

u p o r a b n a
INFORMATIKA

1996

ŠTEVILKA 2

APR/MAJ/JUN

LETNIK IV

ISSN 1318-1882

Matematični model zavarovalnega procesa

Organizacijski vidiki varovanja informacijskih sistemov

Pojmovnik kakovosti

Spoštovani bralke in bralci,

pred nekaj dnevi je na Brdu potekalo pomembno posvetovanje na temo prilagajanja slovenske državne uprave in njene harmonizacije z Evropsko unijo. Srečanje so z naše strani organizirali Ministrstvo za finance, Ministrstvo za notranje zadeve ter Inštitut za javno upravo, sodelovale pa so tudi nekatere mednarodne organizacije, kot so OECD in SIGMA. Žal se srečanja nisem mogel udeležiti, imel sem pa priložnost pregledati nekaj gradiv in program srečanja. Kar me je pri vsem skupaj najbolj zbadlo v oči je, da informatizacija uprave in prenova njenega poslovanja nikjer nista bili niti omenjeni.

Analiza dogajanj v sodobnih upravnih sistemih kaže, da so le-ti pravzaprav povsod v tranziciji, le da so razlogi za spremembe od primera do primera nekoliko različni. Za upravo v centralno in vzhodno evropskih državah velja, da se mora najprej prilagoditi spremembam v političnih in ekonomskih sistemih teh držav. To pomeni, da se mora uprava spremeniti iz podaljšane roke partijskega aparata, ki prek inštrumentov centralno-planskega aparata dirigira poslovnim subjektom, v avtonomen in strokoven sistem, ki je sposoben ustvarjati čim bolj učinkovito ekonomsko-socialno okolje za razvoj tržnega gospodarstva in delovanje podjetij.

Vendar pregled dogajanj v zahodno evropskih državah kaže, da v teh upravnih sistemih prav tako vre, le da so razlogi za to nekoliko drugačni. Poleg procesov evropeizacije in globalizacije se v strokovnih in političnih krogih največ razpravlja o tem, kako upravo, ob enaki ali celo povišani ravni storitev, poceniti.

Kot posledica teh prizadevanj je nastala paradigma tako imenovane "vitke uprave" (v angleščini "lean administration" ali v nemščini "Schlanke Verwaltung"). Pri tem pa velja precej enotno prepričanje, da je večje uspehe na tem področju mogoče doseči le s prenovo poslovanja vseh segmentov javnega sektorja in z na njej temelječo informatizacijo.



Če nekoliko bolj kritično pogledamo dogajanje v slovenski državni upravi v zadnjih letih, lahko hitro ugotovimo, da v njej ni o prenovi samega poslovanja ne sluha ne duha, kljub globokim spremembam. Spremembe so doslej potekale predvsem iz političnih razlogov in po političnih linijah. Reorganizacije ministrstev so bile izpeljane zaradi prenosa pooblastil in centralizacije, ne poznamo pa primera, da bi bila doslej opravljena kakšna temeljita analiza poslovanja, bodisi v ministrstvih ali v lokalni upravi, ki bi lahko služila kot osnova za ustrezne organizacijske in tehnološke spremembe.

Informatizacija javne uprave je potekala do zdaj bolj ali manj stihijsko. Uvajanje novih tehnologij je bilo pogosto bolj posledica močnega pritiska trga informacijske tehnologije in storitev in le redko posledica sistematično in premišljeno postavljenih vsebinskih ciljev. Tehnična opremljenost naše državne uprave, če jo merimo po številu osebnih računalnikov na zaposlenega, ne zaostaja veliko za povprečjem zahodnoevropskih držav. Vendar je to precej varljiv kazalec. Površna analiza ravni uporabe novih tehnologij v upravnih organih na republiški in lokalni ravni pokaže, da je razpoložljiva oprema večinoma zelo slabo izkoriščena, upravno poslovanje pa se kljub široki uporabi najsodobnejše tehnologije v zadnjih dvajsetih letih ni bistveno spremenilo, niti tam, kjer bi se že zdavnaj lahko.

Tako kot v podjetjih, kjer se zadnji dve leti govori največ o tako imenovanem reinženiringu ali prenovi poslovnih procesov, se tudi v javni upravi in še posebno v državni upravi, postavlja vprašanje prenove upravno-administrativnega poslovanja. Hierarhične strukture, delitev in organizacija dela in pristojnosti, delovni procesi in postopki, ki so bili v preteklosti, v časih pretežno ročnega in papirnatega poslovanja funkcionalni in nujni, že dolgo zavirajo nadaljnji razvoj in jih bo potrebno popolnoma prenoviti, predvsem pa poenostaviti in poceniti.

Zato bi bilo potrebno v naših glavah, predvsem pa v ministrstvih najprej spremeniti prepričanje, da je za približevanje Evropi dovolj formalno-pravna prilagoditev državne uprave, pa bomo na konju. Če se ne bomo ob organizacijski in pravni reformi resno lotili tudi vsebin poslovanja ter prenove poslovnih procesov in postopkov, se lahko zgodi, da bo ta konj crknil, še preden bo od blizu zagledal vrata Evrope.

Mirko Vintar
glavni in odgovorni urednik

UVODNIK**AKTUALNO**

- 5** ■ ■ ■ ■ Nagovor predsednika republike Milana Kučana
na Dnevh slovenske informatike

STROKOVNE RAZPRAVE

- 7** ■ ■ ■ ■ *JULIJANA BIZJAK-MLAKAR:*
Matematični model zavarovalnega procesa
- 12** ■ ■ ■ ■ *NEVENKA GORENŠEK:*
Skupinska programska oprema in informacijski sistemi
- 21** ■ ■ ■ ■ *TOMAŽ POŠTUVAN:*
Organizacijski vidiki varovanja informacijskih sistemov

TERMINOLOGIJA

- 28** ■ ■ ■ ■ Pojmovnik kakovosti procesa razvoja programske opreme

POROČILA

- 42** ■ ■ ■ ■ Blejska konferenca o elektronskem poslovanju

KOLENDAR PRIREDITEV

- 43** ■ ■ ■ ■

Izid te revije so finančno podprli:

Revijo v tem letu sofinansira

Ministrstvo za znanost in tehnologijo

nagovor

PRESEDNIKA REPUBLIKE MILANA KUČANA

NA DNEVIH SLOVENSKE INFORMATIKE

Portorož, 17. april 1996

Spoštovane gospe in gospodje,

rad sem se odzval vabilu organizatorjev, da se udeležim tega vašega tretjega posvetovanja. To je priložnost, da se neposredno seznanim z zadnjimi dogajanjem v slovenski informatiki. Razumeli boste, da me posebej zanima obstoječa, mogoča in načrtovana povezava informacijske infrastrukture s procesi upravljanja in odločanja na državni in na podjetniški ravni a tudi na ravni strateškega odločanja o razvoju slovenske družbe. Brez kvalitetne informacije danes ni mogoča kvalitetna odločitev, ni mogoče videti njenih posledic in ne razumeti alternativ. Vse pa praviloma ne vodijo do smiselnega cilja in zato niso v oporo odgovornosti, ki je povezana z odločanjem. Informatika je v oporo profesionalnim odločitvam, a je v oporo tudi tistim, ki nadzirajo odločitve, njihovo strokovnost, smiselnost in utemeljenost. Z vso svojo razvito tehnologijo v bistvu omogoča demokratiziranje procesa odločanja. Zato je veliko več kot sredstvo za razkazovanje in dokazovanje naše sodobnosti - od najnovejših generacij osebnih računalnikov, ko naj bi PC imel vsak deveti Slovenec, do kar spoštovanja vredne informacijske tehnologije v podjetjih in v državni upravi, ampak predvsem sredstvo in dejavnost v funkciji učinkovitega delovanja javnega sektorja, pa seveda tudi hitre odzivnosti in prilagajanja slovenskih podjetij spremembam na domačih in tujih trgih.

