

Spolehlivost softwaru

Radek Mařík

ČVUT FEL, K13133

December 12, 2010



1 Softwarové metriky

- Definice
- Metriky kvality produktu
- Metriky kvality běhu procesu
- Metriky kvality údržby

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Obsah

1 Softwarové metriky

- Definice
- Metriky kvality produktu
- Metriky kvality běhu procesu
- Metriky kvality údržby

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Metriky kvality softwaru ^[Kan95]

Metriky produktu

popisují charakteristiky produktu jako je velikost, komplexity, návrhové vlastnosti, výkonnost, úroveň kvality, apod.

Procesní metriky

mohou být využity při zlepšování vývoje softwaru a procesu údržby.

- efektivita odstraňování defektů během vývoje,
- vzor přírůstků defektů během testování,
- doba odezvy procesu oprav.

Metriky projektu

popisují charakteristiky projektu jeho provedení.

- počet vývojařů softwaru,
- vývoj personálu během životního cyklu softwaru,
- cena, rozvrh, produktivita.

Metriky kvality softwaru ^[Kan95]

Metriky produktu

popisují charakteristiky produktu jako je velikost, komplexity, návrhové vlastnosti, výkonnost, úroveň kvality, apod.

Procesní metriky

mohou být využity při zlepšování vývoje softwaru a procesu údržby.

- efektivita odstraňování defektů během vývoje,
- vzor přírůstků defektů během testování,
- doba odezvy procesu oprav.

Metriky projektu

popisují charakteristiky projektu jeho provedení.

- počet vývojařů softwaru,
- vývoj personálu během životního cyklu softwaru,
- cena, rozvrh, produktivita.

Metriky kvality softwaru ^[Kan95]

Metriky produktu

popisují charakteristiky produktu jako je velikost, komplexity, návrhové vlastnosti, výkonnost, úroveň kvality, apod.

Procesní metriky

mohou být využity při zlepšování vývoje softwaru a procesu údržby.

- efektivita odstraňování defektů během vývoje,
- vzor přírůstků defektů během testování,
- doba odezvy procesu oprav.

Metriky projektu

popisují charakteristiky projektu jeho provedení.

- počet vývojařů softwaru,
- vývoj personálu během životního cyklu softwaru,
- cena, rozvrh, produktivita.

Metriky kvality softwaru ^[Kan95]

- jsou podmnožinou softwarových metrik.
- zaměřují se na aspekty kvality produktu, procesu a projektu.
 - metriky finálního produktu,
 - metriky průběhu procesu.



Obsah

1 Softwarové metriky

- Definice
- **Metriky kvality produktu**
- Metriky kvality běhu procesu
- Metriky kvality údržby

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Metriky kvality produktu - orientované na zákazníka ^[Kan95]

Střední doba k selhání(MTTF - mean time to failure)

- často se používá v bezpečnostně kritických systémech jakou jsou systémy řízení letového provozu, letecká elektrotechnika a zbraně.
- implementačně velmi drahé.

Intensita defektů

- Obecný koncept rychlosti defektů vychází z počtu defektů vzhledem k možnosti vzniku chyb v určitém časovém rámci.
- Ve jmenovateli vystupuje velikost softwaru:
 - KLOC (thousand lines of code) ... tisíce řádek kódu,
 - SSI (shipped source instructions) ... předané zdrojové instrukce,
 - CSI (changed source instructions) ... změněné zdrojové instrukce,
 - problémy s vlastním počítáním
 - Počítej pouze proveditelné řádky.
 - Počítej proveditelné řádky a definice dat.
 - Počítej proveditelné řádky, definice dat a komentáře.
 - Počítej proveditelné řádky, definice dat, komentáře a příkazy řízení dávek.

Metriky kvality produktu - orientované na zákazníka ^[Kan95]

Střední doba k selhání(MTTF - mean time to failure)

- často se používá v bezpečnostně kritických systémech jakou jsou systémy řízení letového provozu, letecká elektrotechnika a zbraně.
- implementačně velmi drahé.

