Ein risiko-orientiertes statistisches Software-Testverfahren

B. Krzykacz, Garching

Zusammenfassung

Der vorliegende Beitrag befaßt sich mit einem statistischen Software-Testverfahren unter dem Gesichtspunkt des Risikos. Risiko ist definiert als der Erwartungswert des durch Softwareversagen bedingten Verlustes im künftigen betrieblichen Einsatz. Das Testverfahren liefert statistische Konfidenzaussagen über dieses Risiko; es ermittelt

a) eine obere Konfidenzgrenze für das Risiko zum Konfidenzniveau von (z.B.)95%

b) die Anzahl der durchzuführenden Testläufe, die notwendig sind, um eine vorgegebene obere Konfidenzgrenze für das Risiko einzuhalten.

Abstract

This paper introduces a statistical software test procedure with respect to risk. Risk is defind as the expected loss due to software failure in real operation. The main results of the procedure are some confidence statements about the risk; it evaluates

a) an upper (e.g. 95%) confidence bound for the risk

b) the smallest number of test cases required in order to verify some specified figure r to be the upper condidence limit for the risk.

1. Einführung

Die meisten statistischen Software-Testverfahren befassen sich mit der Berechnung oder Abschätzung bestimmter Software-Zuverlässigkeitskenngrößen wie MTTF (=mittlere Zeitdauer bis zum Versagen), MTBF (=mittlere Zeitdauer zwischen Versagensfällen), Versagensrate, Versagenswahrscheinlichkeit pro Anforderung, Anzahl der Fehler im Programm, usw. (vgl. [1],[2] und Schrifttum darin).

Diese Verfahren berücksichtigen in ihrer ursprünglichen Form weder die Möglichkeit unterschiedlicher Bewertung von Programmfehlern noch die Möglichkeit unterschiedlicher Folgen und Auswirkungen von Programmversagen im betrieblichen Einsatz.

Der vorliegende Beitrag befaßt sich daher mit einem statistischen Software-Testverfahren unter dem Gesichtspunkt des Risikos. Risiko ist hier definiert als der Erwartungswert des durch Softwareversagen bedingten Verlustes. Es setzt sich zusammen aus den Wahrscheinlichkeiten für Programmversagen und den Verlusten (gemessen in geeigneten Einheiten) die durch Programmversagen verursacht werden.

Der Test gehört zu den sog. geschichteten Testverfahren, bei welchen der Eingabedatenraum in Teilmengen (=Schichten) zerlegt wird und die Eingabedaten für die durchzuführenden Testläufe aus diesen Schichten in geeigneter Weise zufällig ausgewählt werden.

In diesem Beitrag erfolgt die Zerlegung des Eingabedatenraums in Schichten nach dem sog. Verlustprofil und die Zufallsauswahl der Testläufe nach dem sog. geschichteten Auswahlverfahren [3,S.318-334].

Das Ziel unserer Betrachtungen ist die Ermittlung
a) einer oberen Konfidenzgrenze für das Risiko zum Konfidenzniveau von (z.B.)95%.
b) der Anzahl der durchzuführenden Testläufe, die notwendig sind, um eine vorgegebene obere Konfidenzgrenze für das Risiko einzuhalten.

2. Grundlegende Bezeichnungen, Definitionen und Annahmen

Der Eingabedatenraum für das zu testende Programm wird in folgender Weise in Schichten zerlegt:
Es seien $L_1, L_2, \ldots, L_K$ alle möglichen Verluste (gemessen in geeigneten Einheiten, z.B. DM) die infolge eines Programmversagens im Betrieb entstehen können. Die Schicht h ist definiert als die Menge

aller Eingabedaten die im Falle vom Programmversagen zum Verlust L_h führen, h=1,...,K.

Auf diese Art wird der gesamte Eingabedatenraum vollständig in dispunkte Teilmengen (=Schichten) zerlegt.
K = Anzahl der Schichten.
Sei π_h die Wahrscheinlichkeit für die Schicht h, d.h. die Wahrscheinlichkeit dafür, daß im betrieblichen Einsatz Eingabedaten aus dieser Schicht ausgewählt werden.

Annahme: Die Zahlen $L_1,\ldots,L_K$ und $\pi_1,\ldots,\pi_K$ seien bekannt.

