

SYSTÉMOVÝ PŘÍSTUP K JAKOSTI SOFTWARE

Branislav Lacko

Fakulta strojního inženýrství VUT v Brně, Ústav automatizace a informatiky
lacko@uai.fme.vutbr.cz

Abstrakt

Příspěvek zdůrazňuje význam kvalita software a ukazuje, že systémový přístup k této problematice může přinést potřebné výsledky. Posuzuje současné problémy v oblasti kvality software z hlediska softwarového inženýrství, z hlediska ekonomického, z hlediska organizačního a personálního.

Klíčová slova: kvalita software, systémový přístup, problémy kvality software, softwarové inženýrství

1. Úvod

Cílem příspěvku je zdůraznit význam kvality software a ukázat, že jen systémový přístup k zajišťování jakostního software může přinést potřebné výsledky.

Do jisté míry je příspěvek pokusem o zodpovězení otázky: Co je příčinou přehlížení zásad tvorby kvalitního programového vybavení nejen v České republice, ale i v zahraničí?

Příspěvek zdůrazňuje problematiku jakosti software v souvislosti s využitím počítačů pro automatizaci řízení technologií a procesů, kdy chyby programů mohou ohrozit bezpečnost či dokonce životy mnoha lidí.

Příspěvek je zpracován v rámci Národního programu jakosti ČR pro rok 2004 jako příspěvek k celostátnímu a Evropskému úsilí o zvýšení jakosti výrobků a služeb.

Revize norem řady ISO 9000:2000 a vstup České republiky do Evropské unie, staví české firmy před problém, navrhnout a realizovat systémy řízení jakosti podle těchto revidovaných požadavků. Pro mnoho průmyslových firem to znamená často jen provedení adekvátních změn ve svých systémech řízení jakosti spolu s absolvováním recertifikačního auditu. Pro softwarové firmy však většinou soubor revidovaných norem představuje zcela nové požadavky, které si vyžádají odlišné přístupy k celé řadě každodenních činností.

2. Současné důvody pro kvalitu software

Existují tři důvody, proč je problematika kvality programů v současnosti tak aktuální:

- Software se stalo velmi rozšířeným zbožím a zákazníci by měli být chráněni před nekvalitními produkty
- Do software se investují obrovské finanční částky a z celospolečenského i tržního hlediska je potřeba zajistit jejich efektivní zhodnocení
- Stále více se používá software v aplikacích pro automatické řízení procesů, kde chyby v programovém vybavení mohou mít velké, často katastrofální důsledky (řízení jaderných reaktorů, řízení zdravotnických zařízení, navigace letadel, apod.).

Přesto je současný stav v oblasti kvality programového vybavení dosti neuspokojivý. V důsledku chyb v software ztrácí ekonomika USA ročně 59,9 miliardy dolarů, jak ukázala studie National Institute of Standards and Technology v roce 2002 [3]. Studie také konstatovala, že přestože v současné době při tvorbě software nelze všechny chyby odstranit, více než třetinu by bylo možno se vyhnout při dokonalejším systému zkoušek.

Firma Microsoft přiznala v polovině roku 2003, že kvůli softwarové chybě mohly být v ohrožení účty 200 milionů uživatelů, protože potenciální útočník se mohl dostat jak k e-mailovému účtu, tak k číslům kreditních karet, pokud uživatel použil ve službě Internet Passport platbu kartami pro virtuální obchodní domy. [4]

Uvedené příklady mají jen dokumentovat význam kvality software. Ve vyjmenovávání dalších by bylo možno úspěšně pokračovat.

Proč se problematika zvyšování jakosti software neprosazuje ve větší míře ve světě i u nás?

Objektivní skutečnost je taková, že problematika kontroly tvorby a vyhodnocení kvality software je podstatně složitější problém, než u většiny běžných výrobků a služeb (např. strojírenských součástí, chemických směsí, apod.). Software je nehmotné zboží! Současné programové produkty jsou nesmírně komplikované. Neexistují jednoduché a efektivní metody pro vyhodnocení kvalitního software. Tuto skutečnost respektoval i proces certifikace podle norem ISO 9000 v letech 1995 až 1999, když se primárně zaměřil na základní průmyslové produkty a elementární služby, nikoliv na software.

Kromě těchto objektivních faktů, existuje však i řada subjektivních faktorů, které představují bariéry v zavádění systémů řízení jakosti v softwarových firmách.

