

# Kto by mal niesť zodpovednosť za chyby v dodávanom softvéri?

MICHAL ŠTÍPEK

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava*

**Abstrakt.** V súčasnosti je zodpovednosť softvérových firiem za chyby v ich produktoch len veľmi obmedzená. Firmy sa zriekajú akejkolvek zodpovednosti, pričom sa obhajujú najmä argumentmi o rozdielnosti softvéru oproti bežným produktom, akými sú napríklad autá a argumentmi o nemožnosti vyvinúť softvér, ktorý by neobsahoval chyby. V eseji analyzujem súčasný stav, pričom sa na daný problém pozerám z oboch strán a zamýšľam sa nad jednotlivými argumentmi odporcov i zástancov zavedenia zodpovednosti za softvér. V práci som sa pokúsil odpovedať na otázku, akým smerom sa bude vývoj v danej oblasti uberať v budúcnosti a aký bude mať dopad na vývoj a predaj softvérových produktov.

## Úvod

V súčasnosti sa, z hľadiska zodpovednosti za kvalitu a bezpečnosť dodávaných produktov, pozerá na softvér inak ako na bežný tovar a služby, akými sú napríklad autá, domáce spotrebiče alebo stavebné práce. Ak by sa vám napríklad pri používaní auta stala nehoda, ktorá by bola dôsledkom preukázateľného zlyhania brzd, je výrobca automobilu povinný zaplatiť za vzniknuté škody. Navyše, okrem získania náhrady za priame škody, máte aj právo vymáhať škody, ktoré vám v dôsledku havárie vznikli nepriamo. A dokonca je tu vysoká pravdepodobnosť, že na súde uspejete. Prečo tomu tak nie je pri softvéri? Odpoveď na túto otázku je zložitá.

## Softvér ako „tovar“

Softvér ako „tovar“ má určite svoje špecifiká. Čo ho ale odlišuje od ostatných výrobkov? Mnoho ľudí sa obhajuje argumentmi o tom, že vytváranie softvéru je komplexný proces, na jeho vývoji sa často krát podieľa veľké množstvo ľudí a najmä tým, že vytvoriť softvér, ktorý by neobsahoval chyby, je z jeho podstaty priam nemožné. Musím uznať, že je v tom istý kus pravdy. Na druhej strane, podobné

*Manažment v softvérovom inžinierstve, máj 2005, s. 1-6.*

tvrdenia môžeme aplikovať aj na niektoré bežné produkty alebo služby a i napriek tomu pre ne platia iné pravidlá. Naopak, ak si niekto kupuje softvér, zaplatí zaň a autor má určitý zisk. Z tohto pohľadu môžeme tvrdiť, že sa jedná o klasickú kúpnopredajnú zmluvu a mali by pre ňu platiť rovnaké podmienky ako pri kúpe bežného tovaru. Tento pohľad je relatívne správny, pokiaľ sa jedná o platený softvér. Inak tomu zákonite musí byť, ak hovoríme o softvéri, ktorý je zadarmo alebo má otvorenú licenciu. V takomto prípade nie je uhradenie určitej sumy podmienkou jeho používania. Autori takýchto programov si určite položia otázku, prečo by mali niesť zodpovednosť za niečo, čo vytvorili zadarmo a ponúkli ostatným bez nároku na odmenu.

Softvérové firmy sa snažia obchádzať zákony, ktoré platia pre bežné produkty najmä tým, že namiesto samotného produktu predávajú len licenciu na jeho používanie. Určite ste sa už stretli s tým, že pri inštalácii nejakého programu ste boli nútený odklepnúť licenčné ujednanie. Existuje však veľmi málo ľudí, ktorí si toto ujednanie aj prečítali. Ak by ste tak urobili, dozvedeli by ste sa, že výrobca vám v podstate za nič neručí, nie je zodpovedný za žiadne priame, ani nepriame škody a vy môžete byť radi, že môžete daný softvér vôbec používať a nedajbože, aby ste ho niekomu skopírovali. Je však otázne, akú má váhu takéto „odklepnutie“ z právneho hľadiska. Nikde sme sa predsa nepodpísali a často krát predložený text nie je ani v slovenčine. Určite by bolo zaujímavé zistiť, či je takéto „súhlasenie“ s licenciou právne v poriadku, a či sa nejedná o konanie pod nátlakom.