Vsekakor pa se mi zdi prav poudariti pomen informacije za razumevanje in orientiranje upravljalcev države v razpotjih in dilemah globalnih družb, ki jih objektivno prav globalni trgi in univerzalnost tehnologij med seboj vse bolj povezujejo in postavljajo v medsebojno odvisnost. Ti

procesu terjajo oz. povzročajo ekonomsko pa tudi ekološko in vsestransko vse bolj povezan in soodvisen svet, v katerem posamezne države kot entitete spreminjajo svojo identiteto in izgubljajo njen del. Prav zato poskušajo čimbolj suvereno v novih razmerah uveljavljati in potrjevati svojo individualnost, različnost, svoj jezik, kulturo, svojo zgodovino. Biti globalen in biti hkrati svoj, individualen, je izziv in je priložnost za managerje sodobne države. Tudi slovenske. Če ne bomo znali ustvarjalno odgovoriti na ta izziv, najti svoj prostor znotraj trendov, ki vse bolj obvladujejo evropski in svetovni prostor, se kaj lahko zgodi, da bomo za svet zanimivi zgolj kot spomin na preteklost, ki je to postala, ker tisti, ki jim je to bilo v Sloveniji naloženo, niso razumeli znamenj svojega časa in ne znali obvladati sredstev, ki so jim za to bila na razpolago. Mislim pri tem posebej na informatiko. Nenazadnje naj bi bila informacijska tehnologija tudi sredstvo za kakovostnejše življenje posameznika v tržno socialni in ekološko ozaveščeni državi, kot Slovenijo radi razglajamo v promocijskih nastopih.

Umeščanje informacijske tehnologije v tak zgodovinsko razvojni kontekst ter takšno razvijanje in uporaba terjajo predvsem jasno predstavo o funkcijah slovenske države, spoznanje o njenih možnostih ter odgovornost nosilcev vodenja in upravljanja države.

Nedvomno smo uspešno sklenili prvi del zgodovinskega razvojnega projekta neodvisne in samostojne slovenske države, vendar zaostajamo pri koncipiranju nacionalnega razvojnega projekta, ki bi ga lahko poimenovali identiteta

države Slovenije. Govorim o projektu, ki bi iz različnih zornih kotov odgovarjal na vprašanja: kaj smo, kaj bi želeli biti, kaj bi mogli biti glede na svoje razvojne potenciale ter možnosti doma in v svetu, kaj bomo in kako bomo znotraj procesov, ki so zaznani, spoznani in obvladovani v svetu. Pred to nalogo so vse sodobne države, toda nanjo ne odgovarjajo in ne morejo odgovarjati vse na enak način. Morajo pa in morejo pri iskanju odgovorov uporabljati iste informacije, ako jih znajo in želijo uporabiti. Prek iskanja odgovorov na ta strateška vprašanja se učimo razumevati svojo prihodnost, tako kot na posameznih ravneh in parcialnih področjih že znamo delati in sprejemati tekoče odločitve. Govorim o konceptualni krizi, v kateri se Slovenija nahaja, o problemu, ki pa, ponavljam, ni lasten samo nam. Lahko bi rekli: če se ta hip Zahodna Evropa oz. Evropska Unija na eni strani ukvarja s problemi svoje prostorske in druge organizacije, se na drugi strani srednjeevropske in vzhodnoevropske države, ki želijo postati članice EU, bolj ali manj uspešno ukvarjajo s problemi svojih razvojnih konceptov in vizij. Iskanje te vrste konceptov je povezano z izzivi, ki jih prinaša novo tisočletje. Če to izrazim z besedami našega filozofa Hribarja, ki pravi: "Če je bilo 19. stoletje čas produkcije in naravoslovja, 20. pa čas organizacije in družboslovja, bo 21. stoletje čas orientacije in modroslovja (filozofije)".

Za iskanje smeri in vsebine nadaljnjega razvoja Slovenije so potrebne prave informacije; za njihovo pridobivanje in hiter prenos pa ustrezna, sodobna informacijska infrastruktura in mehanizmi, ki bodo onemogočali monopolizacijo. Kakšna ter kje in v kakšnih povezavah, na to predvsem boste - verjame - skušali odgovoriti na letošnjih Dnevih slovenske informatike. To so strokovna in politična vprašanja. Sam bi želel predvsem to, da v tem vozlišču potrebnih povezav in sodelovanja med nosilci informacijske infrastrukture in nosilci upravljanja slovenske družbe in države informacije in sodobna informacijska tehnologija ne bi bili izrabljeni za manipulacije oblasti z ljudmi, za zlorabe pri vplivanju na javnost v korist parcialnih političnih ali ideoloških opcij političnih strank, posebej ne v volil-

nem letu, ter za blatenje in poniževanje lastne države v svetu, kar ji jemlje potrebno verodostojnost in avtoriteto v partnerskem povezovanju in v dialogu o globalnih razpotjih, pred katerimi je človeštvo. Bile naj bi sredstvo za demokratizacijo odločanja na vseh ravneh in za uspešno pozicioniranje Slovenije v svetu; predvsem pozicioniranja Slovenije kot sodobne in podjetne evropske države in slovenskih podjetij kot gazel v poslovnem svetu tržne prodornosti in celovitega upravljanja kakovosti. Treba je le spremeniti razvojne probleme v izzive in priložnosti za dobro pripravljene in visoko motivirane strokovnjake, managerje, podjetnike in politike. Brez tega se sicer uspavamo na pohvalah, dodamo še kaj samohvale, a to dejstev resničnega življenja ne more spremeniti.

Državni zbor Slovenije je prejšnji teden sprejel stališča in sklepe o odnosih RS z EU, Italijo in NATO-m. Ena ključnih razvojnih orientacij, proevropska, je tako v parlamentu sprejeta. Sedaj jo je treba operacionalizirati tudi na področju informatike, tako na vladni ravni in v posameznih ministrstvih kot v državni upravi in v javnem sektorju nasploh. To delo je potrebno voditi in usklajevati, postaviti mu jasn cilj in določiti skupno voljo. Ni ugledno, če prihajamo pred strokovnjake iz evropskih institucij nepripravljeni in na državni ravni med seboj neuskklajeni. To ni pravi način za utrjevanje vtisa o solidni državi, ki ima jasne cilje in jasne predstave o sredstvih, s katerimi jih je mogoče uresničiti. Imamo tudi dejavnosti, službe in stroke, ki so na evropski ravni v vsakem pogledu, kot je n.pr. slovenska statistika. A tudi te ne morejo dati željenega, še manj optimalnega učinka, če je okolje, v katerem deluje statistika, iz katerega zajema in mu služi, neprilagojeno sodobni statistični informaciji.