Intensita defektů

- Obecný koncept rychlosti defektů vychází z počtu defektů vzhledem k možnosti vzniku chyb v určitém časovém rámci.
- Ve jmenovateli vystupuje velikost softwaru:
 - KLOC (thousand lines of code) ... tisíce řádek kódu,
 - SSI (shipped source instructions) ... předané zdrojové instrukce,
 - CSI (changed source instructions) ... změněné zdrojové instrukce,
 - problémy s vlastním počítáním
 - Počítej pouze proveditelné řádky.
 - Počítej proveditelné řádky a definice dat.
 - Počítej proveditelné řádky, definice dat a komentáře.
 - Počítej proveditelné řádky, definice dat, komentáře a příkazy řízení dávek.

Metriky kvality produktu z pohledu zákazníka ^[Kan95]

- **Problémy hlášené zákazníkem** měří potíže zákazníka používajícího produkt.
 - **Metrika problémů** se obvykle vyjadřuje pomocí problémů vztažených na uživatele a měsíc (PUM - per user month).

$$PUM = \frac{\text{Celkový počet problémů, které ohlásili zákazníci} \\ \text{(skutečné defekty a potíže nevztahující se k defektům)} \\ \text{za danou časovou jednotku}}{\text{Celkový počet licencovaných měsíců pro daný software} \\ \text{za danou časovou jednotku}}$$

- Měl by se rovněž monitorovat celkový počet problémů zákazníka.
- **Spokojenost zákazníka** se často měří pomocí průzkumů použitím pětibodové škály:
 - Velmi spokojen
 - Spokojen
 - Neutrální
 - Nespokojen
 - Velmi nespokojen



Obsah

1 Softwarové metriky

- Definice
- Metriky kvality produktu
- **Metriky kvality běhu procesu**
- Metriky kvality údržby

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Metriky kvality průběhu procesu ^[Kan95]

- Metriky kvality průběhu procesu
 - jsou méně formálně definované.
 - jednoduše znamenají zaznamenávání přírůstků defektů během formálního testování.
- **Hustota defektů během formálního testování**
 - vyšší rychlosti defektů nalezených během testování indikují vyšší hladinu injektování chyb do softwaru během jeho vývoje.
 - defekty na KLOC
- **Vzor přírůstků defektů během formálního testování**
 - může indikovat, že testování začalo pozdě, že testovací sada nebyla dostatečná, nebo že testování skončilo předčasně.
- **Profil odstraňování defektů podle fází** vyžaduje trasování defektů ve všech fázích vývojového cyklu.
- **Efektivnost odstraňování defektů**

$$DRE = \frac{\text{Defekty odstraněné během vývojové fáze}}{\text{Defekty setrvávající v produktu}} \times 100\%$$



Obsah

1 Softwarové metriky

- Definice
- Metriky kvality produktu
- Metriky kvality běhu procesu
- **Metriky kvality údržby**

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Metriky kvality údržby I ^[Kan95]

- **Oprava nevyřízených věcí** vyjadřuje nároky na pracovní zátěž vázanou na údržbu softwaru.
 - Počet ohlášených problémů, které zůstávají otevřené na konci každého měsíce či týdne.
- **Řídicí index nevyřízených věcí (BMI - backlog management index)**

$$BMI = \frac{\text{Počet problémů uzavřených problémů během daného měsíce}}{\text{Počet přírůstků problémů během měsíce}} \times 100\%$$

- **Čas odezvy opravy** se obvykle počítá jak pro všechny problémy tak i pro problémy rozdělené podle úrovně vážnosti:

střední doba života problému od otevření po uzavření



Metriky kvality údržby II ^[Kan95]

- **Procento delikventních oprav:**

- Jestliže doba opravy překročí časové limity vzhledem k vážnosti, je klasifikována jako delikvent.

$$\text{Procento delikventních oprav} = \frac{\text{Počet oprav, které překročily časový limit vzhledem k vážnosti}}{\text{Celkový počet oprav odvedených ve specifikovaném čase}} \times 100\%$$

- Metrika **vadné opravy, kvalita opravy** měří procento vadných oprav z celkového počtu za nějaký časový interval.