## Chyby v softvéri

Ako som už spomínal, vytvoriť softvér bez chýb je takmer nemožné. Prečo tomu tak je? Softvér je výsledkom zložitého procesu, ktorý pozostáva z viacerých fáz. Každá fáza je sama o sebe komplexná a vstupuje do nej mnoho faktorov. Ďalší významný faktor je ten, že softvér vytvárajú ľudia a ľudia robia chyby. Ak vezmeme do úvahy fakt, že na vývoji softvéru sa takýchto ľudí podieľa viac a títo ľudia musia, okrem iného, medzi sebou efektívne komunikovať, je zrejme prečo softvér obsahuje tak veľké množstvo chýb. Posledný a najvýznamnejší faktor je ten, že až príliš málo času sa venuje samotnému testovaniu. Veľká väčšina projektov sa dokončuje v časovom sklze a s tým súvisí aj zvýšený stres programátorov. A ako vieme, človek pod stresom robí viac chýb.

Len tak pre zaujímavosť, podľa prieskumu, ktorý bol vykonaný pred viac ako dvomi rokmi[2], je medián chýb v softvéri 0.15 chýb na 1000 riadkov kódu. Na prvý pohľad sa nezdá, že sa jedná o veľké číslo, ale keď vezmeme do úvahy fakt, že v súčasnosti zložitejší softvér obsahuje aj niekoľko miliónov riadkov zdrojového kódu, môže počet chýb v programe narásť až na niekoľko tisíc. Zaujímavosťou pritom je, že najviac chybový softvér sa vytvára v Spojených štátoch Amerických (medián 0.4 na 1000 riadkov) a najmenej v Japonsku (medián 0.02 chýb na 1000 riadkov)

Pri klasifikovaní chýb treba rozlišovať či sa jedná o chyby, ktoré sa prejavujú pri jeho normálnom používaní a nie sú zapríčinené pôsobením tretej strany, alebo či sa jedná o bezpečnostné chyby. Typickým príkladom bežných chýb sú chyby vo výpočte

alebo chyby v algoritmoch, ktoré následkom neošetreného vstupu spôsobia pád programu. Je logické, že za takéto chyby by mal výrobca niesť plnú zodpovednosť, nakoľko sú dôsledkom jeho práce a nedokonalého testovania. I keď býva pravidlom, že chyby tohto typu, ktoré by spôsobili vážne škody, sa skoro vôbec nevyskytujú a väčšinou sa jedná len o maličkosti.

Na druhej strane môžeme hovoriť o bezpečnostných chybách. Bezpečnostné chyby sú chyby, ktoré sa prejavia až pôsobením tretej strany. Môže sa jednáť o pôsobenie škodlivého kódu, ktorým sú napríklad vírusy alebo červy alebo o pôsobenie človeka. I keď, škodlivý kód musel naprogramovať aj tak vždy len človek. Je teda otázne, či by mali softvérové firmy niesť zodpovednosť za škody spôsobené treťou stranou. Jediné v čom je jasno už teraz je to, že za škody spôsobené zneužitím bezpečnostných dier by mala niesť osoba, ktorá ich zneužíva. Takáto osoba využíva chyby kvôli vlastnému prospechu s cieľom obohatiť sa alebo poškodiť iné osoby. Otázne však ostáva do akej miery by mal za takéto chyby zodpovedať samotný výrobca. Ak by sme dali ten istý softvér naprogramovať dvom rôznym firmám, každý by obsahoval iné bezpečnostné chyby a iného charakteru. Kde sa nachádza ona pomyselná hranica, pri ktorej môžeme povedať, že daný softvér je dobre zabezpečený alebo že je deravý? Nie je chyba ako chyba. Niektoré chyby sú relatívne primitívne a dokáže ich zneužiť aj laický záškodník. Naopak, sú aj chyby, ktoré dokáže zneužiť len naozajstný profesionál a aj toho to stojí nemalé úsilie. Ak by sme zobrali náš prípad s autom, môžeme povedať, že niektoré autá sa samé otvoria už pri jemnom zatlačení na dvere a naopak, že na niektoré treba použiť krompáč. Podobná analógia platí aj pre softvér, i keď s tým rozdielom, že určenie danej hranice, kedy softvér považujeme za dostatočne zabezpečený, je takmer nemožné a nanajvýš diskutabilné.