Nalog je torej dovolj za vse, ki se specializirano in z različnih vidikov ukvarjate z informatiko. Zato vam tembolj želim delovno uspešen posvet in tudi vam, informatikom, izmenjavo za vas aktualnih informacij.

MATEMATIČNI MODEL ZAVAROVALNEGA PROCESA

Julijana Bizjak-Mlakar

Povzetek

Modeliranje finančnih in denarnih tokov v zavarovalnicah je ena izmed zahtevnejših nalog, ki jo v zavarovalnicah opravljajo zavarovalniški matematiki ali aktuarji*. Spremenljivke, ki nastopajo v takšnih modelih, so večinoma slučajne spremenljivke. Pri sestavi modela za škodni proces si aktuarji pogosto pomagajo s simulacijo sestavljene Poissonove porazdelitve. Simulacija je lahko zelo zapletena, če uporabljamo klasične numerične metode. Še posebno zahtevna in problematična je tedaj, ko želimo v model vključiti številne predpostavke, simulirati stroške, inflacijo, donose od naložb, vključiti dinamično kontrolo ali razširiti model na obdobje več let. V članku, namesto klasičnega pristopa, celotni zavarovalni proces razstavimo na posamezne procese, ki so zato lažje obvladljivi. Posamezno realizacijo izvedemo z generatorjem slučajnih števil. Uporabnost modela povečamo tako, da v model vključimo še periodičnost finančnih oziroma denarnih tokov.

Abstract

Modeling of financial and cash flow in insurance is one of the exacting tasks performed in insurance by actuaries. Variables in such models are mostly random variables. When preparing a model of claim process, actuaries frequently use a simulation of compound Poisson distribution. The simulation can be very complex, when using the classical methods. It might be particularly difficult and problematic in case we want to include into the model various suppositions, simulation of costs, inflation, investment return, including dynamic control or enhancing the model to a period of many years. In the paper, instead of the classical process, we divide the insurance process into individual processes which are thus easier to control. Individual calculations are made by the generator of random processes. The usefulness of the model might be enlarged by including periodical nature of financial and cash flows.



1. Vloga zavarovalstva pri nas

Razvoj in vloga zavarovalstva v posameznih državah merimo z deležem zavarovalne premije v bruto domačem proizvodu, zavarovalni standard pa z zneskom zavarovalne premije na prebivalca. Slovenija po deležu zavarovalne premije v bruto domačem proizvodu pa tudi po zavarovalnem standardu zaostaja za razvitimi evropskimi državami.

Eden izmed vzrokov za tako stanje je v sistemu socialne varnosti, ki je bil že v bivši Jugoslaviji skoraj povsem v rokah države. Sredstva za zdravstveno, pokojninsko in invalidsko zavarovanje, s katerimi je država skrbela za socialno varnost državljanov, so se zbirala s pomočjo obveznih stopenj od dohodkov v skladu z vsakokrat veljavno zakonodajo. Tudi pri varstvu pred naravnimi in drugimi katastrofalnimi škodami, razen

pri požarnih zavarovanjih, zavarovalnice niso imele velike vloge. Sistem solidarnosti tudi danes predstavlja pomemben vir sredstev pri odpravljanju posledic naravnih nesreč. Država s svojo davčno politiko, posojili oziroma namenskimi subvencijami, ne vpliva vzpodbudno na obseg zavarovanosti pred naravnimi nesrečami. V Ustavi Republike Slovenije je določeno, da imajo državljani pravico do socialne varnosti v skladu z zakonom. Država ureja obvezno zdravstveno, pokojninsko, invalidsko in druge oblike obveznega socialnega zavarovanja. Uživalcem pravic iz pokojninskega in invalidskega zavarovanja je trenutno v primerjavi s plačami zaposlenih zagotovljena sorazmerno visoka raven pokojnin in drugih pravic iz pokojninskega in invalidskega zavarovanja. Nizka povprečna pokojninska

* Aktuarji izračunavajo s pomočjo statističnih in matematičnih metod, iz zbranih statističnih podatkov ter na podlagi delovanja zakona velikih števil del skupne škode, ki odpade na posamezen nevarnostni objekt, ki ga je zavarovalnica prevzela v zavarovanje. Vloga aktuarja in pooblaščenega aktuarja je opredeljena tudi z zakonom. Pooblaščen aktuar potrjuje skladnost podatkov v letnih računovodskih izkazih z zavarovalnimi računovodskimi standardi in pravilnost obračunov v letnih računovodskih izkazih za oblikovanje zavarovalnotehničnih rezervacij in rezerv.

doba zavarovancev in neugodno razmerje med številom uživalcev pravic iz zavarovanja in aktivnimi zavarovanci se odraža v vse težjem zagotavljanju sredstev za izplačevanje že pridobljenih pravic iz pokojninskega in invalidskega zavarovanja. Višina prispevkov in zagotovljene pravice namreč niso odraz aktuarskih izračunov, temveč se prispevki vsakokrat obračunavajo glede na razmerje med številom upokojujencev in aktivnimi zavarovanci. Tudi zato lahko v prihodnje pričakujemo spremembe v višini zagotovljene socialne varnosti naših državljanov. Uvedba prostovoljnih zavarovanj se kaže kot nujnost za razrešitev težav, v katerih se je znašel sistem socialnega varstva.

Primeren sistem prostovoljnega zavarovanja je možno vzpostaviti le v primeru strokovno ustrezno pripravljenih zavarovalnih podlag (zavarovalni pogoji, premijski sistemi), učinkovitega državnega nadzora nad izvajanjem zavarovanj, vključitve davčnih vzpodbud za vključevanje v prostovoljno zavarovanje, in seveda ob vzpostavitvi takšnega gmotnega standarda državljanov, ki prebivalstvu omogoča hranjenje denarnih sredstev.

Razvite evropske države so velik del skrbi za socialno varnost prevalele neposredno na državljanke. Zato imajo prostovoljna zavarovanja v teh državah večji pomen na področju socialne varnosti državljanov kot pri nas.

Visoka inflacija je v preteklih obdobjih negativno vplivala na razvoj zavarovalnic. Njeni negativni učinki so se odražali v razvrednotenju zavarovanj, zato so še posebej prizadeli tiste zavarovance, ki so sklenili življenjska zavarovanja. Zavarovalnice lahko učinke inflacije omilijo, če sredstva ustrezno nalagajo. Zaradi odsotnosti tržne ekonomije, naše zavarovalnice sredstev niso dobro nalagale. Uspeh nalaganja sredstev je namreč odvisen od gospodarskega in političnega stanja v državi.

V nasprotju z razvitimi državami, kjer je polovica ali celo več zavarovalne premije zbrane iz raznih oblik življenjskih zavarovanj, se je pri nas ta delež šele v letu 1993 približal 10%, pred tem pa je bil vrsto let znatno nižji. Različne oblike zavarovanja premoženja zavzemajo največji delež v strukturi zavarovalne premije zavarovalnic.