Obsah

- 1 Softwarové metriky
 - Definice
 - Metriky kvality produktu
 - Metriky kvality běhu procesu
 - Metriky kvality údržby

- 2 Spolehlivost
 - Definice
 - Statické modely
 - Dynamické modely

Spolehlivost softwaru ^[Kan95]

- **Spolehlivost** je často definována jako pravděpodobnost, že systém vozidlo, stroj, zařízení, atd. bude vykonávat svou zamýšlenou funkci v daných operačních podmínkách po specifikovou dobu.
- **Modely spolehlivosti softwaru** se používají k odhadu spolehlivosti nebo počtu zbývajících defektů softwarového produktu, který byl uvolněn mezi zákazníky.
- Důvody:
 - 1 objektivní vyjádření kvality produktu,
 - 2 plánování zdrojů pro fázi údržby softwaru.
- **Sledovanou proměnnou** studovaných kritérií je počet defektů (nebo rychlost defektů normalizovaná počtem řádku kódu) za daný časový interval (týdny, měsíce, atd.), nebo doba mezi dvěma selháními.



Spolehlivost softwaru ^[Kan95]

- **Statický model** používá atributy projektu nebo programových modulů k odhadu počtu defektů v softwaru.
 - Parametry modelů jsou odhadovány na základě řady předchozích projektů.
- **Dynamický model** používá průběžného vývoje vzorů defektů k odhadu spolehlivosti finálního produktu.
 - Parametry dynamických modelů jsou odhadovány na základě mnoha údajů zaznamenaných o hodnoceném produktu k danému datu.
 - Kategorie:
 - Modeluje se celý vývojový proces. Model vychází s Rayleighova modelu.
 - Modeluje se fáze formálního testování. Model vychází z exponenciálního modelu a jiných modelů růstu spolehlivosti.



Obsah

- 1 Softwarové metriky
 - Definice
 - Metriky kvality produktu
 - Metriky kvality běhu procesu
 - Metriky kvality údržby

- 2 Spolehlivost
 - Definice
 - **Statické modely**
 - Dynamické modely

Weibullova distribuce ^[Kan95]

- jedna ze tří známých distribucí extrémálních hodnot,
- konce history pravděpodobnosti se blíží asymptoticky k nule, ale nikdy ji nedosáhnou.
- kumulativní distribuční funkce (CDF):

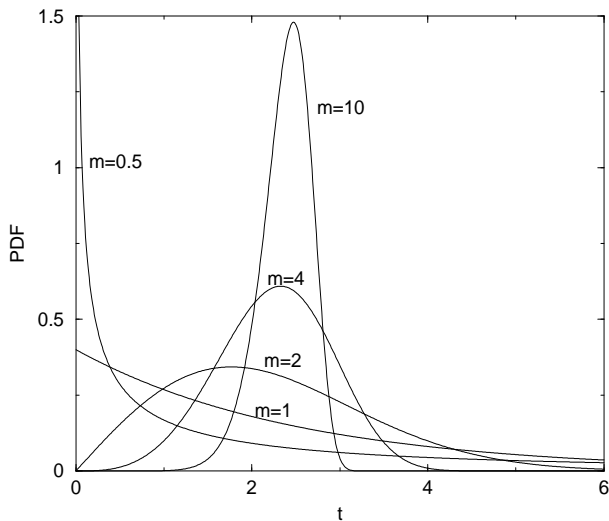
$$F(t) = 1 - e^{-(t/c)^m}$$

- funkce hustoty pravděpodobnosti (PDF):

$$f(t) = \frac{m}{t} \left(\frac{t}{c}\right)^m e^{-(t/c)^m}$$

- kde
 - m je tvarový parametr,
 - c je parametr měřítka,
 - t je čas.
- Pokud se aplikuje v softwaru, pak PDF typicky znamená hustotu defektů (rychlosti) v době přírůstkového vzoru defektů (platné defekty) a CDF znamená kumulativní vzor přírůstku defektů.

Tvar Weibullový distribuce



Rayleighův model ^[Kan95]

- patří do rodiny Weibullových distribucí.
- $m = 2$
- kumulativní distribuční funkce (CDF):

$$F(t) = 1 - e^{-(t/c)^2}$$

- pravděpodobnost hustoty (PDF):

$$f(t) = \frac{2}{t} \left(\frac{t}{c}\right)^2 e^{-(t/c)^2}$$

- t_m je okamžik, ve kterém křivka dosahuje svého maxima.

$$t_m = \frac{c}{\sqrt{2}}$$

- Jakmile je odhadnut t_m , může být určen tvar celé křivky. Plocha pod křivkou do t_m je 39.35% celkové plochy.



