the above-mentioned remark, the mathematical model [3] can be put in practice for the manual measurement of energy characteristic of the software in the software science style. We have encountered characteristic measurement in the software science style for software that was developed using multiple programming languages for the first time.

On the basis of our results we can confirm the hypothesis that the Halstead's level for Ada 95 language is about 1.5.

Table 2. The detailed values of the parameters

	V*	E	η all	N	W	V	V spec	A	Q	σ	τ
	integral (integral.adb, integral.ads)										
19-March-2003	206,87	**1560735,3**	97	2529	18305,9	18305,93	1208,5	**1619861**	**-59125,65**	-0,0379	1,0379
12-May-2003	211,96	**1072455,3**	118	2565	17654	17653,98	916,14	**1470386**	**-397930,3**	-0,3159	-0,139
16-Nov-2003	211,96	**1072455,3**	118	2565	2394	17653,98	916,14	**27039,23**	**1045416,1**	1,3458	-1,346
19-Jan-2004	215,36	**1126448,1**	129	3616	25352,6	25352,6	834,42	**2984500**	**-1858052**	-2,5775	2,6255
	io (io.adb, io.ads)										
19-March-2003	308,61	**6702791,5**	98	1945	16528,4	15319,83	2288,3	**885209,6**	**5817581,9**	0,8679	0,1321
12-May-2003	258,03	**4423100,1**	129	1858	13949,5	13949,48	1752,7	**396488,4**	**4026611,7**	-0,4049	-0,11
16-Nov-2003	332,47	**9921041,8**	130	1381	10495,5	10495,48	2310	**331328,3**	**9589713,5**	0,5607	-0,007
19-Jan-2004	223,44	**1715526,3**	138	1206	9138,38	9138,379	1443,4	**373747,3**	**1341779,1**	-4,8078	0,0247
	Diffraction On Rectangle (diffraction_on_rectangle.adb)										
19-March-2003	98,107	**425337,85**	82	482	3064,34	3064,34	0	**95713,18**	**329624,67**	0,775	0,225
12-May-2003	28,529	**10459,284**	58	312	1827,69	1827,69	0	**117088,3**	**-106629**	-41,71	2,0437
16-Nov-2003	53,303	**68215,279**	58	312	1827,69	1827,69	0	**62669,14**	**5546,1435**	1,6444	-0,798
19-Jan-2004	98,107	**425337,85**	118	834	5740,12	5740,124	0	**335846,2**	**89491,677**	0,1974	0,6423

References

[1] Мищенко В. О. Применение математического моделирования в системном анализе проекта программного обеспечения методов дискретных особенностей // Труды VII Международного «Методы дискретных особенностей в задачах математической физики». – Феодосия, 1997. – С.117-120.

[2] Холстед М.Х. Начала науки о программах.- М.:, Финансы и статистика, 1981,- 128 с.

[3] Мищенко В. О. Математическая модель стиля Software Science для метрического анализа сложных наукоёмких программ // Вестник Харк. нац. ун-та. Сер. "Матем. модел. Информац. технологии. Автоматизированные системы управления", вып. 3, - Харьков, 2004г. – С. 70-85

[4] Харченко В.С,, Скляр В.В., Тарасюк О.М, Методы моделирования и оценки качества и надёжности программного обеспечения. – учеб.пособие.- Харьков: НАЦ.Аэрокосм. Ун-т «Харьк.авиац.ин-т». 2004.- 159 с.

[5] Форсайт, Дж., др., Машинные методы математических вычислений, М.: Мир, 1980. - 279 с

[6] Белоцерковский С. М., Лифанов И. К. Численные методы в сингулярных интегральных уравнениях. – М.: Наука, 1985г. – 256 с.

[7] Гахов А. В., Мищенко В. О. "Вычислительный эксперимент на базе численного решения гиперсингулярного интегрального уравнения для прямоугольной области" // Харк. нац. ун-та. Сер. "Матем. модел. Информац. технологии. Автоматизированные системы управления": вып. 1, - Харьков, 2003г. - С.84-91