Zakonodaja, ki ureja področje zavarovalstva, se v letih po osamosvojitvi Slovenije postopoma sprejema in vnaša novosti na to področje. Država si strokovnega in finančnega nadzora nad izvajanjem zavarovanj še ni povsem zagotovila, vendar so v teku postopki, ki vodijo v to smer. Zaradi pomanjkanja zavarovalnega nadzora imajo zavarovalnice možnost izkazovanja nižjih potrebnih višin sredstev rezervacij, kot jih ugotavljajo aktuarski izračuni. To lahko vodi v podcenjenost teh sredstev in v prikrito nesolventnost zavarovalnic. Ko se pojavijo likvidnostne težave zavarovalnic kot posledici

ca nesolventnosti, je zaradi narave zavarovalnega posla običajno že prepozno za uspešno saniranje stanja. Zavarovalnica je namreč lahko likvidna kljub nesolventnosti. Glede na višine oblikovanih zavarovalnotehničnih rezervacij in značilnega škodnega poteka pri nekaterih vrstah zavarovanj, predvidevanje, da so zavarovalnotehnične rezervacije pri večini naših zavarovalnic prenizke, ni nerealno. Poleg tega so celo prikazani relativni merodajni zavarovalnotehnični rezultati slovenskih zavarovalnic pri zavarovanju avtomobilske odgovornosti že od leta 1991 negativni, kar ni nezamisljivo, ker predstavljajo škode teh zavarovanj okoli 30% vseh škod, ki jih zavarovalnice letno obračunajo.

Potreba po poglobljenem spoznavanju zavarovalnega procesa izvira iz pomena tega področja gospodarjenja za vso državo in zaradi nastalih sprememb ter težav na področju zavarovalstva.

2. Matematični model zavarovalnega procesa

Stanje sredstev zavarovalnice je z ozirom na to, kako se to stanje zvišuje oziroma znižuje, pogosto predstavljeno kot vodni rezervoar, ki se polni s pritokom in prazni z odtokom sredstev. Ta primerjava v celoti ne ustreza, saj se stanje sredstev zavarovalnice lahko spreminja tudi zaradi sprememb tržnih vrednosti sredstev.

Finančni oziroma denarni tokovi so v zavarovalstvu odvisni od spremenljivk, ki so večinoma slučajne spremenljivke. Za opredelitev teh tokov je potrebno dobro poznavanje zavarovalno matematične oziroma aktuarske teorije.

Nekateri tokovi nastanejo zaradi pozavarovalne dejavnosti. Tem se v tem članku ne bomo posebej posvetili. Zato iz vseh tokov, ki jih v članku obravnavamo, najprej izvzamemo tokove, ki nastanejo zaradi pozavarovanja in spremljamo le ostale tokove zavarovalnega procesa.

V nadaljevanju opisujemo matematični model zavarovalnega procesa, postavljen na temeljih aktuarskega izračuna. Podrobnosti aktuarskega izračuna, sestave delnih modelov in uporabe modela v tem članku ne obravnavamo, ker bi sicer presegli zamišljeni obseg članka. Slučajnostne procese, ki nastopajo v večini postavk zaporedja prehodnih stanj, zajamemo v enačbo prehoda, ki ponazarja tok sredstev zavarovalnice v posameznih časovnih trenutkih. Pri sestavi modela uporabljamo spoznanja statistike, verjetnostne teorije in zavarovalno matematične teorije.

Pogosto se v praksi pri sestavljanju matematičnih modelov finančnih oziroma denarnih procesov pojavlja dilema izbire med stohastičnimi in determinističnimi matematičnimi modeli. Le-ta se, zaradi preprostejšje sestave stohastičnih modelov, nagiba v smer izbire stohastičnih modelov. Številne spremenljivke in medsebojne

odvisnosti med spremenljivkami lahko namreč povzročijo izjemno zapletenost determinističnega modela.

Kadar opazujemo le denarne tokove zavarovalnice, opazujemo namesto prihodkov oziroma odhodkov zavarovalnice prejemke oziroma izdatke. Z besedo prejemki označujemo neposredna povečanja denarnih sredstev, z besedo izdatki pa neposredna zmanjšanja denarnih sredstev. Med prejemke sodijo: plačane zavarovalne premije, prejemki v obliki denarnih sredstev iz investicijske dejavnosti oziroma od naložb (prejemki od odtujitve sredstev in naložb, prejemki od obresti od naložb, prejemki od najemnin, prejemki od deležev iz dobička drugih...) in prejemki pri dejavnosti financiranja od vplačanega kapitala ali od dobljenih posojil. Izdatki pa so: izdatki za plačane škode, ki lahko vsebujejo tudi cenilne stroške, izdatki za izplačilo dividend in drugih deležev lastnikov iz dobička oziroma bonusov iz sklenjenih zavarovanj ter izdatki za stroške obratovanja, kamor lahko vključimo tudi izdatke za davke in za preventivno dejavnost, izdatke pri investicijski dejavnosti in izdatke pri dejavnosti financiranja za vračila kapitala in posojil.

Za ugotavljanje finančne moči zavarovalnice, za celoten sklop investicijske dejavnosti in za različne aplikacije je pomembna vrednost sredstev in ne le stanje denarnih sredstev. Z modelom, ki ga obravnavamo v tem članku, ne bomo spremljali le prejemkov in izdatkov denarnih sredstev, temveč vključujemo v model takšne tokove sredstev, da nam iskani presežek pritoka sredstev nad odtokom pomeni možna sredstva za nove investicije. V model zato vključujemo celotno obračunano zavarovalno premijo v obračunskem obdobju, čeprav se del te ob koncu obračunskega obdobja nahaja v obliki terjatev do zavarovalcev, do zavarovalnih zastopnikov itd. Vedeti pa moramo, da je del teh sredstev v obliki terjatev oziroma nelikviden in brez donosa. Podobno zajamemo v model obračunane in ne le plačane škode obračunskega obdobja.

Matematični model, s katerim iščemo sredstva, ki jih je potencialno možno ob koncu obračunskega obdobja investirati, zapišemo z enačbo prehoda takole:

$$A(t) = A(t-1) + ZP_o(t) + D(t) + K(t) - X_o(t) - St(t) - DI(t) \quad 1$$

Pri tem pomenijo:

- $A(t-1)$ je stanje sredstev ob koncu obračunskega obdobja $t-1$;
- $ZP_o(t)$ je obračunana zavarovalna premija v obračunskem letu t ;
- $D(t)$ je prihodek denarnih sredstev v obračunskem letu t od sredstev A ;
- $K(t)$ so pritoki pri financiranju zaradi povečanih sredstev v obračunskem obdobju t s povečanjem kapitala ali dolgov. Pri simulacijah zavarovalnega procesa

lahko privzamemo, da v obračunskem obdobju ni bilo novega financiranja, torej, da je $K=0$ za vsak t .

- $X_o(t)$ je znesek škod, obračunanih v obračunskem obdobju t ;
- $St(t)$ so stroški obratovanja v obračunskem obdobju t , kamor so lahko vključeni tudi odhodki za davke, sicer pa so vanje vključeni odhodki za preventivno dejavnost, odhodki pri investicijski dejavnosti in odhodki pri dejavnosti financiranja za vračila kapitala in posojil;
- DI so dividende in drugi deleži lastnikov iz dobička oziroma bonusi iz sklenjenih zavarovanj v obračunskem obdobju t .