Rayleighův model v praxi ^[Kan95]

- Při praktických aplikacích se vzorec násobí konstantou K (K je celkový počet defektů nebo celková kumulativní rychlost defektů).

$$c = t_m \sqrt{2}$$

$$F(t) = K \left[1 - e^{-(1/2t_m^2)t^2} \right]$$

$$f(t) = K \left[\left(\frac{1}{t_m} \right)^2 t e^{-(1/2t_m^2)t^2} \right]$$

- Softwarové projekty sledují vzor životního cyklu popsany křivkou Rayleighovy hustoty.
- Vzorec odstraňování závad také sleduje Rayleighův vzorec.
- Celkový skutečný počet defektů se liší do 5% až 10% od počtu defektů predikovaný modelem.

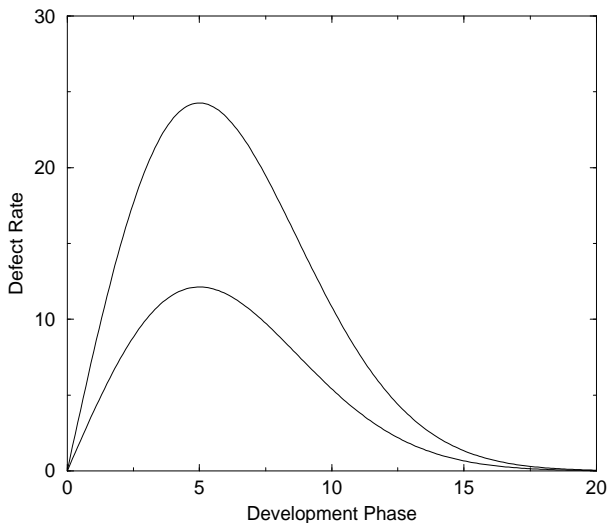


Základní předpoklady ^[Kan95]

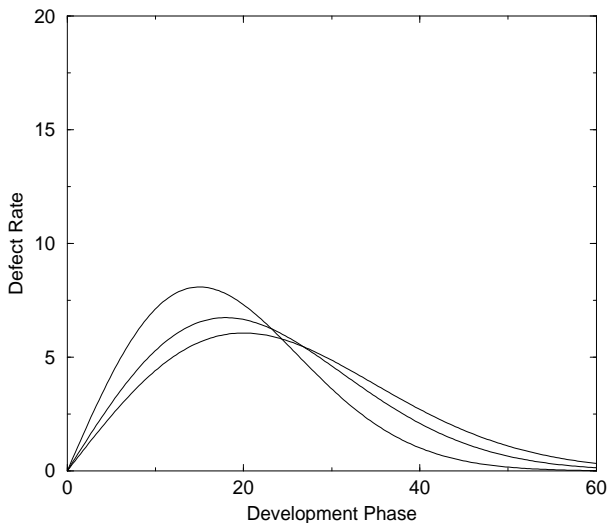
- *Rychlost defektů pozorovaných během vývojového procesu je pozitivně korelovaná s rychlostí defektů v poli nasazení*
- *Za předpokladu stejné rychlosti injektáže chyb, čím více defektů je objeveno a odstraněno dříve, tím méně jich zůstane na pozdější fáze.*
- Princip **“Udělej to správně hned napoprvé”**: jestliže každý krok vývojového procesu se provede s minimálním vznikem chyb, pak finální produkt bude dobrý.
Rovněž znamená, že vzniklé chyby se mají odstraňovat co nejdříve.



Základní předpoklady v grafech: korelace ^[Kan95]



Základní předpoklady v grafech: odstraňování defektů [Kan95]



Spolehlivost a validace predikce ^[Kan95]

- **Spolehlivost** vyjadřuje stupeň změny výstupu modelu vzhledem možnostem fluktuací ve vstupních datech.
- Čím užší je konfidenční interval, tím je odhad spolehlivější.
- Větší vzorky vedou na užší konfidenční intervaly.
- Používejte pokud možno více modelů a spoléhejte se na jejich společné hodnocení.
- Základní podmínkou dosažení **platnosti predikce** je zajištění přesnosti a spolehlivosti vstupních dat.
- Platnost se hodnotí porovnáním odhadů z modelů a jejich skutečných hodnot.



