# Kvalita softwaru a související metody jeho tvorby

**Čestmír Halbich**, Katedra informatiky, Česká zemědělská univerzita v Praze, Provozně-ekonomická fakulta, 165 21 Praha 6-Suchdol, halbich@pef.czu.cz

**Abstrakt**

Jedním z trendů současné turbulentní doby je využívání virtuálních organizací k dosažení zisku a přežití organizace v tržním prostředí. V příspěvku je úvodem charakterizována jak současná doba, tak i podstata virtuálních organizací. Virtuální organizace jsou životně závislé na softwaru a zejména na jeho kvalitě a bezchybnosti. Dále je tedy definována kvalita softwaru podle normy ISO 8402 a norem souvisejících, a metody zvyšování jeho kvality jako objektově orientované technologie, využití business objektů v praxi, vztah k business process reengineeringu apod.. Dalšími klíčovými problémy je společenská objednávka objektově orientovaného programování a překonávání sémantické mezery při tvorbě softwaru. Kvalita softwaru dosažená efektivně optimální metodou tvorby softwaru je jedním z prvků pozitivně ovlivňujících i bezpečnost informačních systémů.

## Úvod

Žijeme v období informační exploze, můžeme hovořit o *teorii přežití*, ať již živočišného druhu (třeba člověka), nebo podnikatelského subjektu a v tomto případě je tedy podvědomě cítit význam *pro organizaci citlivé informace*.

Práce odráží autorovu zkušenost z výuky aplikované informatiky a managementu od roku 1986 až po dnešek, nejdříve na Policejní akademii České republiky v Praze a od roku 1995 na katedře informatiky provozně ekonomické fakulty České zemědělské univerzity v Praze.

Na závěr první části úvodu můžeme citovat představitele systémové vědy Rosického [1], str. 16 - „Neexistují a nemohou existovat jasné, spolehlivé a univerzálně platné návody, jak dosáhnout úspěchu. Ten není možné dosáhnout bez osobního úsilí a individuální odpovědnosti za vlastní poznání."

Snad každá rozsáhlejší práce zabývající se problematikou informatiky dokumentuje nutnost nějakého (autorem zvoleného) řešení známým popisem situace v informačních technologiích (dále jen IT) z doby zcela nedávné, viz například [2], následuje citace:

„- méně než dvě procenta projektů byly použity přímo, tak jak byly dodány;
- další tři procenta projektů byly použity, ale tyto projekty byly předtím změněny;

- celých 19 procent projektů bylo použito po značném přepracování, nebo bylo opuštěno;
- podstatných 29 procent projektů nebylo nikdy použito, ačkoli byly v některých případech i zaplaceny (vapourware);
- zbývajících 47 procent projektů bylo nabídnuto a odevzdáno, ale nikdy nebyly ve skutečnosti provozovány (shelfware)!"

Situace v informačních technologiích je dnes o něco lepší, než jak ji popisuje uvedený citát, snad také díky novým technologiím jako je např. objektově orientovaná technologie programování (dále jen OOP), viz např. Merunka [3] apod.

Jak je známo, softwarová krize byla poprvé veřejně rozpoznána na konferenci NATO „Softwarové inženýrství" v Garmischi v roce 1968. Tato konference identifikovala problém a začala diskusi o jeho řešení. Třicet let po konferenci v Garmischi stále ještě hledáme řešení, softwarová krize stále přetrvává ve formě trhliny mezi komerčními koncovými uživateli a vývojáři IT. Bylo vytvořeno mnoho technik k překonání této mezery, mimo jiné OOP, avšak dosud s nevelkým úspěchem.

Pesimistický americký specialista jisté světové softwarové firmy dokonce tvrdí, že investice do informační technologie se nemohou nikomu zaplatit. Slouží jenom k získání převahy nad konkurencí[1] viz [4]. Tento specialista dále hovoří o tom, že architektura IS se vyvíjí a špatně vybraný informační systém je tragédie. Škodolibě dodává, že ze života vidíme, že tragédií je většina.