Uvedme jen některé:

- Odborníci a propagátoři jakosti obvykle nerozumí problematice software a jeho tvorbě.
- Programátoři většinou podceňují problematiku jakosti a filosofii TQM převážně ignorují nebo jí prostě nerozumí.
- Velké softwarové firmy kvalitu software přehlížejí ať již proto, že nepatří k jejich marketingové strategii (Microsoft), nebo v důsledku překotného vývoje svých programových produktů (ostatní firmy).
- Běžný zákazník si myslí, že se musí s chybami v programech smířit v čemž ho programátoři vehementně utvrzují.
- Programátoři velmi často redukuje problematiku jakosti programů jen na problematiku testování, což je samozřejmě zjednodušený a nekomplexní přístup.

Výzkum, který byl prováděn u nás v rámci projektu Evropské unie SPIRE [1], vykázal následující současné překážky ve zvyšování jakosti programů v tomto pořadí četnosti:

1. Jiné
2. Technika
3. Finance
4. Produkt
5. Poptávka
6. Pracovníci
7. Legislativa

Zajímavý je důvod JINÉ, který byl nejméně zajímavý je druhý nejméně zajímavý důvod, který uvádí technické vybavení jako zdroj nespolehlivosti software, což je, při současné vysoké spolehlivosti mikroelektroniky, jasná výmluva!

Zajímavé také je, že 14% českých firem (dotazníky byly rozeslány 850 firmám z nichž 108 vrátilo vyplněný dotazník) stále nepovažuje oblast řízení kvality za důležitou. Celkem 47% firem uvedlo, že nemají dostatek informací o ISO 9000 pro zavedení systému řízení jakosti. Nezanedbatelný je i počet softwarových firem, které uvedly, že kvalita se nedá v jejich oboru aplikovat – celkem 12%. Pouze 8% firem uvedlo, že má certifikát svého systému řízení jakosti.

Uvedeny jsou obecné příčiny nejakosti software. Některé další, vybrané příčiny budou uvedeny v následujících odstavcích.

Na závěr konstatování neutěšeného stavu uvádí autor tvrzení, na základě kterého byly zpracovány další kapitoly:

Dílčí pohledy např. testování, jako kontrola kvality programu, nemůže přinést potřebné zvýšení kvality. Řešení je pouze v systémovém přístupu.

Tento přístup je ostatně obsažen i v zásadách revidovaného souboru norem ISO 9000.2000 (viz zásadu č. 5 System approach to management).

Kompletní analýza problému kvality software ze všech relevantních hledisek by byla velmi rozsáhlá. Proto s ohledem na omezený rozsah příspěvku se zaměříme jen na presentaci následujících, vybraných hledisek:

- Hlediska softwarového inženýrství
- Hlediska ekonomického
- Hlediska organizačního
- Hlediska personálního

Tato hlediska autor považuje za nejdůležitější v současné době u nás, i když lze upozornit na skutečnost, že jsou to nejen technicko-ekonomické problémy, ale také právní aspekty, které vedou ke vzniku programů plných chyb. [15]

3. Vybrané pohledy na kvalitu software

3.1 Hledisko softwarového inženýrství

Jeden z důležitých problémů softwarového inženýrství bylo stanovit faktory jakosti programu. To bylo učiněno v normě – ISO/IEC 9126-1 Information Technology – Software Product Quality – Part 1: Quality model, přijaté také jako ČSN a navazujících technických zprávách ISO/IEC 9126-2, ISO/IEC 9126-3, ISO/IEC 9126-4, obsahujících návrhy metrik pro měření jednotlivých atributů jakosti. Dále pak v šesti normách řady ISO/IEC 14598 Information Technology – Software product evaluation, popisujících jak jakost hodnotit z různých pohledů. I tyto normy byly převzaty jako ČSN. V budoucnu je záměr nahradit tyto normy pro jakost produktu řadou norem ISO/IEC 25000, připravovanou v rámci projektu SQuaRE. Blíže viz [26] a [27], a příspěvek Prof. Vaníčka na této konferenci..