In analoger Weise zur Definition des Anforderungsprofils, bezeichnet man $(L_1,\ldots,L_K)$, $(\pi_1,\ldots,\pi_K)$ als Risikoprofil des zu testenden Programms.
Sei ρ_h definiert durch

$$\rho_h = \frac{L_h \pi_h}{\sum_{i=1}^{K} L_i \pi_i}$$

ρ_h ist bekannt, da L_h und π_h als bekannt vorausgesetzt wurden. ρ_h heißt "Bedeutung der Schicht h", da es interpretiert werden kann als Anteil der Schicht h an den zu erwartenden Verlusten im Falle von Programmversagen.

Sei p_h die Wahrscheinlichkeit für Programmversagen in Schicht h, d.h. p_h = Wahrscheinlichkeit aus der Schicht h Eingabedaten gemäß dem Anforderungsprofil auszuwählen, die zum Versagen führen.

p_h ist konstant und wird nicht als bekannt angenommen.
Sei L der Verlust, der durch Programmversagen verursacht wird. L ist eine diskrete Zufallsgröße und kann nur die Werte $0, L_1, L_2, \ldots, L_K$ annehmen.

Das Risiko r des zu testenden Programms ist definirt als Erwartungswert EL von L, d.h.

$$r = EL$$

Es ist leicht zu zeigen, daß r dargestellt werden kann durch

$$r = EL = \sum_{h=1}^{K} L_h \pi_h p_h$$

Da die Wahrscheinlichkeiten $p_1,\ldots p_K$ unbekannt sind, muß das Risiko r als eine feste aber unbekannte Konstante betrachtet werden.

3. Das Testverfahren

Sei H eine über der Menge {1,...,K} definierte diskrete Zufallsgröße mit $p(H=h)=\rho_h$ (h=1,...K);d.h. H ist die Zufallsgröße, die die Schicht h mit der Wahrscheinlichkeit ρ_h auswählt.

Der Test wird in zwei Schritten durchgeführt:

1) Eine Schicht wird gemäß der Verteilung von H zufällig ausgewählt (Zufallsauswahl nach der Bedeutung der Schichten)
2) Ein Testfall aus der zuvor ausgewählten Schicht wird ausgeführt. Die Inputdaten für diesen Testfall werden gemäß der Verteilung im Betrieb aus dieser Schicht entnommen.

Die beiden Schritte werden n mal unabhängig voneinander wiederholt.

Als Ergebnis des Testverfahrens werden die beiden Zahlen
n: Anzahl sämtlicher durchgeführter Programmläufe
k: Anzahl derjenigen Programmläufe, die zum Versagen geführt haben
bestimmt.

4. Konfidenzaussagen für das Risiko

Nach der Durchführung dieses Testverfahrens lassen sich einige nützliche Konfidenzaussagen für das Risiko r formulieren.

1) Die obere Konfidenzgrenze $\hat{r}_{0.95}$ für das Risiko r zum Konfidenzniveau von (z.B. 95%) lautet:

$$\hat{r}_{0.95} = \left(\sum_{h=1}^{K} L_h \pi_h\right) \frac{(k+1)\, F_{0.95}(2(k+1), 2(n-k))}{(n-k) + (k+1)\, F_{0.95}(2(k+1), 2(n-k))}$$

wobei
n = Anzahl der durchgeführten Testläufe
k = Anzahl der Testläufe die zum Versagen geführt haben
$F_{0.95}(\ldots,\ldots)$ = 95%-Quantil der F-Verteilung mit (2(k+1), 2(n-k)) Freiheitsgraden.

2) Im wichtigen Spezialfall k=0, d.h. wenn kein einziger der Testläufe zum Programmversagen geführt hat, lautet die obere 95%-Konfidenzgrenze für das Risiko

$$\hat{r}_{0.95} = \left(\sum_{h=1}^{K} L_h \pi_h\right)\left(1 - \sqrt[n]{0.05}\right)$$

$$\approx \left(\sum_{h=1}^{K} L_h \pi_h \right) \frac{3}{n}$$

für große n, z.B. n≧100.