**Charakter současné epochy**

Dnešní doba se všeobecně charakterizuje jako *turbulentní*, což zároveň anticipuje pojem *chaos*.[2] Jako analogii situace ve fyzice to již v roce 1980 uvedl P. F. Drucker. Vývoj jeho jasnozřivé názory jen potvrdil. Turbulentní doba dává připraveným lidem více příležitostí, než doba, která tento přívlastek nemá. Ve shodě s mnoha autory se v předkládané práci turbulentní prostředí pojímá ne jako ohrožení stávajících systémů, ale jako výzva pro tyto systémy a jako existence nových příležitostí k jejich prosperitě. Přitom metod posilujících schopnost přežití podnikatelského subjektu v podmínkách chaosu je celá řada[3], jak pojednává např. [5].

**Virtuální organizace**

Virtuální organizace může být popsána jako společnost okamžitě produkující a provádějící marketing vysoce užitnými produkty nebo službami ve velkém množství[4]. Virtuální produkt nebo služba je úmyslná (záměrná), přichází vždy když je vyžádána ve formě události, a je vždy adaptována na požadavky zákazníků. Virtuální korporace je plovoucí, flexibilní a adaptabilní, tj. není závislá na definitivní strukturní formě moderní organizace. Virtuální korporace může být pojata jako

[1] Rosický [6], str.47 uvádí, že je vhodnější označení ne konkurenční výhoda, ale konkurenční schopnost a dále také viz Moore, Death of Competition

[2] jak uvádí Halbich [7], je přitom dosti těžké chaos například z časových řad diagnostikovat

[3] jednou z cest jsou virtuální organizace, jež jsou životně závislé na softwaru a zejména na jeho kvalitě. Otázkami kvality softwaru se zabývá Vaníček [8]

[4] paradoxně často v individuálním provedení na základě požadavku konkrétního zákazníka

začátek organizační formy postmoderní éry, do které právě vstupujeme, a která je charakterizována globálním světovým trhem, změnou zákaznických požadavků a zrychlováním dynamiky prostředí a technologického rozvoje. Schopnost rychlé reakce organizace na dynamiku prostředí přináší nejkritičtější faktor úspěchu v tomto světě konstantních změn.

Informační technologie jsou považovány za začátek hlavní schopnosti pro nové a produktivnější organizační formy. IT mění svoji roli z v podstatě nástroje podpory pro existující organizace na moderátor hledání nových organizačních forem, které jsou vhodnější k navigaci na vysoce soutěživých trzích.

Mnoho času, peněz a úsilí je investováno do *redesignu* nebo *business process reengineeringu* organizací. Jsou vyvíjeny strategické aliance, společnosti podstupují *„downsizing"* a části z nich jsou podrobeny proceduře *„outsourcing"*. Vzdor těmto přístupům má mnoho společností málo efektivity a efektivnosti v adaptaci na vzrůstající dynamiku prostředí. Dá se říci, že se mají hledat nové a možná dokonce nevyzkoušené koncepty k zabezpečení přežití a úspěchu v postmoderním světě.

V závislosti na podstatě organizace, virtualita může být vyjádřena různými způsoby. Nicméně můžeme identifikovat čtyři základní charakteristiky z kterých virtuální organizace může vykazovat jednu nebo více těchto charakteristik (jako jsou *nezávislost na čase a prostoru, je založena na znalostech, externí kooperace, dočasné aliance)*.

Čas a prostor vždy omezovaly návrh a vývoj organizace[5]. Ve virtuální organizaci ztrácejí svoji omezující podstatu, jedinci mohou pracovat doma, ve vlaku, letadle atd. při použití nejnovějších IT jako jsou mobilní telefony připojené k přenosným počítačům, virtuálním sítím atd. Existuje již tedy dnes nová flexibilní forma práce. Nicméně prostorová nezávislost není pouze jediným aspektem virtuální organizace. Soudobé, moderní IT vedou k snadné, flexibilní a přirozené komunikaci.

Budoucí organizace budou méně „fyzické" a více založeny na znalostech a vnitřních kompetencích svých členů. Více a více z organizace je uvnitř mysli lidí a uvnitř IS. Stabilita a kontinuita je založena na lidech, jako součásti korporace. Tento koncept dává významnou výzvu tradičnímu chápání korporace jako sestavy fyzických zdrojů (strojů, kapitálu a práce) včetně změny zaměření z dělby podle práce na dělbu podle znalostí a kompetencí. Vznik týmově orientované korporace je příkladem přijetí znalosti jako významného faktoru restrukturalizace korporace. Virtuální organizace se proto zdá více „neviditelná"[6] než tradiční korporace.