Tyto normy stanoví následující charakteristiky jakosti:

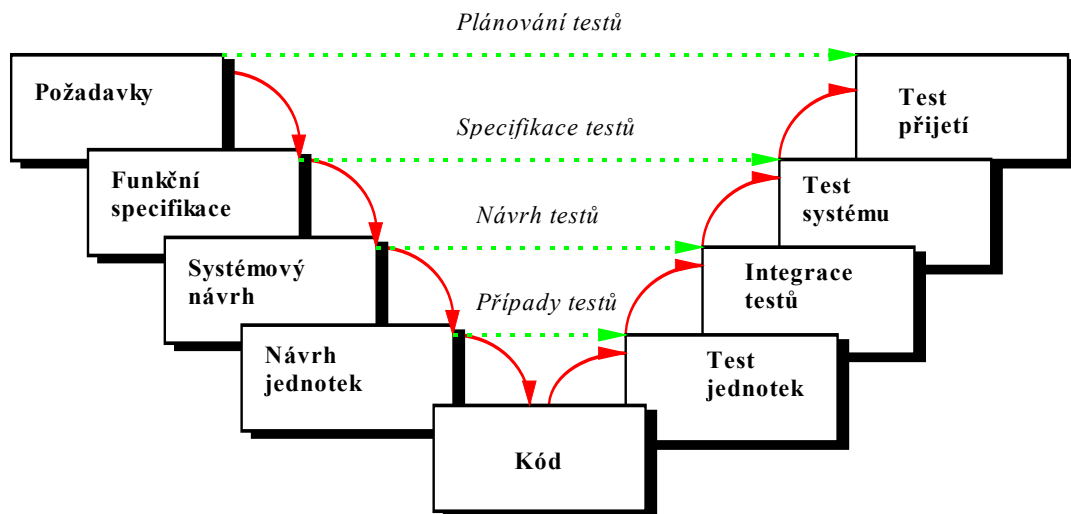
- Funkčnost
- Spolehlivost
- Použitelnost
- Efektivnost
- Udržovatelnost
- Přenositelnost

Každá z nich se pak rozkládá na několik podcharakteristik. Tyto podcharakteristiky lze hodnotit různými měřitelnými atributy.

K charakteristikám a podcharakteristikám byly doporučeny interní a externí míry (ve starší terminologii metriky) jednotlivých atributů uvedených charakteristik jakosti.

Tyto normy a technické zprávy se zatím u nás příliš nerozšířily. Většina softwarových firem, ani zákazníků nehodnotí software podle nich.

Softwarové inženýrství v současné době nemůže prokázat správnost programu dokazováním. Může pouze dokazovat testováním, že program určitou chybu má. Tato situace je z hlediska hodnocení kvality programu velmi nepříjemná. O to více však klade důraz a vysoké nároky na postup testování software. Softwarové inženýrství vypracovalo tzv. V model, prosazovaný zejména v Německu jako standard [21] :



Obr. 1. V- model životního cyklu

Význam tohoto modelu pro vlastní proces testování popsal u nás pracovník německé firmy Micro TOOL R.Nagy [20].

Rozbory potvrzují, že kvalita testovacího procesu ovlivňují zejména na následující faktory [9]:

- Množství testovacích dat
- Kvalita testovacích dat
- Použité metody testování
- Použité nástroje při testování
- Znalosti testovacích pracovníků

Proto součástí kvalitního vývoje softwaru jsou dobře zpracované plány testů, které určují *Jak, Co, Kdy, Kým a Čím* testovat. Ve většině softwarových firem se však takové plány v návaznosti na V-model nevypracovávají. Proces testování se provádí nahodile a pracovníci většinou poslušnost testů improvizují. Zřídka kdy se přistupuje systematicky k celému procesu testování programů [8] a to i přesto, že špatný postup testování zvyšuje existenční rizika softwarových projektů, na což upozornil pracovník firmy Slovakodata J.Kubiš [22] a doložil simulačními modely s využitím Markovových řetězců.

Pro zvýšení kvality testování je potřeba používat řadu testovacích nástrojů, jako jsou:

- Generátory testovacích dat
- Sledovače chodu programů
- Analyzátoři zdrojových textů programů
- Prostředky automatického testování
- atd.

Tyto prostředky nejsou u nás v dostatečné míře využívány. Většina testů se provádí s náhodně sestavenými daty, často jen na vzorcích vybraných dat z praxe. Pro podporu testování se používají v největší míře prostředky připojené ke kompilátoru do jeho vývojového prostředí.

Velkou pozornost věnovalo softwarové inženýrství v 60-tých a 70-tých letech problematice, která se váže ke stanovení množství testů a k zásadám, abychom získaly dobře otestovaný program [23]. V současných programovacích prostředcích však tyto snahy nenašly dostatečnou odezvu.