Obsah

1 Softwarové metriky

- Definice
- Metriky kvality produktu
- Metriky kvality běhu procesu
- Metriky kvality údržby

2 Spolehlivost

- Definice
- Statické modely
- Dynamické modely

Model založený na exponenciálním rozložení a růstu spolehlivosti ^[Kan95]

- Modely růstu spolehlivosti se obvykle odvozují z dat fáze formálního testování.
- Odůvodnění vychází z toho, že vzory procesu přírůstků defektů této fáze jsou vhodným indikátorem spolehlivosti produktu pociťovanou zákazníky.
- Během tohoto testování po fázi vývoje, kdy se objevují selhání, identifikují a opravují defekty, se softwarový produkt stává stabilnější a jeho spolehlivost roste s časem. Tyto modely se proto nazývají **modely růstu spolehlivosti**.



Exponenciální model ^[Kan95]

- další speciální případ Weibullový rodiny, $m = 1$
- kumulativní distribuční funkce (CDF):

$$F(t) = 1 - e^{-(t/c)} = 1 - e^{-\lambda t}$$

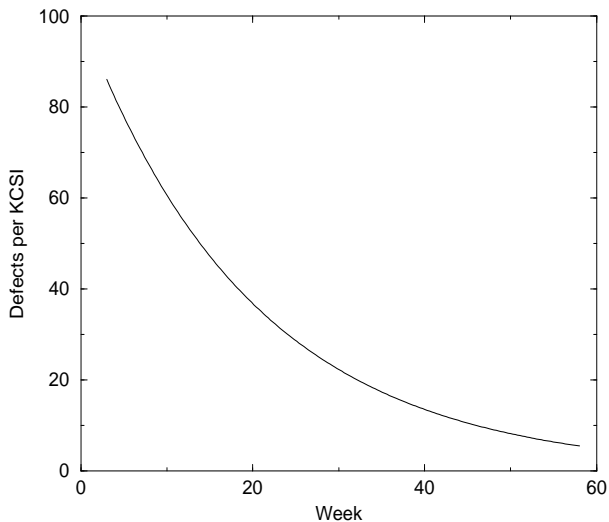
- hustota pravděpodobnosti (PDF):

$$f(t) = \frac{1}{c} e^{-(t/c)} = \lambda e^{-\lambda t}$$

- kde
 - c je parametr měřítka,
 - t je čas,
 - $\lambda = 1/c$
- λ se nazývá **rychlost detekce chyby** nebo **okamžitá rychlost selhání** (ve statistice také *rychlost hazardu*).
- V praktických aplikacích se vzorce přenásobují celkovým počtem defektů nebo celkovou kumulativní rychlost defektů K .



Exponenciální model - distribuce hustoty



Modely růstu spolehlivosti ^[Kan95]

- Ne mnoho modelů je ověřeno praktickým nasazením s reálnými daty.
- **Model doby mezi selháním:**
 - Očekává se, že doby následných selhání se prodlužují po každém odstranění defektu produktu.
 - Často se předpokládá, že se doba mezi selháním ($i - 1$) a i řídí rozložením, jehož parametry mají vztah k počtu skrytých defektů setrvávajících v produktu po ($i - 1$) selhání.
- **Modely počtu vad:**
 - počet vad či selhání (nebo normalizované rychlosti) ve specifikovaném časovém intervalu.
 - Časový interval je pevný *a priori*.
 - Počet defektů nebo selhání pozorovaných během intervalu se považuje za náhodnou proměnnou.
 - Očekává se, že se počet pozorovaných selhání za jednotku času bude zmenšovat.



Model Jelinski-Moranda (J-M) ^[Kan95]

- jeden z prvních modelů softwarové spolehlivosti (1972),
- model doby mezi selháním,
- Předpoklady:
 - Software má N nedostatků na počátku testování.
 - Selhání se projeví čistě náhodně.
 - Všechny vady přispívají rovnocenně k příčině selhání během testování.
 - Čas opravy je zanedbatelný.
 - Oprava defektu je dokonalá.
- Hazardní funkce v čase t_i , doba mezi selháním $(i - 1)$ a i , je dána:

$$Z(t_i) = \phi[N - (i - 1)]$$

- kde
 - ϕ je konstanta úměrnosti.
- Hazardní funkce je konstantou mezi dvěma selháními, ale zmenšuje se po krocích ϕ po každém odstranění defektu.
- Jak se jednotlivé vady odstraňují, doba k následnému selhání se očekává být delší.