Tradičně bylo dosti jednoduché definovat hranice mezi organizací a jejím okolím. Naproti tomu virtuální organizace jsou charakterizovány rozmazanými, nejasnými hranicemi vůči například dodavatelů a zákazníkům. Staly se více plovoucími a flexibilními, a dokonce zde IT hrají rozhodující povolovací (aktivovací) roli. Masová spotřeba produktů a služeb, jakož i významnější vliv zákazníků na vývoj a návrh

[5] Z ekonomické teorie jsou známy jako jedny z produkčních faktorů a zdrojů a jsou svou podstatou vzácné.

[6] Virtuální organizace dovedená ad absurdum - tvorba virů pomocí Internetu. Spolupracující autoři virů se nikdy osobně neviděli (konspirace)

produktů a koncentrace korporace na podstatné požadované kompetence více zdůrazňují spolupráci s dodavateli a zákazníky. Do budoucna se očekává další stírání hranic mezi korporací a dodavateli a zákazníky. *Paradoxně podnikové tajemství, strategické plány a zdroje mohou být sdíleny s dodavateli a zákazníky*[7] ve společném snažení a usilování o redukci času v přístupu na trh a rychlé adaptabilitě na změnu zákaznických požadavků. Virtuální organizace může významně zvýšit své kapacity, všechny služby, které nemusejí být dodávány z jádra organizace jsou „kupovány" od externích vlastníků na bázi záměrně dosahovaných společných cílů.

**Bezpečnost jako konkurenční výhoda**

Je jisté, že rychle získané a rychle zpracované informace jsou nenahraditelnou konkurenční výhodou v boji o zákazníka. Především čas je tím kritériem, které způsobilo tak překotný vpád počítačů do kanceláří, za přepážky bank a do výrobních hal. Ve firmách, kde se podařilo informační technologie rychle integrovat do celého provozu a řízení podniku, je samotný systém donutil pracovat nejen rychle, ale i pružně (metoda pokusu a omylu). Umožnily jim rychle reagovat, rychle korigovat případné chyby, rychle implementovat nové nápady.

Pokud se podaří pracovat se zdroji informací efektivněji a účinněji než konkurenci, představuje to výhodu v konkurenční soutěži na trhu [9]. Pokud k základním informacím přidáme ještě hodnotu pomocí analytických informací získaných např. metodou datového skladu, stávají se informace zbožím s vysokou tržní hodnotou a pro konkurenci budou značně lákavé. V tomto mechanismu se vkládají do technologie a provozu IS nemalé časové a finanční zdroje, a z výše uvedených důvodů by se měla část zdrojů věnovat i na ochranu informací.

Pokud firma nechrání dostatečně své informační zdroje, díky dostupnosti informací pro toho kdo je chce získat nelegálně, konkurence bude úspěšně parazitovat na našich zdrojích.

**Vývoj programovacích technik**

Je potřeba vymezit hlavní pojmy v příspěvku, když přesnost je jistě jedním z atributů používání inženýrských postupů.

Inženýrství má mnoho různých definic, ale můžeme se podívat například do Britské encyklopedie, která ve volném překladu říká, že „*Inženýrství* je aplikace vědy pro optimální využívání disponibilních zdrojů k blahu lidstva". Inženýrství je tedy spojeno s vědou, zároveň vysoce preferuje možnosti měření různých veličin, protože pokud bychom neměřili, těžko bychom mohli říci, že něco využíváme lépe, tedy optimálněji než dříve. Co není „měřitelné" obvykle nebývá předmětem zájmu inženýrů. Proto kromě definic musíme zavést jednotky pro měření jimi sledovaných veličin. Kromě toho technici se snaží vždy o dosažení co nejvyšší účinnosti.
Podíváme-li se na programování jako na inženýrskou činnost, je nápadný jeho velmi dynamický rozvoj, který lze označit jako stálý boj se složitostí. Tento vývoj probíhal v určitých vlnách, které mají výhradně metodický charakter (Technický pokrok