Bohužel je nutno konstatovat, že většinu programů vytvářejí programátoři stále bez ohledu na skutečnost, že program bude testován! [24] To ukazuje na nedostatečnou metodiku výuky programování, kterou je potřeba v dalším období podstatně zlepšit.

Softwarové inženýrství zhruba od 70-tých let, kdy byly publikovány zásady strukturovaného programování, doporučuje vytvářet programy, které mají minimum chyb. Nikoliv nekvalitní programy s množstvím chyb, které se pak následně hledají pomocí testů a postupně se odstraňují. Pravděpodobnost zavlečení následných chyb při zásahu do existujícího programu je podstatně vyšší, než pravděpodobnost vzniku chyby při správném metodickém postupu. Za tím účelem byla vypracováno velké množství metod, v západních zemích hojně používaných. U nás se o nich vedly diskuse v 80-tých letech, ale nijak se nerozšířily. S používáním metod se objevují dvě krajnosti. Jejich přeceňování a jejich zásadní odmítání. V ČR je hojně rozšířeno odmítání metod obecně, samozřejmě i v oblasti metod programování. Jen poměrně málo softwarových firem u nás zavedlo např. metody programování, využívající UML nebo OMT současně s využitím produktů CASE (Computer Aided Software Engineering). Přitom právě výsledky používání produktů CASE ukazují, že se chybovost programů při jejich používání snižuje až o řád! Používání produktů CASE např. takových, jaké jsou nabízeny firmami UNICORN (www.unicorn.cz - produkty firmy Rational) nebo LBMS (www.lbms.cz - produkt Select Component Architect) by rozhodně zvýšilo nejen kvalitu našich produktů, ale i produktivitu českých programátorských týmů. Proto je velmi potěšitelné, že v souvislosti s nástupem objektově orientované technologie a s ní související metodou UML, byla u nás vytvořena metoda BORM [13], která přispěla ke zlepšení analýzy a návrhu současných, objektově orientovaných informačních systémů nejen u nás, ale následkem spolupráce tvůrců se zahraničními pracovišti i v cizině.

K tradičně opomíjené problematice u nás patří postup při vytváření dobré specifikace software. Přitom se jedná o klíčový moment v otázce jakost software! Jak můžeme hovořit o kvalitě softwaru, když nejsou specifikovány a zaznamenány zákaznickovy požadavky – viz zásada komplexní jakosti- *Jakost je schopnost produktu nebo služby uspokojit zákaznickovy vyslovené i předpokládané požadavky*. K podpoře sledování a správy požadavků jsou přitom dnes k dispozici velmi efektivní nástroje (Caliber RM viz stránky LBMS).

3.2 Hledisko ekonomické

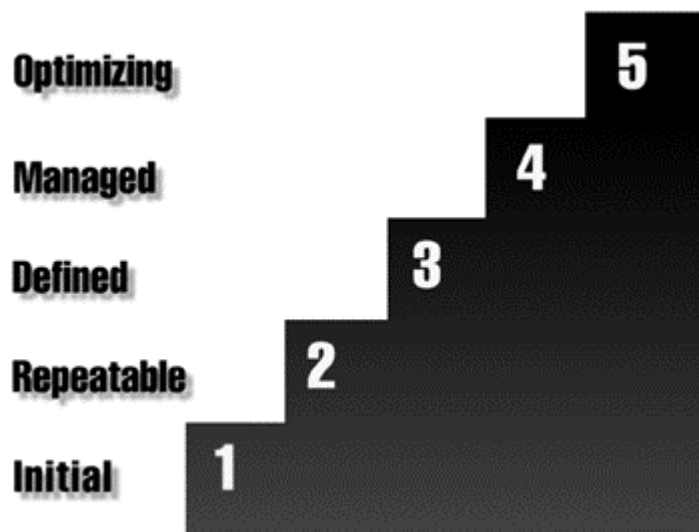
Softwarové firmy již léta vykazují vysoké zisky a těží z převisu poptávky po programech všeho druhu, takže je nic nenutí zavádět nákladový controlling. Proto náklady na najakost jsou mimo pozornost ekonomů v softwarových firmách. Ti obvykle jen sečítají naběhnuté náklady, které programátoři vykazují jak z hlediska mzdových, tak i ostatních nákladů. Zatímco např. ve strojírenství, v chemickém průmyslu, elektrotechnice a dalších odvětvích, je kalkulace výrobních nákladů na vysoké úrovni (např. v automobilovém průmyslu se z hlediska výrobních nákladů snaží firmy šetřit doslova haléře), snižování nákladů na tvorbu software je stranou pozornosti jak vedení softwarových firem, tak ekonomů, a dosažené náklady se přenášejí jednoduše na zákazníky.

