UDC 681.3.06, 519.76

O.O. SLABOSPITCKAYA

A PROCESS FOR CONSISTENT AND INFORMED ASSESSMENT OF SOFTWARE RELIABILITY OVER ITS LIFE CYCLE

Abstract. *The software process is improved with a new process of unified, informed and consistent software reliability predicting/assessing over software life cycle. It is represented with both the ratio of faults and Function Points (FD) and FD-depending probability of functioning without failures (PD). The model is built and methods are elaborated for the process with Bayesian net and Value tree (to ensure informational continuity and permanent increasing of FD estimates formalized consistency) and J. Musa's reliability model (to PD uniformly assess on their base).*
Key words: *software reliability, residual faults early prediction, Bayesian net, Value tree, viewpoints merging, estimates' consistency.*

Анотація. *Процес розроблення програмних систем (ПС) вдосконалено новим процесом уніфікованого, інформаційно підтриманого та обґрунтованого прогнозу й оцінювання надійності ПС у життєвому циклі ПС. Її подано парою: відношення кількостей залишкових дефектів і показників функційності (FD) та FD-залежна імовірність безвідмовної роботи (PD). Розроблені модель і методи процесу з використанням мережі Байєса та Дерева цінності (на підтримку інформаційної спадкоємності й підвищення формалізованої обґрунтованості оцінок FD) та моделі надійності Дж. Муси (для уніфікованого оцінювання PD на їх ґрунті).*
Ключові слова: *надійність програмної системи, раннє прогнозування залишкових дефектів, Байєсова мережа, Дерево цінності, узгодження поглядів, обґрунтованість оцінок.*

Аннотация. *Процесс разработки программных систем (ПС) усовершенствован новым процессом унифицированного, информационно поддержанного и обоснованного прогноза/оценивания надежности ПС в жизненном цикле ПС. Она представлена парой: отношение количеств остаточных дефектов и показателей функциональности (FD); FD-зависимая вероятность безотказной работы (PD). Разработаны модель и методы процесса, использующие сеть Байеса и Дерево ценности (в поддержку информационной преемственности и непрерывного повышения формализованной обоснованности оценок FD) вместе с моделью Дж. Мусы (для унифицированного оценивания PD на их основе).*
Ключевые слова: *надежность программной системы, раннее прогнозирование остаточных дефектов, Байесовская сеть, дерево ценности, согласование взглядов, обоснованность оценок.*

1. Introduction

Continuous increasing of software reliability still remains crucial in the development of all types of keen software. Current standards for software process (ISO/IEC 12207, 15504, 15939, IEEE 1012:1998, ANSI/IEEE 1008:1987) and for safety (IEC 61508 etc.) determine general requirements for software quality and configuration control and due techniques for its design being assumed to increase reliability through reducing the faults. But no quantification of faults number and corresponding reliability after meeting these requirements is given [1]. It makes impossible quantitative monitoring of software process's (interim) products quality and informed choice of alternative strategies based on the estimates obtained.

At the same time, the SW-CMM [2] just fixes such quality evaluation as essential premise to increase the organizational maturity. Moreover, ISO/IEC and IEEE standards for V&V and SQA processes as well as Software Engineering Institute's (SEI) risk management [3] clarify four kinds of it. These differ in agents (decision maker alone or with experts representing stakeholders' viewpoints) and in input/output information certainty (probabilistic or deterministic one). But the above guidelines require the results of all these evaluations to be uniform, shared, viewed within the organizational context and continuously used for new evaluations and (interim) products improving whole over the software life cycle (SLC).

Let the single assessed characteristic of software quality be further its reliability (R) represented with both the ratio of faults and Function Points named the residual faults density (FD) (as an inner metric) and the probability of functioning without failures $(PD(FD))$ (as the outer one).

Being widely used at the late SLC phases, multiplicative regression models for FD [4] do fail at the early ones. There the Bayesian net (BN) provides most mature individual inferring the FD distribution from its affecting factors' priors and fault evidence. But BN alone cannot support three other kinds of FD evaluation and therefore needs adjunction of MCDA model adequate to their specifics [5]. R.Keeney's Value tree [6] seems to be perspective in such a role.

The article drafts the new process for informed and consistent prediction/assessment of reliability $R=\langle FD;PD\rangle$, unified and continuous over all SLC phases combining the techniques of software quality management with early BN-based FD prediction [7] and its VT-based uniform expert assessment [8].