Modely Littlewooda (LW) ^[Kan95]

- podobné modelu J-M,
- Předpokládá se, že různé vady mají různou velikost a nerovnost příspěvků k selháním.
- Vady většího rozsahu mají tendenci být detekovány a opraveny dříve.
- Uvažováním velikosti chyby se předpoklady modelu přibližují realitě.
- Ve podmínkách reálného softwaru, předpoklad rovnocenné rychlosti selhání pro všechny vady běžně vůbec nenastává.



Goel-Okumotův (G-O) model nedokonalého opravování ^[Kan95]

- J-M vychází z dokonalé opravy
 - zanedbatelný čas,
 - vada skutečně opravena
- .
- Procento vadných oprav se pro velké komerční softwarové společnosti pohybuje okolo 1% až 2%, může však přesáhnout i 10%.
- Předpoklad nedokonalého opravování.
- Hazardní funkce v čase t_i , době mezi selháním $(i - 1)$ a i , je dána:

$$Z(t_i) = [N - p(i - 1)]\lambda$$

- kde
 - N je počet vad na počátku testování,
 - p je pravděpodobnost nedokonalé opravy,
 - λ je rychlost selhání na vadu.



Goel-Okumotův model nehomogenního Poissonova procesu (NHPP) ^[Kan95]

- modeluje počet selhání pozorovaný v daných intervalech testování
- předpoklady:
 - Kumulativní počet selhání pozorovaný v čase t , $N(t)$, se modeluje jako nehomogenní Poissonův proces - Poissonův proces s časově závislou rychlostí selhání.
 - Časově závislá rychlost selhání sleduje exponenciální distribuci.
- Model je dán

$$P\{N(t) = y\} = \frac{[m(t)]^y}{y!} e^{-m(t)}, \quad y = 0, 1, 2, \dots$$

- kde

$$\begin{aligned} m(t) &= a(1 - e^{-bt}) \\ \lambda(t) &\equiv m'(t) = abc^{-bt} \end{aligned}$$



NHPP model ^[Kan95]

- $m(t)$ je očekávaný počet selhání v čase t ,
- $\lambda(t)$ je hustota pravděpodobnosti,
- a je očekávaný počet selhání, který by mohl být pozorován,
- b je rychlost detekce vad vztažená na vadu.
- Počet vad a , které se mají detekovat, se považuje za náhodnou proměnnou, jejíž pozorovaná hodnota závisí na testu a dalších faktorech prostředí. Tím se tento model principiálně liší od ostatních, které předpokládají fixní neznámý počet vad.
- Vyskytuje se řada případů, kdy rychlost selhání nejprve roste a poté klesá (Goelův model zobecněného nehomogenního Poissonova procesu):

$$\begin{aligned}m(t) &= a(1 - e^{-bt^c}) \\ \lambda(t) &\equiv m'(t) = abc^{-bt^c} t^{c-1}\end{aligned}$$

- b a c jsou konstanty, které odrážejí kvalitu testování.



Musa-Okumotův (M-O) model logaritmického Poissonova prováděcího času ^[Kan95]

- Předpoklad, že se pozorovaný počet selhání v daném čase, τ , řídí nehomogenním Poissonovým procesem.
- Funkce střední hodnoty se snaží postihnout to, že pozdější opravy mají menší efekt na spolehlivost softwaru než počáteční opravy.
- Logaritmický Poissonův proces se velmi vhodný k popisu vysoce neuniformních operativních profilů uživatele, kdy některé funkce jsou prováděny mnohem častěji než jiné.
- Funkce střední hodnoty je dána vztahem

$$u(\tau) = \frac{1}{\theta} \ln(\lambda_0 \theta^\tau + 1)$$

- kde
 - λ_0 je počáteční hustota selhání,
 - θ je rychlost redukce normalizované intenzity selhání na jednotku selhání.