3) Wenn a priori anzunehmen ist, daß während des gesamten Tests kein Versagen beobachtet werden wird, d.h. k=0 sein wird, kann man die erforderliche Testanzahl n so angeben, daß eine vorgegebene 95%-Konfidenzgrenze $\tilde{r}$ eingehalten wird.

$$n = \frac{\ln 0.05}{\ln\left(1 - \frac{\tilde{r}}{\sum_{h=1}^{K} L_h \pi_h}\right)}$$

$$\approx \frac{3}{\tilde{r}} \sum_{h=1}^{K} L_h \pi_h \qquad \text{falls} \quad \tilde{r} \ll \sum_{h=1}^{K} L_h \pi_h$$

5. Beweis der Konfidenzaussagen

ad 1):
Sei H_1, H_2,... eine Folge von unabhängigen und wie H verteilten Zufallsgrößen. H_i bezeichnet die Nummer der Schicht, die für den i-ten Testlauf zufällig ausgewählt wird; $p(H_i=h)=\rho_h$. Für eine vorgegebene Schicht h sei $X^{(h)}$ die Zufallsgröße, die das Ergebnis eines Tests mit Inputdaten aus dieser Schicht bezeichnet, d.h.

$$X^{(h)} = \begin{cases} 1 & \text{falls Programmversagen eintrat} \\ 0 & \text{sonst} \end{cases}$$

Da die Eingabedaten gemäß dem Anforderungsprofil ausgespielt werden, gilt:

$$p(X^{(h)} = 1) = p_h$$

Sei Y_i die Zufallsgröße, die das Testergebnis des i-ten Testfalls bezeichnet, d.h.

$$Y_i = \begin{cases} 1 & \text{falls im Testfall Nr i Versagen eintritt} \\ 0 & \text{sonst} \end{cases}$$

Da für den Testfall Nr i die Schicht von der Zufallsgröße H_i festgelegt wird, gilt die Darstellung:

$$Y_i = X^{(H_i)} \qquad i = 1,\ldots,n$$

Die Wahrscheinlichkeit für Programmversagen beim i-ten Test lautet:

$$p(Y_i = 1) = p(X^{(H_i)} = 1)$$

$$= \sum_{h=1}^{K} p\left(X^{(H_i)} = 1 \,/\, H_i = h\right) p(H_i = h)$$

$$= \sum_{h=1}^{K} p(X^{(h)} = 1)\, p(H = h)$$

$$= \sum_{h=1}^{K} p_h \rho_h$$

Die Anzahl Y aller fehlerhaften Programmläufe ist eine Zufallsgröße und wird dargestellt durch

$$Y = \sum_{i=1}^{n} Y_i$$

Da $Y_1,\ldots,Y_n$ stochastisch unabhängig sind, hat Y eine Binomialverteilung mit den Parametern n und

$$p = \sum_{h=1}^{K} p_h \rho_h \qquad \text{[3,S.167], d.h.}$$

$$Y \sim Bin\left(n;\ p = \sum_{h=1}^{K} p_h \rho_h\right)$$

Nach der Durchführung des Tests und der Beobachtung von Y (z.B. Y=k die Anzahl der Versagensfälle), kann man die 95%-Konfidenzgrenze $p_{0.95}$ für p nach der Formel

$$\hat{p}_{0.95} = \frac{(k+1)\, F_{0.95}(2(k+1), 2(n-k))}{(n-k) + (k+1)\, F_{0.95}(2(k+1), 2(n-k))}$$

angeben [3,S.224].
Da p durch r wie folgt ausgedrückt werden kann

$$p = \sum_{h=1}^{K} \rho_h p_h = \sum_{h=1}^{K} \frac{L_h \pi_h}{\sum_{i=1}^{K} L_i \pi_i} p_h =$$

$$= \frac{1}{\sum_{i=1}^{K} L_i \pi_i} \cdot r$$

erhält man die gewünschte Konfidenzgrenze für r:

$$\hat{r}_{0.95} = \left(\sum_{h=1}^{K} L_h \pi_h\right) \hat{p}_{0.95}$$

$$= \left(\sum_{h=1}^{K} L_h \pi_h\right) \frac{(k+1)\, F_{0.95}(\ldots,\ldots)}{(n-k) + (k+1)\, F_{0.95}(\ldots,\ldots)} \quad .$$

ad 2):
Falls k=0, so gilt: $\hat{p}_{0.95} = \dfrac{F_{0.95}(2,2n)}{n + F_{0.95}(2,2n)}$

Nach [3,S.225] gilt außerdem:

$$X \sim F(2,2n) \;\Rightarrow\; Z := \frac{X}{n+X} \sim \text{Beta}(1,n) \text{, d.h.:}$$

hat die Zufallsgröße X eine F-Verteilung mit (2,2 n) Freiheitsgraden, so hat $Z = \frac{X}{n+X}$ eine Beta-Verteilung mit den Parametern (1,n).