Náklady na testování programů zejména naši manažeři softwarových firem vždycky kritizují a zpochybňují. Protože nedovedou stanovit náklady na nejakost software, nevidí možné přínosy jakostního software a považují náklady na testování software za zbytečné, takže se je snaží snížit na minimum. O potřebách, účelnosti a významu správného testování se u nás snaží pražská firma KOMIX přesvědčit naši programátorskou veřejnost pořádáním specializovaných seminářů LATES (již proběhly tři ročníky), ale jak vedení softwarových firem, tak programátoři dobře míněné rady zatím ignorují, takže na rozdíl od západního trhu, objem prodeje testovacího software u nás stagnuje, přestože je nabízen v dostatečném sortimentu (viz www.komix.cz – produkty firmy Mercury).

Vztah dodavatel – zákazník je zatím v oblasti software většinou konfliktní, takže dobrá a společně odsouhlasená specifikace programového produktu je řídkým jevem. Jedna i druhá strana se snaží dosáhnou svých zisků různými úhybnými manévry a někdy i dost pochybnými úskoky.[13] Problematika nákladů se pak jeví jako vedlejší, primárně se firmy zaměřují na vyjednávání, dohadování, soudní spory, apod., protože tato cesta dovoluje získat daleko větší finanční částky s daleko menšími náklady. Firmy proto nákladům, které vyplývají z nekvalitního software, nevěnují (a nemusí věnovat) zatím potřebnou pozornost.

3.3 Hledisko organizační

Organizace procesů při vytváření software patří tradičně ke slabinám softwarových firem. Vedoucí pracovníci softwarových firem se málo věnují problematice procesního řízení, takže většina softwarových firem u nás patří do úrovně 1. a 2. softwarového modelu zralosti procesů (SW-Capability Maturity Model – Software Engineering Institut - Carnegie Mellon University) [14]:



Jednotlivé stupně modelu CMM

Jak už bylo konstatováno, metody tvorby software se u nás nerozšířily. Není divu, že se tedy nerozšířily v softwarových firmách ani metody, které souvisejí s postupy pro zvyšování jakosti jako např. FMEA nebo Ishikavovy diagramy [6,7,9] a další (nejnověji SixSigma [25]), které lze aplikovat v rámci procesu tvorby software. Protože se nepoužívají metody pro tvorby programů, nelze dost dobře aplikovat produkty CASE, takže nelze využít schopností počítače pro automatickou kontrolu tvorby software a pro podporu organizování postupu vytváření software.

Konkrétní organizace tvorby softwarových produktů musí být řešena v rámci softwarových projektů. Termín „projekt“ se v našich softwarových firmách sice běžně používá, ale vlastní průběh a řízení projektů je velmi vzdálen zásadám moderního projektového řízení, jak je popisuje norma ISO 10 006 (Směrnice jakosti managementu projektu) a jak je prosazuje např. mezinárodní organizace IPMA (International Project Management Association). Význam této normy pro jakost softwarových projektů ukázal příspěvek pracovníků z Fakulty informačních technologií (Kubát-Kreslíková [12]).

Publikace o testování software, která u nás vyšla v překladu nakladatelství Computer Press, správně poukazuje na význam organizování testovacích postupů [16], ale velmi málo softwarových firem organizuje testování podle v ní doporučovaných zásad.

3.4 Hledisko personální

Programy zatím konec konců vytvářejí lidé. Proto je velmi důležité si povšimnout přístupu programátorů ke tvorbě softwaru. Zkusme si uvědomit některá důležitá fakta:

- Není tajemstvím, že současní manažeři (a pro manažery softwarových firem to platí dvojnásob), jsou orientováni na dosažení rychlých zisků v krátkodobém horizontu. Pak je ovšem jasné, že investice do jakosti a problematika kolem jakosti software nepatří k jejich preferovaným prostředkům.
- Programátoři mají ve zvyku ignorovat vše, co bezprostředně nepatří k programovacím jazykům.