[7] Na tento paradox upozornil např. Halbich [7]

programátorské problémy neřešil, spíše je prohluboval, protože výkonnější počítače znamenaly možnost větších, tj. rozsáhlejších i složitějších programů). První programátoři neměli jinou metodickou pomůcku než svůj vlastní „selský" rozum, ale skoro ihned se přišlo na první typ strukturalizace na podjednotky (podprogramy a makra). Na programování později vadily dvě věci - příliš velká pracnost a individuální charakter programů, tj. že údržba programů byla možná jen jeho autorem. Programování dostávalo průmyslový charakter a vznikal tlak na jeho technologizaci (software by měl být produkt jako každý jiný). Metodiku členění programu na podjednotky dovedl teoreticky do důsledků E. W. Dijkstra svými „dobře strukturovanými programy" a technologii strukturovaného programování pak vytvořil Jackson. Následovaly další etapy vývoje programovacích metodologií, které skončily dosud poslední široce přijímanou technologií - *objektově orientovaným programováním*. K výhodám OOP patří to, že dovoluje aplikovat v oblasti tvorby programového vybavení pro počítače metodu „softwarových čipů", která má analogii v tolik osvědčeném a efektivním způsobu výroby počítačů a jiných elektronických výrobků. To sice neznamená že OOP musí být stejně úspěšné, ale vidina stejného úspěchu je jistě svůdná a stojí za námahu spojenou s vývojem technologie OOP obecně. Jak již bylo řečeno, současná doba potřebuje naléhavé metody, které by dovolily inovovat stávající systémy. Navíc praxe ukazuje, že jen malé procento viz [2] vyvinutých produktů se použije bez úprav. Větší část je potřeba upravit a později dále *měnit s ohledem na měnící se požadavky uživatelů*. Koncept OOP je založen právě na realizaci těchto požadavků.

Z technické praxe víme, že každá věc něco stojí. Výhodou objektově orientované technologie je, že zvýšené nároky, které vznáší na technické prostředky výpočetní techniky (větší potřeba operační paměti, větší požadavky na výpočetní výkon počítače) jsou saturovány nepředstavitelně rychlým rozvojem hardware.

**Jakost softwaru a jeho vztah k bezpečnosti IS**

Pojem kvality intuitivně cítíme, přesto je nutno zmínit některé normy ke kvalitě se vztahující.

Obvykle jsou používána slova kvalita a jakost jako synonyma. Podle jazykové normy je správným českým termínem „jakost", i když slovo kvalita je v běžné řeči frekventovanější. Přesto budeme většinou používat slova jakost.

Jakost je definována pro jakékoliv produkty lidské činnosti, nejen pro oblast informačních systémů. Obecná definice jakosti podle normy ISO 8402 definuje jakost jako souhrn význačných rysů a charakteristik produktu nebo služby, které určují schopnost produktu uspokojovat obvyklé nebo stanovené či vyžádané potřeby, za předpokladu využívání výrobku stanoveným způsobem. Pro daný okruh výrobků mohou totiž být některé potřeby všem uživatelům společné, a pro všechny samozřejmé, jiné potřeby mohou být pro daného uživatele či skupinu uživatelů specifické.

Uvedená definice jakosti jednak nezahrnuje cenu (jakost tedy nebude jediným údajem, který bude vstupovat do rozhodovacího procesu výběru výrobku či řešení), jednak je založena pouze na potřebách, ať již obecných, či specifických (Za úvahu

by stálo, zjistit, kolik našich potřeb je skutečně nezbytných a kolik uměle vyvolaných např. masivní reklamní kampaní). Atributy výrobku, které nepotřebujeme, tedy zvýší patrně jeho cenu, ne však jeho kvalitu, protože míru uspokojení našich potřeb neovlivní. (Tato situace poměrně často nastává v oblasti tvorby softwaru pro počítače PC IBM kompatibilní, protože uživatel většinou využívá možností komerčních produktů pouze z několika procent.) Je však s podivem, kolik uživatelů si tuto nezdravou situaci neuvědomuje. A pokud si ji uvědomuje, stejně se vůbec nechová tržně a podléhá diktátu producentů výrobků.