Za zneska škod in premij veljata naslednji obračunski enačbi:

$$X(t) = X_o(t) + C(t) - C(t-1) \quad 2$$

$$ZP(t) = ZP_o(t) - PP(t) + PP(t-1). \quad 3$$

Pomen oznak je:

- $ZP(t)$ je zavarovalna premija, ki se nanaša na obračunsko obdobje t ;
- $X(t)$ je znesek škod, ki se nanašajo na obračunsko leto t ;
- $C(t)$ je znesek, ki ga ob koncu obračunskega leta t rezerviramo za škode, nastale v letu t ali prej, ki do konca obračunskega leta t še niso obračunane. Ocenjeni znesek C za nastale, nerešene škode, imenujemo škodne rezervacije.
- $PP(t)$ je del zavarovalne premije, obračunane v obračunskem obdobju t , pri kateri se jamstvo zavarovalnice nanaša na prihodnja obračunska obdobja $t+1$, $t+2, \dots$. Za zavarovalno premijo PP uporabljamo izraz prenosna premija.

Zapišimo še model obračunske enačbe za sredstva solventnosti, ki jih v našem modelu definiramo kot presežek sredstev v izkazu stanja nad obveznostmi iz naslova zavarovalnotehničnih rezervacij in drugimi obveznostmi. Med obveznostmi pa ne bomo prištevali obveznosti do delničarjev, torej lastnega kapitala in rezerv ter dobička poslovnega leta. V tuji literaturi se za ta sredstva, ki jih bomo označili s črko U , uporabljajo tudi izrazi: risk reserve, capital at risk, equalization reserve, adjustment reserve.

$$U(t) = U(t-1) + ZP(t) + D(t) + \Delta A(t) + K(t) - X(t) - St(t) - DI(t) \quad 4$$

Spremenljivki U in ΔA pomenita:

- $U(t-1)$ sredstva solventnosti ob koncu obračunskega obdobja $t-1$;
- $\Delta A(t)$ sprememba vrednosti sredstev A v obračunskem obdobju t .

Modeliranja zavarovalnih procesov, zapisanih z enačbami 1, 2, 3 in 4 se lotimo tako, da najprej sestavimo stohastični model za škode $X(t)$. V primeru veljavnosti določenih pogojev (Hipp, Michel, 1990) lahko predpostavimo, da je število škod $n(t)$, nastalih v letu t , porazdeljeno po Poissonovi porazdelitveni funkciji. Nadalje predpostavimo, da so zneski škod X_i , ki sestavljajo spremenljivko X , neodvisni in identično porazdeljeni po porazdelitvi $S(u) = P(X_i \leq u)$. Porazdelitev spremenljivke X imenujemo tedaj sestavljena Poissonova porazdelitev.

Sestavljeno Poissonovo slučajno spremenljivko X generiramo z WH generatorjem (Daykin, Pentikäinen, Pesonen, 1994). Nato simuliramo eno izmed aktuarskih metod izračunavanja škodnih rezervacij npr. metodo verižnih količnikov in odtod izračunamo vrednosti X_p . Zneske ZP izračunamo po aktuarskih postopkih iz generiranih podatkov za zneske škod. Najprej določimo pričakovani znesek škod $E[X(t)]$, ki ga povišamo z ustreznim varnostnim dodatkom oziroma na račun stroškov. Delež prenosne premije ocenimo na osnovi znanih statističnih podatkov zavarovalnic, nato pa iz obračunske enačbe 3 izračunamo zneske ZP_p . S pomočjo zbranih statističnih podatkov, z ocenami na podlagi izkušenj ali z uporabo avtoregresivnih procesov (Montgomery, Johnson, 1976) oblikujemo še preostale delne modele zavarovalnega procesa, ki sestavljajo enačbo 1.

Pri oblikovanju modela zavarovalnega procesa ni možno spregledati vpliva, ki ga ima na posamezne tokove sredstev inflacija.

Gibanje inflacije je pogosto povezano z ekonomskimi cikli oziroma s stabilnostjo ali nestabilnostjo gospodarskih sistemov. Zavarovalno kritje in s tem posledično višine zneskov obračunanih škod se v času od nastanka zavarovalnega primera ali pa do izplačila škod lahko spreminjajo v odvisnosti od vsebine posameznih zavarovalnih pogodb oziroma načina zavarovanja. Vpliv inflacije je drugačen pri vsotnih zavarovanjih, drugačen pa pri škodnih zavarovanjih in v okviru škodnih zavarovanj drugačen pri zavarovanju polne vrednosti oziroma pri zavarovanju prvega rizika. Stopnja spremembe višin zneska vseh škod zaradi spremembe višine posameznega zneska škod v nekem obdobju imenujemo inflacija škod. Inflacija škod torej ni nujno enaka inflaciji, merjeni s cenami na drobno.

O "inflaciji stroškov" govorimo, ko se ti v času spreminjajo zaradi rasti cen posameznih stroškov. Rast plač poleg rasti stroškov ogrevanja, pisarniške opreme, računalniške opreme in rasti drugih stroškov, vpliva na rast zneska za pokrivanje stroškov obratovanja zavarovalnice in na rast cenilnih stroškov, ki jih v našem modelu obravnavamo skupaj z ostalimi stroški zavarovalnice. Kumulativni indeks cen na drobno se v stabilnem gospodarstvu običajno dolgoročno ujame s kumulativnim indeksom rasti plač.

Ker vpliva inflacija na stroške in zneske škod, je neizogibno pri izračunu zavarovalne premije upoštevati tudi vpliv inflacije. V praksi se negativnim vplivom inflacije izogibamo na različne načine. Zneske zavarovalne premije prilagajamo povečanim obveznostim zavarovalnice s pomočjo obračunavanja enoletnih premij pri večletnih zavarovanjih in pri zavarovanjih z nedoločenim rokom trajanja. Tak način obračunavanja zavarovalne premije je pri nas uveden pri večini zavarovalnih vrst premoženjskih zavarovanj. Ob zapadlosti premij v plačilo se lahko zavarovanje preuredi. Premija se pri nekaterih zavarovalnih vrstah prilagaja spremembam cen tako, da se že ob sklenitvi zavarovanja oziroma ob preureditvi ali obnovitvi obračuna doplačilo na zavarovalno premijo glede na ocenjeno stopnjo bodoče inflacije. Za kritje povečanih obveznosti zaradi inflacije je seveda tudi možno uporabiti sistem naknadnih plačil zavarovalne premije ob koncu obračunskega obdobja oziroma zavarovalnega leta.

Premijo izračunavamo s pomočjo preteklih in sedanjih podatkov o škodah in o porabljenih stroških, ki jih prevedemo na sedanje vrednosti, ter s pomočjo predvidenega gibanja škodnih procesov in potrebnih stroškov v prihodnosti. To je razlog, da so nepričakovane oziroma nenadne spremembe v inflaciji škod in stroškov upoštevane šele po določenem časovnem zamiku.