- Je zajímavé, že zákazníci, kteří jsou jinak nároční na kvalitu (Zkuste někomu prodávat v autosalonu špatně nalakované auto s hlučícím motorem, bez jedné gumové zástěrky nebo s odlišným druhem pneumatiky na jednom kole - přitom jsou to „kosmetické“ vady ve srovnání s chybami v SW produktech!) jsou ochotni přijmout nejakostní programy. Pak ovšem prodejci ani tvůrci softwaru nejsou nijak motivováni k jakosti. Zdá se však, se tato situace v poslední době mění. Stále více zákazníků se s chybami v software odmítá smiřovat a v USA se zvyšuje např. počet žalob na firmu Microsoft. [2]
- Programátoři mají dosti malé znalosti o problematice testování. Je to také v důsledku výuky na školách, kde se většinou učí právě jen programování a problematice testování se věnuje minimum pozornosti. V některých případech dokonce žádná pozornost. Skriptum prof. Vaníčka je [11] o jakosti software je příslovečnou bílou vránou. V minulosti se této problematice věnoval doc. Hořejš, který ve své eseji na semináři SOFSEM 76 velmi vtipně hovořil o prevenci, diagnose a terapii při testování programů. [10]
- Zanedbatelný počet vědeckých a výzkumných pracovníků ve srovnání se západními zeměmi se věnuje problematice jakosti software. Práce doktorandů v této oblasti jsou spíše výjimkami. [16, 17, 18]
- Široká veřejnost u nás nahlíží na problematiku jakosti a norem ISO 9000 jako na případ zakuklené eurodiskriminace naší výroby a projev eurobyrokracie. Viz článek o isoještěrech a podobné případy. [5]

Z výše uvedeného je patrné, že problematika jakosti software si vyžádá ještě mnoho úsilí, aby byly překonány negativní postoje, které nepříznivě působí na proces tvorby software.

4. Závěr

Nesystémový přístup ke kvalitě nemůže v dostatečné míře vyřešit problematiku tvorby potřebného velkého množství kvalitního software pro současnou informační společnost.

Systémový a systematický přístup, na jehož základě firma navrhne a zavede dobrý systém řízení jakosti, přináší firmě značný přínos a nezanedbatelnou konkurenční výhodu, což platí i pro softwarové firmy.

Tlak na kvalitu software stále poroste, protože filozofie vysoké jakosti tvoří součást základní vize Evropské unie.

Proto softwarové firmy budou postaveny i u nás již od tohoto roku před problém vybudování efektivního systému řízení jakosti podle ISO 9000:2000. Analýza současného stavu a odstranění dosavadních bariér pro dosahování vyšší kvality jsou nutným předpokladem k vytvoření dostatečně účinných procesů, zajišťující vysokou kvalitu našeho software.

Velmi komplikovanou situaci z tohoto hlediska budou mít softwarové firmy, které budou realizovat programové vybavení pro řídicí systémy, pracující v režimu reálného času. Proto se na tuto problematiku zaměřují i výzkumný záměr „Automatizace technologií a výrobních procesů“.

Kromě problémů, které problematika řízení kvality představuje z hlediska softwarového inženýrství, bude potřeba se zaměřit na změnu postojů jak vedoucích pracovníků, tak programátorů a analytiků v softwarových firmách. S tím úzce souvisí i zařazení této problematiky do výuky na našich školách.

Jak už bylo zdůrazněno, EU má jednoznačný cíl ve zvýšení jakosti všech svých realizovaných výrobků a služeb. Tato konference je pořádána v rámci akcí Národního programu pro jakost (viz www.npj.cz). Tím se chtějí organizátoři konference přihlásit k pozitivnímu a konstruktivnímu přístupu ke kvalitě software u nás.

Referáty na 30. Jubilejní konferenci Tvorba softwaru 2004 ukazují konkrétní postupy a nástroje, které umožňují kvalitního software skutečně dosáhnout.

Nezbývá než vyslovit přesvědčení, že se naše softwarové firmy s problematikou kvality software v nové Evropě na počátku XXI. století úspěšně vypořádají a že software tak bude moci tvořit důležitou složku naší hospodářské produkce.