Úsilí o dosažení jakosti se projevilo ve zpracování mezinárodních norem řady ISO 9000, 9001, 9002, 9003, 9004, které souhrnně popisují, v čem jakost spočívá, jak ji dosáhnout a jak ji řídit (platí pro jakékoliv produkty). Vezmeme-li v úvahu tvrzení, že software je produkt stejný jako každý druhý, měl by být prodáván na trhu se všemi náležitostmi jako ostatní produkty. Tj se zcela jednoznačně stanovenou zárukou na vady, možností jakost přesně změřit (pravděpodobně při přejímce zboží) a tím případné vady prokázat a potom vadný produkt vyměnit za kvalitní, či odstoupit od kupní smlouvy se všemi důsledky. Tato sentence nám zní jako z jiné planety, neboť uživatelé a zákazníci zatím stále dobrovolně a současně nesmyslně přistupují na prodejní politiku dominantních případně i jiných výrobců softwaru a vzdávají se svého práva na ochranu spotřebitele. I v zemi tak chránící práva zákazníků jako jsou USA, je situace velice podobná. Např. za banální poškrábání laku na novém voze BMW získal před několika lety zákazník u soudu od výrobce náhradu cca 300 tisíc dolarů, za což by si mohl koupit téměř deset takových nových vozidel, zatímco chyba v softwaru vedoucí ke zmíněným škodám případně i smrti postižené osoby zdaleka není ekvivalentně odškodněna ani v USA. Co teprve u nás, kde na zákoně o ochraně spotřebitelů sotva stačil uschnout inkoust, a něco tak subtilního a diskutabilního jako software si zřejmě budou soudy přehazovat jako horký brambor (už třeba jen kvůli místní příslušnosti). Přitom řešení je nasnadě - sledování kvality, dosahování kvality např. pomocí OOP a následně měření kvality softwaru. Mezinárodní normou ISO/IEC 9126 bylo pak definováno šest charakteristik kvality softwaru, s co nejmenším rozsahem překrytí *( funkčnost, bezporuchovost, použitelnost, efektivnost, udržovatelnost, přenositelnost)*.

Z toho nejméně tři charakteristiky se velmi úzce dotýkají bezpečnosti IS. Požadavky různých uživatelů budou různé (srovnej požadavky na bezporuchovost softwaru jaderné elektrárny, řízení letového provozu, řízení nemocničních přístrojů, výukových programů, počítačových her...).

Podle [10] bezpečný software musí být navržen „správnou" technologií, co je to „správná" technologie však nespecifikuje. Že neexistuje bezchybný software ani od renomovaných firem ani obecně, dokládá např. Stanek [11].

Velmi důležitou vlastností se zdá složitost softwaru. V dnešní době se uvádí přes 200 měřitek jako předpokladů a atributů kvality. Více než 100 z nich jsou metriky pro měření složitosti softwaru. Z tohoto pohledu pak plyne, že měření složitosti softwaru má klíčový význam pro všechny další kvalitativní charakteristiky, jak byly uvedeny výše. Dá se říci, že složitost je hlavním problémem v teorii i měření kvality softwaru.

Metriky pak popisují empirickou strukturu (např. informační systém a jeho vlastnosti) užitím formální struktury, zpravidla čísel. Můžeme jimi vyjadřovat klasifikaci a porovnání pokud vyjadřujeme své preference. Jinak je to, když chceme s naměřenými čísly počítat (kdy je určitý softwarový produkt dvakrát kvalitnější než druhý?). Cílem měření není „získat čísla sama o sobě", ale získat relevantní závěry o složitosti a jakosti softwaru. Měřením musíme převést problém rozhodování o v podstatě přímo neuchopitelné realitě pomocí homeomorfismu na matematický model, kde většinou existují vhodné nástroje pro řešení. Poslední a nejdůležitější etapou je interpretace získaných výsledků.