Inflacija lahko vpliva tudi na število škod. V Sloveniji je npr. visoka inflacija v letu 1989 in kasneje povzročila drastično povečanje števila škod iz kreditnih zavarovanj. Razlogi neplačevanja kreditov, kar je za zavarovalnico pomenilo uresničitev škodnih dogodkov, so bili večinoma ekonomski. Anuitete kreditnih pogodb, ki so jih sklenili posamezniki z bankami, zavarovalnica pa je te pogodbe zavarovala, so se namreč poviševale s stopnjami inflacije in z realno obrestno stopnjo, rast osebnih dohodkov kreditojemalcev pa je zaostajala za rastjo anuitet. To je bil razlog, da kreditojemalci svojih obveznosti niso mogli poravnati. Število škod se lahko poveča tudi v primerih, če so franšize določene v absolutnih zneskih, ki se v času trajanja zavarovanja oziroma do obračuna škode ne spreminjajo z inflacijo škod.

Višine donosov od naložb so ozko povezane z inflacijo, merjeno s cenami na drobno. Ta odnos se v praksi mnogokrat izraža tudi tako, da se pri donosih ločeno obravnavata revalorizacijski in realni del donosov.

Inflacijo, v kolikor imamo na voljo dovolj primernih podatkov, vključimo v naš model v obliki ustreznega avtoregresivnega procesa.

3. Periodična nihanja tokov sredstev

Za odločitve glede nalaganja sredstev je pomembna dinamika pritoka in odtoka sredstev znotraj enega ali

več poslovnih let. Pogosto lahko pri tokovih sredstev zavarovalnic zasledimo periodična nihanja oziroma sezonske sestavine. Model bo uporaben le, če bo vanj vključena dinamika pritoka in odtoka sredstev.

Dinamiko pojavov pri posameznih vrstah sredstev analiziramo tako, da opazujemo vrednosti členov v časovnih vrstah in iščemo zakonitosti tega spreminjanja. Preden uporabimo kako izmed metod analize časovnih vrst, preuredimo podatke tako, da so med seboj primerljivi. Primerljivost podatkov zagotovimo s tem, da poskrbimo za enakost razmikov med posameznimi členi vrste, odstranimo vpliv inflacije, sprememb v strukturi posla in drugo. Na časovni vrsti lahko opazujemo osnovno smer razvoja (trend), spremembe, ki izvirajo iz dolgoročnih vzrokov (ciklična nihanja) in spremembe, ki se ponavljajo na stalno razdobje (periodično nihanje). Na vsaki časovni vrsti lahko opazimo tudi slučajna nihanja, ki so izraz manjših vzrokov, ki so stalno prisotni. Iregularnih vplivov tukaj ne upoštevamo.

Znotraj poslovnega leta se posvetimo periodičnim nihanjem pri pritoku in odtoku sredstev s periodami nekaj mesecev. Sezonske oziroma periodične sestavine v časovnih vrstah iščemo na več načinov. Glede na pogoje, ki jim zadošča posamezna časovna vrsta, izberemo ustrezno metodo. Najprej določimo trend z metodo vsot, metodo popravljenih vsot, metodo koeficientov dinamike ali z drugimi metodami. Z metodo kvocienotov na drseče sredine (Blejec, 1973) analiziramo statistično periodično sestavino. Kot rezultat dobimo povprečne sezonske indekse $P_s = 100 \cdot (1 + p_s)$, $s = 1, 2, \dots, S$. Koeficient $1 + p_s$ imenujemo sezonska sestavina.

Predpostavimo periodo enega leta in S korakov ene periode. Naj bo $s = 1, 2, \dots, S$. Sezonske sestavine px_s odhodkov za škode vključimo v model tokov sredstev zavarovalnice. Naj bo $n(v, v+d_s)$ Poissonov parameter za zneske škod, ki so nastale v letu v , obračun zanje pa je narejen v koraku s leta $v+d$. Če je $n(v)$ Poissonov parameter, $k(d)$ pa verjetnost, da bo škoda, ki je nastala v letu v obračunana v letu $v+d$, lahko Poissonov parameter $n(v, v+d_s)$ izrazimo:

$$n(v, v+d_s) = \frac{k(d)}{S} \cdot n(v) \cdot (1 + px_s). \quad 5$$

Z generiranimi zneski škod opravimo aktuarski izračun potrebne zavarovalne premije. Podobno vključimo v modela premije oziroma stroškov ustrezne sezonske sestavine.

4. Uporaba modela

Matematični model zavarovalnega procesa je primeren za testiranje različnih predpostavk pri aktuarski matematiki, pri izračunavanju zavarovalne premije, stroškov, ugotavljanju potrebnih sredstev škodnih rezervacij, pri določanju potrebnega varnostnega dodatka in

pri odločanju glede pozavarovanja. Uporaben je tudi pri poslovnih odločitvah glede investiranja in pri nadzoru stanja sredstev zavarovalnic in pozavarovalnic. Model pomeni osnovo za simulacije zadostnosti sredstev rezervacij, simulacije sredstev solventnosti in izračunavanju verjetnosti nastanka nesolventnosti zavarovalnic in pozavarovalnic. Simulacije denarnih tokov zavarovalnic in pozavarovalnic so v pomoč pri odločitvah o višini prostih sredstev in višinah sredstev, ki jih lahko v posameznih obdobjih dolgoročno oziroma kratkoročno naložimo.

Z modelom lahko raziskujemo učinek inflacije na zavarovalni proces, učinek sprememb na trgu kapitala, učinke zaradi ekonomskih ciklov, učinke sprememb v velikosti portfelja in učinke, ki bi jih povzročili morebitni katastrofalni dogodki. Model je pomembno orodje za razumevanje medsebojnih odnosov med elementi, ki nastopajo v zavarovalnem procesu.

Glede na namen modela je potrebno ustrezne delne modele zavarovalnega procesa sestaviti v enoten model. Model je uporaben v neki konkretni zavarovalnici le, v kolikor začetne vrednosti in množica dopustnih rešitev odražajo resnično stanje zavarovalnega procesa v tej zavarovalnici.

Literatura:

- Beard R. E., Pentikäinen T., Pesonen E.: *Risk Theory, the Stochastic Basis of Insurance*. Cambridge: Chapman & Hall, 1984. 409 str.
- Blejec M.: *Statistične metode za ekonomiste*. Ljubljana: Univerzitetna založba, 1973. 868 str.
- Bogataj M. in Bogataj L.: *Inventory Systems Optimization for Dynamic Stochastic and Periodical Demand*. Amsterdam: Elsevier Science Publishers B. V., 1990. str. 295-299.
- Bülman H.: *Mathematical Method in Risk Theory*. Berlin: Springer-Verlag, 1970. 210 str.
- Daykin C. D., Pentikäinen T., Pesonen M.: *Practical Risk Theory for Actuaries*. London: Chapman & Hall, 1994. 546 str.
- Flis S.: *Zbrani spisi o zavarovanju, I. Knjiga*. Ljubljana: Pozavarovalnica Sava d.d. Ljubljana, Zavarovalnica Triglav d.d. Ljubljana, 1995. 311 str.
- Hipp C. in Michel R.: *Risikotheorie, Stochastische Modelle und Statistische Methoden, Schriftenreihe Angewandte Versicherungs-mathematik, Heft 24*. Karlsruhe: Verlag Versicherungswirtschaft e.V., 1990. 198 str.
- Montgomery D. C., Johnson L. A.: *Forecasting and Time Series Analysis*. New York: McGraw-Hill Book Company, 1976. 304 str.