2. Reliability assessment framework

The methodological framework proposed for the above process composes five basic constructs:

1) the vision of software fault;

2) the interrelated templates of BN and VT for FD evaluation;

3) the model for PD evaluation based on FD estimates being obtained;

4) the model of the above process;

5) the methods for uniform, informed and consistent FD evaluation with BN and VT over SLC.

The fault is uniformly viewed within the US NASA risk-based perspective [9] as a feature of SLC phases' interim product potentially causing a loss of final software quality. This vision inspires three types of faults, namely Requirements-, Design- and Code-caused ones.

Figure 1 depicts the evaluation templates that put such a vision into software process practice.

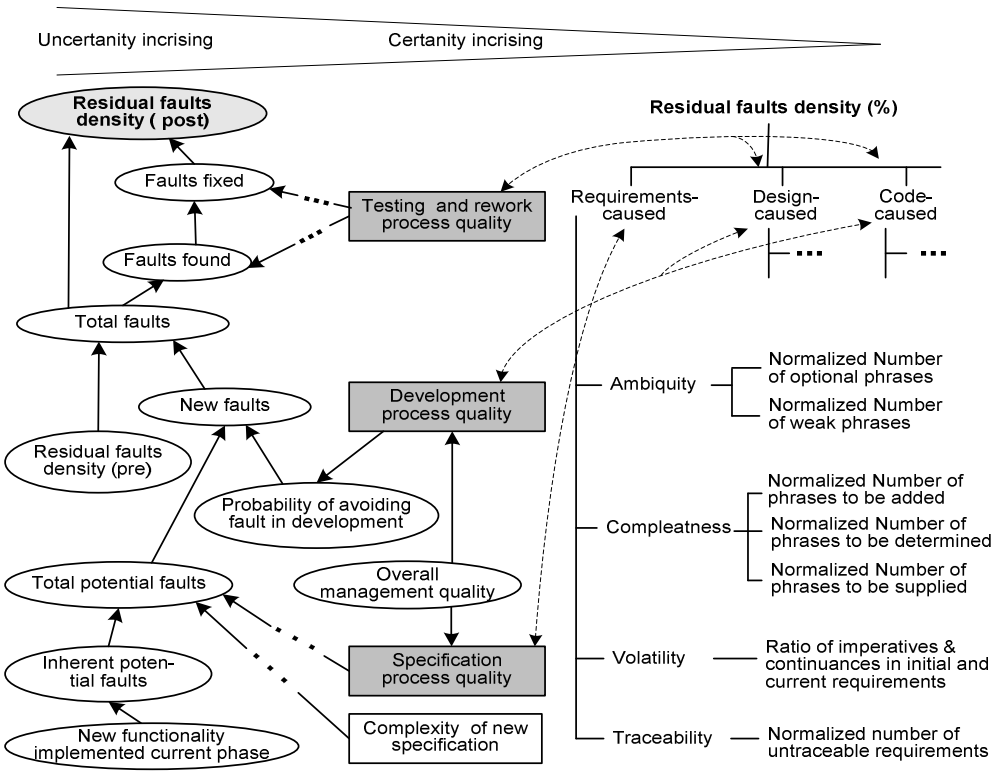

Fig. 1. Interrelated evaluation templates in BN and VT form

On the left is slightly simplified schematic view of BN for the prediction of various types of faults at SLC phases adopted from [10]. Ellipse and rectangle indicates here a BN node and a subnet not needing to be shown. On the right VT is shown built on the base of the software quality model within NASA perspective. The dotted arrows link the upper nodes of BN and VT that represent the same fault affecting factors, but need probabilistic estimating or deterministic one. Such correlation enables these templates to mutually fit each other.

Comparative analysis of modern reliability models [11] clarifies the availability for PD assessment J. Musa's model

$$PD(t)=exp\left[(exp(\rho(S)\times t)-1)\times FD\times S\right], \quad (1)$$

where S is the number of Function Points in software being developed;

$\rho(S)$ – the model parameter depending from the processor rate and the programming language being used;

t – desirable term of software functioning without failures.

The fourth basic construct of the above framework, namely the model M elaborated for new process of reliability $R=\langle FD;PD\rangle$ uniform prediction/assessment over SLC, is formally a triple

$$M=\langle Ag;En;Rm\rangle, \quad (2)$$

where Ag, En and Rm denotes the agents, the environment and the sub-model for a round of this process.

Fig. 2 reflects the formula (2).