Neexistuje dosud zcela konzistentní teorie pro vztah složitosti a kvality softwaru. V první fázi je nutno převést program na vývojový diagram. Vaníček [8] zavedl pro vyjasnění situace v převodu programu na vývojové diagramy čtyři teorémy, na jejichž základě lze rozhodnout o složitosti softwaru, např. programy se stejnými vývojovými diagramy jsou stejně komplexní, nebo další - vývojový diagram kombinace programů závisí pouze na vývojovém diagramu každého z programů a další. Z definice jakosti vyplývá, že kvalitu nelze hodnotit absolutně, ale z hlediska specifických potřeb uživatele. Tyto potřeby se liší především nároky na jednotlivé charakteristiky kvality

Tyto charakteristiky kvality se mohou velmi podstatně lišit v závislosti na typu hodnoceného výrobku a jeho předpokládaných uživatelích. Váha bezporuchovosti a použitelnosti bude naprosto opačná u softwaru určeného pro přímé řízení jaderné elektrárny (bezpečný software pro nevelký počet dobře vyškolených pracovníků) a u výukového software či her. Existuje tedy konečný počet N charakteristik jakosti, z nichž každá bude mít hodnotu vyjádřenou kladným reálným číslem $k_1, k_2, ..k_N$.

Pro zjednodušení budeme místo měření atributu popisujícího některou či některé charakteristiky kvality, hovořit o měření kvality.

Podstatou odhadu jakosti je stanovení, které empirické objekty uspokojují uživatele „lépe" či „stejně dobře nebo lépe" než jiné, obvykle konkurenční. Tyto preference lze vyjádřit pomocí relace na množině empirických objektů. Můžeme napsat $a \geq b$ a „je stejně dobrý nebo lepší" než b. Dva zcela různé objekty mohou být z hlediska našich preferenci hodnoceny zcela rovnocenně. Většinou hodnotíme v situaci tzv. slabého částečného uspořádání (relace), kdy posuzujeme jakost z hlediska různých uživatelů, z nichž každý preferuje jiné charakteristiky jakosti. Podstatné vlastnosti studovaných empirických objektů (nejen z hlediska jakosti) lze vyjádřit strukturou (relační $\geq$ a operační $\oplus$), která vyjadřuje množinu objektů a vztahů mezi nimi. Kromě výše uvedeného slabého uspořádání je v empirické struktuře dána operace skládání prvků $\oplus$, která určuje, jak z dvou prvků vytvořit prvek složený. Takovou operací může být např. skládání délek, hmotností nebo při skládání softwaru z komponent užitím daných pravidel např. sekvenčním řazením těchto komponent ($\oplus = \otimes_{seq}$) nebo výběrem jedné z obou možností $\oplus = \otimes_{alt}$.

Jedním z diskutabilních axiómů o metrikách je tzv. *Archimedův axióm*, který říká, že libovolně velký rozdíl metriky v dané stupnici lze vždy kompenzovat a převážit dostatečným počtem opakování jakýchkoliv malých příspěvků, tedy že velké množství velmi malých výhod vždy převáží jakoukoliv závažnou nevýhodu. Bez přijetí

Archimedova axiómu podle Vaníčka [8] nelze použít k hodnocení empirických objektů jinou než ordinální stupnici.

Dalším (viz [8]) problémem je otázka nalezení vhodné empirické struktury pro reprezentaci objektově orientovaných empirických struktur. V těchto strukturách platí tzv. *axióm idempotence*
$A \oplus A = A$

Chování objektu je charakterizováno metodami definovanými svou třídou. Metoda je ve skutečnosti program a může být viděna jako podprogramy vyvolané zprávami, které jsou objektu poslány. Proto tedy může být komplexnost jedné metody měřena mezimodulárními mírami např. užitím výše uvedených doporučení. Ale na uvedeném stupni třída je situace v OOP zcela odlišná od situace v imperativních jazycích. Hlavní výhoda objektově orientovaného návrhu je, že každý problém může být řešen pouze v jednom místě, v nejobecnější třídě s odpovídající vlastností. Další třídy jsou schopny užít všechny metody definované pro obecnější třídy užívající mechanismus dědičnosti. Axiom idempotence je v protikladu s Archimedovým axiomem a tudíž není šance dosáhnout rozšiřující struktury nebo měřítka pro intramodulární metrickou soustavu objektově orientovaného prostředí. Podle [8] se v tomto případě nehodí za formální strukturu čísla, ale podmnožiny dané množiny. Pro skládání metrik pak platí pravidla analogická pravidlům pro skládání pravděpodobností nebo přesněji pravidlům pro skládání funkcí věrohodnosti u posuzování nejistých hypotéz, kde složitost při spojení dvou tříd lze vyjádřit takto:
$m ( A \oplus B ) \geq m (A) + m (B) - m (A \otimes B)$
a nakonec
$\forall A, B \in \tau : A \geq B \Leftrightarrow m (A) \geq m (B)$ viz [8],
což vede v OOP ve svých důsledcích k použití metriky intramodulární složitosti namísto rozšiřitelné struktury či poměrové stupnice.