Zpožděný S model ^[Kan95]

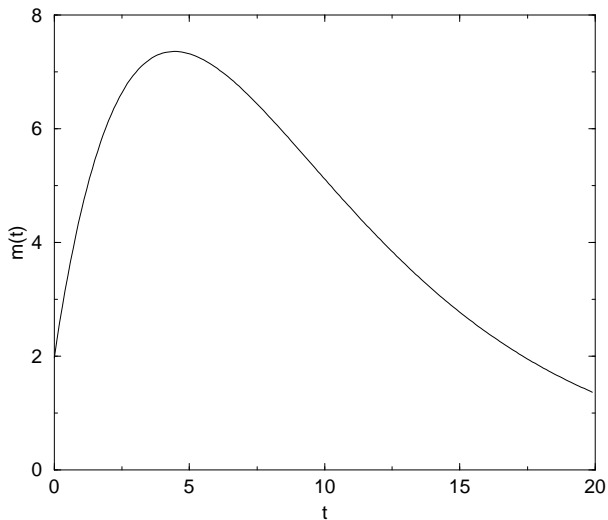
- Testovací proces není jenom o detekování chyb, ale i o lokalizaci defektů.
- Mezi časem první detekce selhání a časem nahlášení může být významné zpoždění, protože je potřeba analyzovat selhání.
- Model zpožděného rustů spolehlivosti S tvaru je založen na nehomogenním Poissonovém procesem s různou funkcí střední hodnoty, aby se popsalo zpoždění nahlášení selhání:

$$m(t) = K[1 - (1 + \lambda t)e^{-\lambda t}]$$

- kde
 - t je čas,
 - λ je rychlost detekce chyby,
 - K je celkový počet defektů nebo celková kumulativní rychlost defektů.



Funkce střední hodnoty S modelu se zpožděním



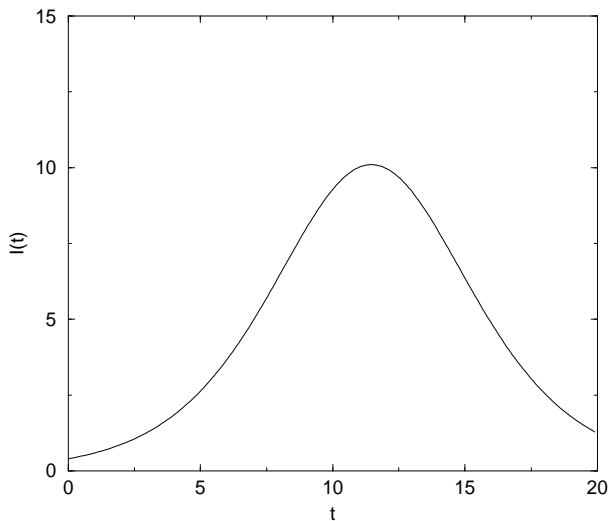
Inflexní S model ^[Kan95]

- další S model růstu spolehlivosti,
- Model popisuje jevy detekce selhání softwaru, jestliže detekované defekty jsou vzájemně závislé.
- Čím více defektů detekujeme, tím se více nedetekovaných selhání stává detekovatelných.
- významné zlepšení ohledně předpokladu nezávislosti vad programu.
- založeno na nehomogenním Poissonově procesu, model funkce střední hodnoty je

$$I(t) = K \frac{1 - e^{-\lambda t}}{1 - ie^{-\lambda t}}$$

- kde
 - t je čas,
 - λ je rychlost detekce chyb,
 - i je faktor inflexe,
 - K je celkový počet defektů nebo celková kumulativní rychlost defektů.

Inflexní S model - funkce střední hodnoty



Modely zpožděného S a inflexního S ^[Kan95]

- zahrnují dobu učení testerů, kdy se seznamují se softwarem na začátku doby testování.
- Doba učení se asociuje se vzory zpoždění nebo inflexí popisovaného funkcemi střední hodnoty.
- Srovnání:
 - Exponenciální model předpokládá vrchol přírůstků defektů na počátku fáze testování a poté klesá.
 - Model zpoždění S předpokládá lehce zpožděný vrchol.
 - Model inflexe S předpokládá pozdější a ostřejší vrchol.



Literatura I



Stephen H. Kan.

Metrics and Models in Software Quality Engineering.
Addison-Wesley, 1995.