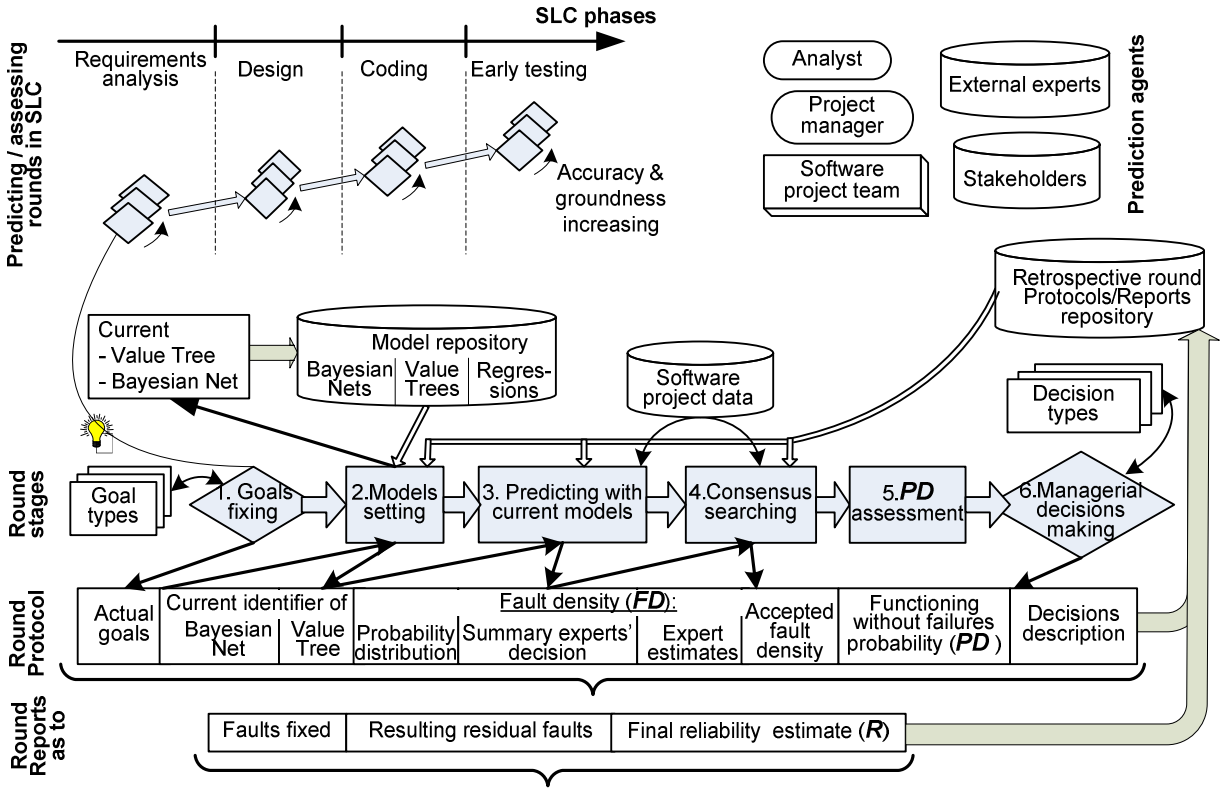

Fig. 2. The model for process of residual software faults continuous prediction/assessment at SLC phases

It stresses that the above process is defined to be a series of unified Rm-modelled rounds of R prediction/assessment being repeated over SLC phases by agents Ag in common environment En. Each round (detailed with a "lamp") produces the Protocol as to R estimate (together with its rational and managerial decisions based on it) and two Reports. The last concern the faults currently fixed in the round and resulting actual state of residual faults.

Round model Rm from formula (2) enables Ag to use not only current data as for software developing but also both BN/VT models and estimates from all previous rounds stored in due repositories. The estimates are summarized with their rationales in knowledge structures being reflected with the Protocol. The repositories may be used whether informally or formally with BN/VT techniques. These provide accounting the above-mentioned kind of evaluation and permanent increasing of accuracy and formalized consistency of resulting estimates over all SLC phases.

Agents Ag are represented with Fig. 2. The individual ones are the Decision Maker (analyst, external expert's team leader, SQA or V&V group member, software project manager or technical leader), an Interviewer (External expert's member) and an Expert (practitioner in safety-related systems development, project member or representative of Stakeholder perspective). External expert and Stakeholder are the institutions that coach safety monitoring over SLC and have a specific viewpoint affected by decisions being made (e.g. government agency, corporation, Armed Forces branch etc.).

The environment En (see formula (2)) consists of six repositories: fixed generic BN fragments that have been found to be the basis for most BN in reliability assessment, Software project data, Unified types of goals and managerial decisions in a round, BN/VT Models (initially with the above templates) and round retrospective Protocols/reports (initially empty). The last four are permanently filled up at SLC phases.