Da Z eine monoton wachsende Funktion in X ist (über dem Intervall $(0,\infty)$), berechnet sich das 95%-Quantil $z_{0.95}$ von Z aus $z_{0.95} = \frac{x_{0.95}}{n+x_{0.95}}$, wobei $x_{0.95}$ das 95%-Quantil von X ist.

Da die Verteilungsfunktion von Z durch

$$F_Z(z) = 1-(1-z)^n$$

gegeben ist, errechnet sich das 95%-Quantl $z_{0.95}$ von Z durch Auflösung der Gleichung

$$1-(1-z_{0.95})^n = 0.95$$

Daraus folgt:

$$z_{0.95} = 1 - \sqrt[n]{0.05}$$

Damit ist

$$\hat{r}_{0.95} = \left(\sum_{h=1}^{K} L_h \pi_h\right) \hat{p}_{0.95}$$

$$= \left(\sum_{h=1}^{K} L_h \pi_h\right) z_{0.95}$$

$$= \left(\sum_{h=1}^{K} L_h \pi_h\right) \left(1 - \sqrt[n]{0.05}\right)$$

Da $e^{-3} \approx 0.05$:

$$\sqrt[n]{0.05} = \sqrt[n]{e^{-3}} = e^{-\frac{3}{n}} \approx 1 - \frac{3}{n} \quad \text{für große } n$$

$$\Rightarrow\; 1 - \sqrt[n]{0.05} \approx \frac{3}{n} \quad .$$

ad 3):
Die gesuchte Anzahl n der benötigten Programmläufe bekommt man durch Auflösung der Gleichung

$$\tilde{r} = \left(\sum_{h=1}^{K} L_h \pi_h\right) \left(1 - \sqrt[n]{0.05}\right)$$

nach n und Anwendung der Näherungsformel

$\ln(1-x) \approx x$ für kleine Werte von x, sowie über das Ergebnis von 2).

6. Abschließende Anmerkungen

1. Setzt man $L_1 = L_2 = \ldots = L_K = 1$, so erhält man das geschichtete Testverfahren nach der Versagenswahrscheinlichkeit (partition testing).
2. Setzt man K=1 und $L_1 = 1$ so erhält man das gewöhnliche Testverfahren (black box testing) nach der Versagenswahrscheinlichkeit.
3. Die Anzahl N_h der Testfälle für jede Schicht h=1,...,K ist eine Zufallsgröße. Offensichtlich ist die gemeinsame Verteilung von $N_1,\ldots,N_K$ eine Polynomialverteilung mit den Parametern $n, \rho_1,\ldots,\rho_K$.
   Die zu erwartende Anzahl $E(N_h)$ der Testfälle für die Schicht h lautet deshalb

$$E(N_h) = \rho_h n = \frac{L_h \pi_h}{\sum_{i=1}^{K} L_i \pi_i}\, n$$

4. Die hier formulierten Konfidenzaussagen über das Software-Risiko machen nur von den Ergebnissen der statistischen Testfälle Gebrauch. Eventuell zuvor durchgeführte deterministische Testfälle oder Verifikationsversuche müssen unberücksichtigt bleiben. Es erscheint daher sinnvoll, Erfahrungen und Ergebnisse aus früheren Testphasen zu quantifizieren und in die statistischen Aussagen einzubeziehen. Für das vorliegende Testverfahren kann dies geschehen durch Anwendung des Bayes'schen Satzes mit einer zuvor geeignet gewählten a priori Verteilung vom Beta-Typ [4,S.175].

Schrifttum

[1] G.J. Schick, R.W. Wolverton: "An Analysis of Competing Software Reliability Models", IEEE Trans. Software Eng., Vol. SE-4, March 1978

[2] G. Rzewski: "Recent Advances in Software Reliability Methods" Third National Reliability Conference 1981, Birmingham 1981

[3] Heinhold, Gaede: "Ingenieur-Statistik" Oldenbourg-Verlag 1968

[4] Kurt Stange: "Bayes-Verfahren", Springer Verlag 1977

Anschrift des Verfassers:

Dipl.-Math. Bernard Krzykacz
Gesellschaft für Reaktorsicherheit (GRS) mbH
Forschungsinstitute

8046 Garching bei München

Tel.: (089) 32004-196