### **Zavedenie zodpovednosti? Pre aj proti.**

O otázke zavedenia zodpovednosti za softvér sa v poslednej dobe vedú mnohé diskusie. Je prirodzené, že proti zavedeniu zodpovednosti sú najmä samotní výrobcovia, ktorí používajú najmä nasledujúce argumenty:

- nie je možné úplne eliminovať chyby
- spomalenie vývoja
- zvýšené náklady a s tým spojený krach niektorých spoločností
- predraženie softvéru
- nejasnosti s určovaním pôvodu chyby
- o kvalitu sa postará samotné trhové prostredie
- súčasná podpora je dostatočná

Naopak, medzi zástancov zavedenia zodpovednosti za chyby v softvéri patria najmä samotní zákazníci a najmä politici. Medzi hlavné argumenty zástancov patria:

- kvalitnejší a bezpečnejší softvér
- spoločnosť je čoraz viac závislá na používaní softvéru a preto treba donútiť producentov softvéru, aby vytvárali bezpečnejšie produkty
- pre softvér by mali platiť rovnaké pravidlá ako pre bežné produkty
- súčasná podpora je nedostatočná

Dodávatelia softvéru tvrdia, že ak budú nútení niesť finančnú zodpovednosť za chyby vo svojich produktoch, budú musieť tieto náklady premietnuť do celkovej ceny. Následkom toho sa dostupnosť, pre našinca už i tak dosť drahého softvéru, ešte viac zníži a vzrastie podiel softvérového pirátstva. To v konečnom dôsledku môže znamenať aj zníženie príjmov a s tým súvisiaci krach niektorých spoločností. Ďalší logický krok, ktorý firmy kvôli zníženiu nákladov súvisiacich s odškodňovaním podniknú bude zníženie počtu chýb. To znamená, že vo všetkých fázach softvérového procesu sa prijímú také opatrenia, ktoré povedú k zníženiu počtu chýb a k zvýšeniu kvality. Možno sa niekomu zdá, že to je presne to, čo sme chceli, ale na druhej strane s tým súvisí spomalenie, ale najmä ďalšie predraženie vývoja. I keď je vždy lepšie zaplatiť viac pri vývoji ako pri eventúálnych súdnych sporoch. A navyše, zákazník bude aj spokojnejší.

Naopak, hlavným cieľom zástancov je najmä skvalitnenie softvérových produktov. Podľa ich názorov to má väčší význam ako prípadné spomalenie vývoja. Tomu nahráva aj fakt, že súčasná moderná spoločnosť je čoraz viac závislá od používania softvéru. Softvér sa používa takmer v každom priemyselnom odvetví, od výroby v továrni až po riadenie elektrickej siete. Výrobcovia však tvrdia, že o to, že sa bude vyvíjať kvalitný softvér sa postará samotný trh. Ich argumenty spočívajú v tom, že softvér, ktorý nebude kvalitný a bude obsahovať veľa chýb, ľudia jednoducho prestanú kupovať a zaobstarajú si ho u konkurencie. To však nemusí byť úplne pravda. Realita je bohužiaľ často krát iná. Mnoho projektov mešká, firmy sú nútené používať produkty tretích strán a nemôžu len tak vymeniť softvér. Často krát teda nemajú príliš na výber. Navyše, veľa spoločností sa príliš neriadi dopytom trhu. V prípade generických produktov je vidieť snaha vydať čo najskôr novú verziu, pričom určite by bolo z hľadiska zákazníkov prospešnejšie keby sa čas venovaný vývoju novej verzie venoval odstráneniu a vylepšeniu tej predchádzajúcej. Typickým príkladom môže byť vydanie novej verzie Windows 2003. Zo strany trhu nebol po novej verzii takmer žiadny dopyt a i napriek tomu spoločnosť Microsoft vydala novú verziu namiesto toho, aby sa plne venovala odstráneniu chýb z tej starej. Podobne je to aj so zákazníckym softvérom. Vďaka tlakom zo strany konkurencie a v snahe získať zákazníka sa firmy uchýľujú k skresľovaniu ponúk a k uvádzaniu nepresných odhadov.

Ďalší závažný problém, ktorý by vznikol pri zavedení zodpovednosti, je presné určenie pôvodu chyby a dokazovanie, že škoda vznikla dôsledkom chyby v softvéri. Softvér môže bežať na rôznom vybavení, pričom niektoré chyby sa prejavajú len za

zložitých podmienok. Ďalší príklad je ten, kedy na počítači beží viac aplikácií súčasne, pričom každá o samote beží normálne, ale keď bežia spolu, vyskytnú sa problémy.

Zaujímavý rozpor medzi odporcami a zástancami zavedenia zodpovednosti nastal v otázke podpory. Súčasná podpora zo strany výrobcov softvéru spočíva v dodatočnom odstraňovaní nájdených chýb a vo vydávaní rôznych záplat a aktualizácií. Mnoho výrobcov tvrdí, že takýto stav je dostatočný. Kvôli objektívnosti však musím podotknúť, že takáto podpora je uskutočňovaná až vtedy, keď sa chyba prejaví, prípadne spôsobí nejakú škodu. To môže byť nežiaduce. Navyše neustále záplatovanie a aktualizácia môžu stáť spoločnosť obrovské množstvo času a i napriek tomu to nemusí byť dostatočné. To je aj hlavný argument, prečo mnoho zástancov zavedenia zodpovednosti tvrdí, že súčasná podpora naopak nie je vôbec dostačujúca.