Fig. 2 demonstrates the inner stages in Rm to provide the framework benefits with BN/VT techniques while the rest stages are ill-supported.

The Model setting stage enables Decision Maker to state FD model (BN, VT or the both with common nodes) through building, selecting in Model repository or tailoring selected model(s) to the project with the methods drafted hereinafter in Section 3. Interrelated current BN and VT may be mutually fit through setting the centres of most probable intervals for BN nodes as their generalized expert estimates in VT. The mean values of those nodes in BN may conversely be their estimates in VT. Then BN-based mean and VT-based estimate of FD should be compared. At the Testing phase and later BN and VT may also be verified with testing data and predictions based on regression models like proposed in [4].

At the next stage of FD evaluating under current models well-known Lauritzen-Spiegelhalter propagation algorithm [1] is used for BN. For actual VT (VT_0) experts' group g is collected, their opinions O_j are elicited and summary decision for FD is formed based on $\{O_j, j \in g\}$. An opinion is a tetrad

$$O_j = \langle X_{0j}, VT_j, X_j, R_j \rangle, \tag{3}$$

where X_{0j} are the estimates for prior VT_0 being assumed obligatory;

VT_j, X_j, R_j are respectively (non-obligatory) version of VT, its estimates and VT_0 fallacies under the perspective of expert j.

The Consensus reaching stage is proposed to be a communicative Delphi process based on M.Turoff guidelines [12], where all FD estimates and their rationales obtained are used to cope with well-known Delphi pitfalls.

The fourth stage of PD assessment is performed automatically in accordance with formula (1).

3. Methods for FD evaluation with BN and VT

As for BN, current forming and tailoring methods only include informal composition of fixed BN fragments and dynamic discretisation enabling the continuous nodes of resulting BN to be automatically defined [10]. To overlap the above BN limitations two formalisms are proposed [8].

The first represents VT under the perspective of viewpoint V as a tuple

$$VT(V) = \langle p_x, w_x, r_x, s_x, A_x \rangle, x \in X(V) \subseteq F, \quad (4)$$

where p_x, w_x, r_x are respectively the ancestor in VT, weight and range for x;

s_x indicates should x be estimated $(s_x=1)$ or not $(s_x=0)$ for FD;

A_x points out the methods for x evaluation (e. g. the above BN-based one);

$X(V)$ denotes those commonly recognized factors F affecting fault that are asserted to be the VT nodes by viewpoint V.

The second formalism proposed defines the quantitative consistency index C as a tetrad

$$C(E,VT) = \langle sm, rs, sf, ac \rangle, \quad (5)$$

where E is generalized VT-based estimate for FD;

sm and rs quantify VT_0 and, respectively, VT_j similarity;

sf and ac are respectively the levels for sm significance and VT acceptability.

The methods for VT constructing are ad hoc individual forming and group one by the representative(s) of perspective V. The first method proposes $VT(V)$ (4) to be built whether with A.Landfield's pyramiding technique or with D. Hinkle's implications grid being constructed on the base of pyramiding results accordingly to F. Fransella guidelines [13]. Competence of VT's author is considered to be the $ac(VT)$ index in formula (5).

The second method applies M. Turoff's Delphi communicative procedure proposed in [12] to the set of all hypothetical versions of $VT(V)$ (4). The last are built automatically from the nodes initially belonging at least one expert VT version VT_j in current opinion O_j (3) of expert $j \in g$. Expert estimates being usually used in Delphi-type procedures are here substituted with special indexes proposed in [8] that quantify the stability of FD generalized estimates under VT_j changes.

The tailoring methods include selected VT individual modifying and $VT_j, j \in J \subseteq g$ merging. Two merged VT are proposed [8] with special K. Bogart metric d [14] and method parameters f, k. These are f-choice and (f,k)-mean represented with their matrices R^c and R^m:

$$R^c = \arg\min_{i \in J} f(d(R_i, R_j), j \in J); R^m = \arg\min_{R \in RL(k)} f(d(R, R_j), j \in J);$$
$$ac = f(d(R^x, R_j), j \in J), R^x \in \{R^c, R^m\}, \quad (6)$$

where $R = \|r_{uv}\|_{u,v \in F}$; $r_{uv} = 1$ if in (3) $u = p_x$, $r_{uv} = -1$ if $v = p_u$, $r_{uv} = 0$ otherwise;

f is a multi-argument function invariant to its arguments' inversion and satisfying Pareto principle (e.g. (weighted) sum or product);

l – the number of levels in (f,l)-mean VT;

$RL(l)$ – the set of matrices representing potential (f,l)-mean VT.