Viri:

Bilančni podatki Zavarovalne družbe Adriatic d.d., Zavarovalniške hiše Slovenica, Zavarovalnice Maribor d.d. in Zavarovalnice Tilia d.d. za leto 1993. Interna gradiva slovenskih zavarovalnic.

Bilančni podatki Zavarovalnice Triglav d.d. Ljubljana za leta 1991, 1992, 1993, 1994. Interna gradiva Zavarovalnice Triglav d.d., Ljubljana.

Predlog zakona o prostovoljnem in invalidskem zavarovanju, Ministrstva za delo, družino in socialne zadeve. Ljubljana: Ministrstvo za delo, družino in socialne zadeve, 29. 6. 1994.

Slovenski računovodski standardi. Ljubljana: Zveza računovodij, finančnikov in revizorjev Slovenije, 1993. 209 str.

Statistični in zavarovalnotehnični rezultati slovenskih zavarovalnic pri zavarovanju avtomobilske odgovornosti za leta 1991, 1992, 1993 in 1994. Interna gradiva slovenskih zavarovalnic.

Mag. Julijana Bizjak-Mlakar se ukvarja z aktuarstvom od leta 1990. Kot samostojni aktuar je zaposlena na Zavarovalnici Triglav d.d. Visoko strokovno izobrazbo in naziv profesor matematike si je pridobila 1981 leta na fakulteti za naravoslovje in tehnologijo v Ljubljani. Pred tem je bila zaposlena v pedagoškem poklicu, kjer je nekaj let opravljala tudi vodilne delovne naloge. Na Ekonomski fakulteti v Ljubljani je leta 1996 dosegla strokovni naziv magister poslovne politike in organizacije z nalogo Matematični model optimalnega upravljanja negotovih denarnih tokov zavarovalnic. Njeno večletno strokovno izobraževanje iz aktuarstva vključuje tudi šestmesečno strokovno izpopolnjevanje v nemških zavarovalnicah. Z Ekonomsko fakulteto v Ljubljani sodeluje že dalj časa kot raziskovalec za področje zavarovalništva.

SKUPINSKA PROGRAMSKA OPREMA IN INFORMACIJSKI SISTEMI

Nevenka Gorenšček

Povzetek

Tehnologije na katerih temelji skupinska programska oprema (groupware) so med najhitreje razvijajočimi se tehnologijami. V mnogih delovnih sredinah se odločajo za izdelke skupinske programske opreme, da se lahko hitreje odzivajo na zahteve strank, izboljšajo produktivnost in uporabijo učinkoviteje razpoložljive informacije. Pri razmisleku o vsebini področja in koristih, ki jih od te opreme pričakujemo, nam lahko pomaga razumevanje pojmov, ki jih srečujemo v zvezi s skupinsko programsko opremo ter natančnejša definicija in kategorizacija izdelkov, ki jih vse 'mečemo v koš' skupinske programske opreme.

Pomembna je povezava s 'klasičnimi' računalniškimi aplikacijami v podjetjih, saj mnogokrat isti uporabnik potrebuje oboje. Rešitev skupinske programske opreme, ki ustreza nekemu oddelku, ni nujno ustrezna tudi za upravljanje dela v podjetju in obratno. Aberdeen Group pa govori o novi kategoriji skupinske programske opreme, ki jo imenuje 'Enterprise Groupware', skupinska programska oprema podjetja. Le-ta temelji na distribuiranih objektih in se vključuje v vse nivoje aktivnosti podjetja. Je fleksibilna, uporabna je tako za razvijalce, uporabnike in seveda upravljalce.

Abstract

Workgroup computing represents a convergence of information technologies and services, from hardware to telecommunications. Many organisations are now leveraging workgroup computing in order to become more responsive to customers, to increase the efficiency of business processes, and to utilize information more effectively. Customers have to understand the terms related to workgroup computing so as to be able to decide where and how they can implement its benefits. More precise definition of workgroup computing and product categories can help with this.

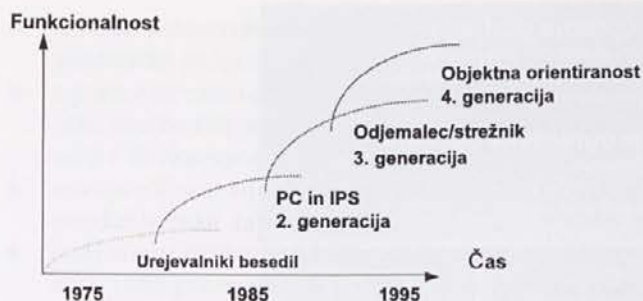
Groupware products do not usually satisfy everyone in the company. Aberdeen Group has identified a new category of collaborative software, called Enterprise Groupware, based on distributed objects. This technology is an integration framework that encompasses disparate clients, servers, and applications. Corporate MIS, developers and end-users benefit from it.



NEKAJ ZGODOVINE

Skupinska programska oprema ni nekaj novega. Mnoga orodja in tehnologije, ki jih razvrščamo pod ta pojem, poznamo že dobrih 20 let. Gartner Group postavlja

začetek avtomatizacije pisarniškega poslovanja, kakor tudi včasih poimenujemo funkcionalnosti in orodja, ki so nam v pomoč pri nalogah posameznih delovnih



Vir: Gartner Group on Office Evolution

Slika 1. Razvoj sistemov pisarniškega poslovanja

sredin, v sedemdeseta leta. Osnovni pripomoček je bil urejevalnik besedil. Razvoj v osemdesetih letih je šel v dve smeri: na eni strani so se uporabniki pogosto odločali za rešitve integriranega pisarniškega poslovanja na mini sistemih, drugi so izbrali pot osebnih računalnikov, najprej kot samostojno delovno mesto in kasneje mogoče povezanih v lokalno omrežje. Skupinska programska oprema v teh letih ni prav zaživela, saj potrebna mrežna infrastruktura še ni bila razvita. Sedaj je mrežna infrastruktura tu in v devetdesetih letih sta obe smeri ubrali pot tretje generacije okolja odjemalec-strežnik.

Naslednja, četrta generacija pisarniških sistemov oz. sistemov skupinske programske opreme, objektno orientirana ogrodja, prihaja na trg in obljublja prednosti objektno orientiranosti tudi v okolju skupinske programske opreme.

SKUPINSKA PROGRAMSKA OPREMA

Z izrazom skupinska programska oprema (groupware) označujemo skupino tehnologij, ki omogočajo medse-

bojno sodelovanje uporabnikov s pomočjo računalnikov. Obstaja več sorodnih interpretacij tega izraza, največkrat vključujejo tudi pojem povečane produktivnosti oziroma funkcionalnosti procesov, v katerih sodeluje več oseb. Skupinska programska oprema naj bi omogočala lažje sodelovanje, boljšo komunikacijo in večjo produktivnost z avtomatizacijo mnogih posameznih nalog in povečano učinkovitostjo drugih.