Myslím si, že ak by niekedy v budúcnosti došlo k zavedeniu zodpovednosti za softvér, splnila by sa väčšina argumentov, ktoré používajú výrobcovia. Softvér by bol drahší, jeho vývoj by trval dlhší čas, ale na druhej strane by bol takýto softvér aj kvalitnejší a obsahoval by menej závažných chýb. I keď vytvoriť softvér bez chýb je asi nereálne, ale ich počet by sa podarilo minimalizovať na únosnejšiu mieru. Keď to ide v Japonsku, prečo nie aj inde?

## **Súčasný stav**

V poslednej dobe sa objavujú prvé prípady odškodňovania za chyby v softvéri [4]. Ako príklad môžem uviesť prípad, ktorý sa stal v Spojených Štátoch Amerických. V roku 1998 si jedna texaská firma vyrábajúca produkty pre psov zakúpila softvér, ktorý jej spôsobil značné škody. Firma nakoniec vysúdila odškodné presahujúce 2 milióny dolárov. Dôvod rozsudku znel, že dodaná technológia sa ukázala ako nestabilná a nedokončená. Problém spočíval v blokovaní servera pri vzrastajúcom počte používateľov. Zaujímavé na tom je, že chyba sa vyskytovala pri použití softvéru len s určitou verziou databázového servera a tento databázový server bol dokonca dodaný tretou stranou. Na druhej strane, treba podotknúť, že vyplatená suma nezodpovedala ani desatine požadovanej sumy.

Okrem iného badať zvýšený záujem o danú problematiku zo strany vlád a vládnych agentúr. Typickým príkladom sú opäť Spojené Štáty Americké, kde badať prvé hlasy politikov o zavedení aspoň určitej zodpovednosti výrobcov. Navyše, ako už bolo spomenuté vyššie, objavujú sa prvé žaloby. Tie sú v prvom rade namierené voči najväčšiemu svetovému výrobcovi softvéru, firme Microsoft. V týchto žalobách sa jedná najmä o bezpečnostné chyby, nakoľko niektoré z nich spôsobili v poslednom období závažné problémy s funkčnosťou a dostupnosťou napadnutých systémov. Ďalším faktom je aj to, že chyby v softvéri spôsobujú priemyslu značné škody. Podľa prieskumov z roku 2002[3], ktorý vykonal americký Národný Úrad pre Štandardy a Technológie (National Institute of Standard and Technology - NIST), vznikli priemyslu v dôsledku chýb v softvéri škody takmer 60 miliárd amerických dolárov. 2/3 z týchto škôd museli zaplatiť samotní používatelia. Tento prieskum bol vykonaný takmer pred tromi rokmi, takže môžeme smelo predpokladať, že v súčasnosti sú tieto

škody ďaleko vyššie. Aj to je jeden z dôvodov prečo je nutné sa danou problematikou zaoberať.

## Záver

Celkovo možno povedať, že sa minimálne otvorila diskusia o danej problematike a aj s ohľadom na čoraz vyšší záujem zo strany exekutív jednotlivých krajín a s ohľadom na narastajúce škody, dôjde v budúcnosti i k legislatívnemu ohraničeniu a k zavedeniu aspoň čiastočnej zodpovednosti za presne špecifikovaných podmienok. To či sa tak stane a či to bude správne rozhodnutie, ukáže až čas.

## Použitá literatúra

1. Cusumano, M.: Who is Liable for Bugs and Security Flaws in Software? *Communications of the ACM* (March 2004/vol. 47, No 3)
2. Cusumano, M. et al.: Software development worldwide: The state of the practice. *IEEE Software*, (Nov.-Dec. 2003).
3. Keefe, T.: Software insecurity. *Computerworld* (Aug. 5, 2002)
4. Sonini, M. J.D.: Edwards user wins arbitration case against ERP vendor. *Computerworld*, (Feb. 21, 2002)

## Annotation

*Who should be responsible for bugs in software?*

Nowadays, software companies hold limited or no responsibility for bugs and security flaws in their products. Companies resign their responsibility, using arguments about differences between software and common products, like cars and arguments about actual state, when it is almost impossible develop bug free software. In this work I analyze actual state and I look on problem from both sides. I think about arguments that are using both, the supporters and the objectors. In my work I try to answer questions about trends towards the future and their impact on developing and selling of software products.