Literatura:

1. Nedomová, L.: Řízení kvality v českých softwarových firmách. Business World červen 2000, str. 34-35, příloha časopisu Computerworld
2. Čermák, M.: Software bez záruky. Lidové noviny ze dne 4.října 2003, Rubrika NÁZORY
3. Chyby v software stojí USA ročně 59,9 miliardy dolarů. AUTOMA, roč. (2002), č.11, str. 50, (přetištěno z ARC News, 28.6.2002)
4. Chyba Microsoftu ohrozila miliony uživatelů. Lidové noviny ze dne 10.května 2003, příloha Ekonomika
5. Isoještěři. Dopis čtenáře Hospodářských novin V.Hubače. Rubrika „Z dopisů čtenářů“ v Hospodářských novinách
6. Lacko,B.: Metoda FMEA v softwarovém inženýrství Sborník celostátní konference TVORBA SOFTWARE 97.TU Ostrava EF-KIE 1997, str.121 – 125
7. Lacko,B.: Aplikace Ishikawových diagramů v jakosti software Sborník konference „Tvorba software 98“, TANGER 1998 Ostrava, str..87-93
8. Lacko,B.:Systematický přístup k testování programů In: Sborník PROGRAMOVÁNÍ '82. Dům techniky ČSVTS Ostrava 1982, str.108-127
9. Lacko,B.:Využití teorie kauzalizy v jakosti software In: Sborník z mezinárodní konference Evropský týden kvality v České republice. Česká společnost pro jakost Praha 2003, str. 493-503
10. Hořejš, J: Ladění programů (Esej o prevenci, diagnose a terapii) In: Sborník semináře SOFSEM'76. Výzkumné výpočtové středisko Bratislava 1976, str. 7-37
11. Vaniček, J.: Měření a hodnocení jakosti informačních systémů. ČZU v Praze – Provozně ekonomická fakulta, Praha 2000. 211 stran
12. Kubát,L.- Kreslíková,J.: Jakost managementu projektu a ISO 9001. IN: Sborník semináře PROMA03. Evida Plzeň 2003, str. 95-108
13. Polák,J.-Merunka,V.: Carda,A.: Umění systémového návrhu. Grada Publishing Praha 2003
14. Paulk C.M.- Curtis B.- Chrissis B.M.- Weber V.C: Capability Maturity Model. IEEE Software, Vol. 10, No. 4, July 1993, pp. 18-27
15. Thibodeau, P.: Chyby softwaru zatím nejsou minulostí. Computer World č.45/2003, str.30
16. Patton,R.: Testování software. Computer Press Praha 2002
17. Halva, M.: Modelování jakosti softwaru pro informační systémy. (Disertační práce) Fakulta strojního inženýrství VUT v Brně, Brno 2003
18. Pyrochta, T.: Navrhování testovacích prostředků a metod pro aplikace informačních systémů v prostředí klient/server. (Disertační práce), Fakulta strojního inženýrství VUT v Brně, Brno 2002

19. Tonar, J.: Měření a hodnocení jakosti informačních systémů specializovaných typů. (Disertační práce). Fakulta provozně-ekonomická ČZU v Praze, Praha 2002
20. Nagy,R.: Procesné modely testovania softwéru. AT&P Journal č.12/2003, str. 17-19
21. Bröhl,A.P.-Dröchel,W.: Das V-model – Der standard für die Softwareentwicklung mit Paxisleitfaden. Oldenbourg Verlag, Wien 1993
22. Kubiš, J.: Existenčné riziká projektu a k nemu viazaných subjektov. In: Sborník Tvorba softwaru 2002. VŠB Ostrava 2002, str. 109-116
23. Knuth, D.E.- Stewnsen,F.R.: Optimal measurement points for program frequency counts. BIT 13 (1973), 313-322
24. Boehm B.W.: Software and its impact: Quantitative assessment. Datamation vol. 19, (1973), 5, 48 – 59
25. Kozubík,L.: Six Sigma v informačních technologiích, proč ne? (viz [www. cvis.cz](http://www.cvis.cz) stránky centra výzkumu informačních systémů Univ. T.Bati Zlín – položka Články)
26. Vaníček, J.: Stav a perspektivy mezinárodní normalizace v oblasti měření a hodnocení jakosti informačních a softwarových produktů. ČZU Praha, Provozně ekonomická fakulta, Praha 2004, 68 stran, ISBN 80-213-1129-0
27. Vaníček, J.: Projekt SQuaRE – Připravovaná řada norem ISO/IEC 25000 pro jakost produktu. Magazín ČSN, ročník 12, č. 9, 2002, s. 267-273 a č. 12, s. 283, ISSN 0862-7932

Zpracováno v rámci výzkumného záměru MŠMT J22/98:26000013 „Automatizace technologií a výrobních procesů“