**Závěr**

Velkou důležitost ve virtuálních organizacích hraje zajištění bezpečnosti počítačových dat a informačních systémů daných virtuálních organizací. Manažeři musí sami rozhodnout, která podniková tajemství a strategické plány budou sdílet se svými zákazníky a odběrateli, a jak je zabezpečí proti nelegálnímu přístupu hackerů a konkurence. Z charakteru soudobé epochy a trendů ve světě vyvozuje autor, že význam kvality softwaru pro přežití organizace v tržním prostředí narůstá. Jednou z metrik pro hodnocení kvality softwaru je i bezpečnost softwaru. Čím je software kvalitnější, tím je bezpečnější. Tuto kvalitu lze dosáhnout s výhodou použitím metod objektově orientovaného programování. Podle Vaníčka [8] je totiž skládání programového celku z jednotlivých částí v objektové technologii subaditivní, nikoli aditivní či dokonce superaditivní, jako při použití jiných metod tvorby softwaru, a zaručuje tedy snadnější dosažení požadované kvality a tedy i bezpečnosti produktu.

**Literatura**

[1] ROSICKÝ, A.: Naděje a limity systémových úspěchů, in Systémové přístupy, Sborník VŠE Praha, 1995

[2] JAYRANTA, N.: Systems Analysis: the need for better understanding, Int. Jour. of Information Management, 1990, No. 10
[3] MERUNKA, V., POLAK, J.: Object-oriented Analysis and Design - Teaching and Application Issues, in Proceedings of 20th ASU International Conference, Prague 1994
[4] NOVOTNÝ, I.: Go global, Chipweek 13/1996, str. 3, Vogel Publishing
[5] HALBICH, Č.: Virtuální organizace a další inženýrské přístupy v oblasti sociálních systémů, Ve sborníku konference Systémové přístupy'97, VŠE Praha 1997, str. 39-43
[6] ROSICKÝ, A.: Informations Systems Concepts For Adaptable Organizations, In Proceedings of IDIMT'96 - 4th Interdisciplinary Information Management Talks, R. Oldenbourg, Wien, Muenchen 1996, pp.33-49,
[7] HALBICH, Č.: Některé problémy systémů pro podporu rozhodování v podmínkách chaosu, habilitační práce, ČZU v Praze, Praha 1997
[8] VANÍČEK, J.: Software complexity as a software quality indicator, Kybernetika, pp. 333-356, volume 33(1997)
[9] HALBICH, Č.: Základy práce s PC IBM II, Ivančice 1998, ISBN 80-238-2068-0
[10]RANNENBERG, K.: Evaluation Criteria and Certificates (How) do they help to gain trust?, Sborník konference „Bezpečnost informačních systémů ve finančním sektoru", Bratislava , ISBN 80-85655-08-X
[11]STANEK, M.: Bezpečnostné problémy operačných systémov (ilustrované na príklade Windows NT), Sborník konference „Bezpečnost informačních systémů ve finančním sektoru", Bratislava , ISBN 80-85655-08-X

***Poznámka recenzenta:*** *Příspěvek se zabývá několika v celku nezávislými oblastmi. Podle mého názoru tato rozptýlenost snižuje jeho kvalitu, takže na to nejzajímavější - operace s metrikami nezbyl prostor. V příspěvku jsou rovněž použity symboly a termíny, které nejsou definovány ani vysvětleny, bez znalosti pramenů je pak příspěvek těžko pochopitelný.*