The selecting methods comprise BN and VT choice by goal/decision correspondence and only for VT – by the distance to VT given with metric d.

The Summary decision as to FD is proposed to obtain through two steps:

a) analysis of concordance and sufficiency for the set of VT_0-based expert estimates $\{X_{0j}, j \in g\}$ from opinions O_j (3) with d-distances and the above stability indexes [8] regarding VT_j;

b) if $\{X_{0j}, j \in g\}$ are correct, evaluation of all VT_0-based versions for FD generalized estimates being constituted with statistically optimal estimates of VT_0 leaves and non-dominated as to consistency index $C(E, VT_0)$ (5), choice the version E with maximum C and fixing summary decision as a tetrad

$$SD = \langle VT_0, g, E, C(E, VT_0) \rangle. \quad (7)$$

Otherwise, if expert opinions $\{X_{0j}, j \in g\}$ fail or resulting maximum C is still insufficient, VT_j from opinions O_j (3) should be merged and a), b) steps repeated with VT^x (6) and $\{X_{0j}, j \in g\}$ from (3).

4. Conclusions

Current author's efforts aim at:

– the framework proposed technical details to elaborate (such as the reports format, the goals and decisions types, the formalisms for its appropriate representation);

– the above Delphi pitfalls to cope;

– the instrumental tools to develop (combining BN-oriented like Hugin Lite and VT-oriented ones);

– the technical guidelines as to resulting software reliability prediction/assessment framework to elaborate;

– the framework above to carefully test.

The author believes the framework to be obtained might be useful for software development sound organizing and its products' (both interim and final) quality continuous increasing.

REFERENCES

1. SERENE. SafEty and Risk Evaluation using bayesian NEts: SERENE. The SERENE Method Manual Version 1.0. – SERENE Partners, 1999. – 200 p.
2. Capability Maturity Model for Software (Vers. 1.1). CMU/SEI-93-TR-024 / Paulk M. et al. – Software Eng. Inst., Carnegie Mellon University. Pittsburg, 1993. – 82 p.
3. Higuera R. Software Risk Management. CMU/SEI-96-TR-012 / R. Higuera, Y. Haimes. – Pittsburg: Carnegie Mellon University, 1996. – 48 p.
4. Lakey P. System and software reliability assurance notebook / P. Lakey, A. Neufelder. – Rome Laboratory Report, Griffits Air Force Base. Rome NY, 1997. – 186 p.
5. Fenton N. A critique of software defect prediction models / N. Fenton, M. Neil // IEEE Trans. On Soft. Eng. – 1999. – Vol. 25, N 5. – P. 675 – 689.
6. Keeney R. Value-Focused Thinking: A Path to Creative Decisionmaking / Keeney R. – London: Harvard University Press, 1996. – 432 p.
7. Lavrishcheva E. An approach to the software quality management / E. Lavrishcheva, G. Koval, T. Korotun // Cybernetics and Systems Analysis. – 2006. – Vol. 42, N 5. – P. 174 – 185.
8. Lavrishcheva E. An approach for expert assessment in software engineering / E. Lavrishcheva, O. Slabospitckaya // Cybernetics and Systems Analysis – 2009. – Vol. 45, N 4. – P. 151 – 168.
9. Hyatt L., Rosenberg L. Software metrics program for risk assessment. – Greenbelt: International Astronautical Federation, 1996. – 11 p.
10. Fenton N. Improved Bayesian Networks for Software Project Risk Assessment Using Dynamic Discretisation / N. Fenton, L. Radliński, M. Neil. – London: Queen Mary, University of London, 2007. – 10 p.
11. Farr W. Software Reliability Modelling Survey / W. Farr // Handbook of Software Reliability Engineering. – McGraw-Hill: Ed. M.R. Lyu. – 1996. – P. 71 – 117.
12. Turoff M. The Delphi Method: Techniques and Applications / M. Turoff, H. Linstone. – London: Addison-Wesley Publ., 2002 – 608 p.
13. Fransella F. The Essential Practitioner's Handbook of Personal Construct Psychology / Fransella F. – Chichester: Wiley, 2005. – 308 p.
14. Bogart K. Preference structure II. Distances between asymmetric relations / K. Bogart // SIAM J. of Appl. Math. – 1975. – Vol. 29, N 2. – P. 254 – 270.

Стаття надійшла до редакції 26.05.2009