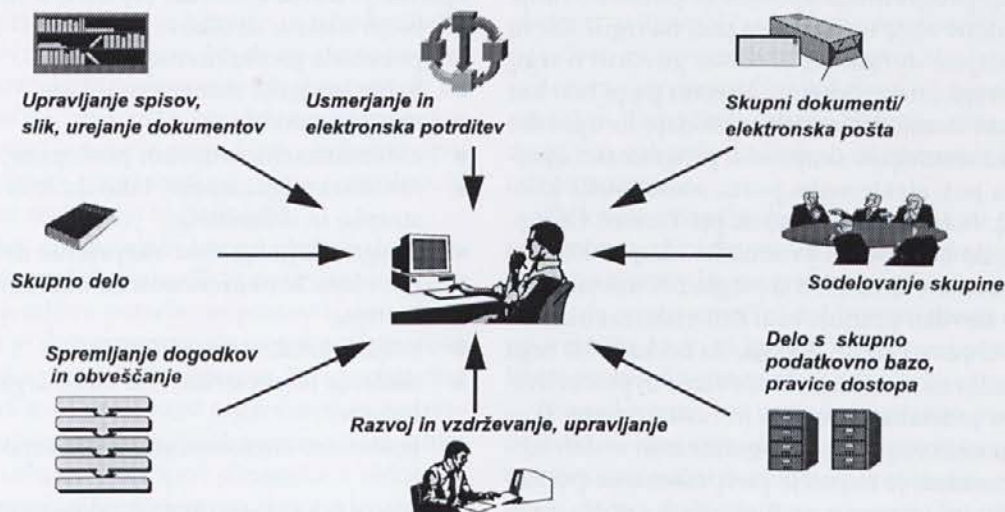
Za natančnejšo pozicioniranje produktov na trgu srečujemo različne klasifikacijske sheme skupinske programske opreme. Produkte lahko razvrstimo po skupinah v kategorije:

- elektronska pošta, koledar, rokovnik (scheduling),
- upravljanje spisov (dokumentov) v krajevnih računalniških omrežjih, z delovno potjo, upravljanjem dokumentov in slik ter skupinsko urejanje dokumentov,
- sistemi za pomoč pri sprejemanju odločitev z audio in video konferenco,
- pripomočki in razvojna orodja, grafični vmesniki (GUI), orodja za vzdrževanje in razvoj,
- produkti za delo s skupno podatkovno zbirko, skupnimi objekti.

Mrežno okolje in skupinska programska oprema

Skupinska programska oprema potrebuje za svoje delovanje mrežno infrastrukturo, ki vsebuje osebne računalnike, operacijske sisteme, ožičenje, upravljalne pripomočke in telefonske linije v primeru WAN.

Je torej del mrežnega aplikacijskega okolja. Vendar ni vsaka mrežna aplikacija in tudi ne dostop do skupne podatkovne zbirke prek omrežja nujno že skupinska programska oprema. Interaktivno dostopna ali proizvedovalna podatkovna zbirka pa je že lahko del skupinskih aplikacij.



Slika 2. Skupinska programska oprema



Vir: David Coleman

Slika 3. Okolje skupinske programske opreme

Skupinska programska oprema temelji na tehnologijah odjemalec-strežnik, multimedia, upravljanje slik in spisov (dokumentov), omrežne aplikacije, porazdeljeno, mobilno, oddaljeno (remote) računalništvo. Organizacijski razvoj gre v smeri manj nivojskih organizacij (flatter organisations), decentralizacije in 'outsourcing'-a. To se odraža tudi v tehnologiji, ki sledi tem smernicam: rast omrežij in manj razvoja na podedovanih (legacy) sistemih. Še vedno funkcionalne in neamortizirane centralne računalnike in lokalna omrežja skušajo vodje oplemenititi z novo kvaliteto.

TRG SKUPINSKE PROGRAMSKE OPREME

Je skupinska programska oprema le poimenovanje nekaterih produktov ali je to stvarnost tudi na trgu? Ker ni enolične, skupne definicije, je težko govoriti o trgu skupinske programske opreme. Vseeno pa je bilo kar nekaj raziskav narejenih na temo prodaje in uporabe tehnologij, ki sestavljajo skupinsko programsko opremo (delovna pot, elektronska pošta, elektronska konferenca, itd.). Bob Flangan, analitik pri Yankee Group, predvideva, da bo prodaja produktov skupinske programske opreme v letu 1998 dosegla 5.5 milijard US dolarjev. To številko potrjuje tudi Britansko raziskovalno podjetje Ovum, Ltd. in dodaja, da bo kar 60% tega zneska odpadlo na stroške izobraževanja in postavitve.

Zanimiva je analiza uporabe in rasti uporabe skupinske programske opreme po posameznih vrstah tehnologije. Trenutno je največje povpraševanje po elektronski pošti in delovni poti. Elektronska pošta, ki še danes pomeni 19% trga skupinske programske opreme,

naj bi v letu 1998 pomenila le še 10% trga, integrirana pa bo v druge tehnologije kot so delovna pot, koledar, sistemi za podporo odločanju.

Pomembno gonilo rasti trga skupinske programske opreme postaja reinženiring organizacij. Skupinska programska oprema nudi namreč mnoga orodja potrebna za izdelavo strategij reinženiranja in tudi pri izvedbi reinženiring programov so orodja skupinske programske opreme na prvem mestu.

ZAKAJ SE UPORABNIKI ODLOČAJO ZA SKUPINSKO PROGRAMSKO OPREMO

Osnovne motive uporabe skupinske programske opreme je David Coleman [4] strnil v naslednje točke:

- boljši nadzor stroškov,
- povečana produktivnost,
- boljše nudenje storitev strankam,
- manj sestankovanja,
- avtomatizacija rutinskih postopkov,
- razširitev organizacije, tako da le-ta vključuje tudi stranke in dobavitelje,
- integracija geografsko razpršenih delovnih skupin,
- povečana konkurenčnost zaradi hitrejši prisotnosti na trgu,
- boljša globalna koordinacija,
- nudenje novih storitev, ki ločijo organizacijo od sorodnih,
- postavitve strokovnosti na ustrezno mesto.

Seveda je tehnologija tista, ki daje rešitve poslovnim zahtevam. In ta je danes tu:

- mrežna infrastruktura je taka, da omogoča uporabo skupinske programske opreme,
- ugodnejše razmerje med ceno in performansami tako strojne kot programske opreme je eden od faktorjev široke uporabe,
- svetovna recesija, "downsizing" zahteva večjo produktivnost zaposlenih,
- tudi znani proizvajalci Digital Equipment, Microsoft, IBM promovirajo produkte skupinske programske opreme,
- vse večja konkurenca zahteva spremembe organizacij, manj nivojev, večjo fleksibilnost,
- večja kompleksnost v proizvodih in poslovnih postopkih vodi v sodelovanje neformalnih skupin, ki potrebujejo skupinsko programsko opremo za uspešno delo,
- in tudi članki na to temo povečujejo zavest in ravednost o skupinski programski opremi.

Vendar pri vsaki lepi, napredni sili so tudi pomisleki in nasprotni usmeritve. Večjo rast uporabe skupinske programske opreme omejuje premajhno znanje, nejasnost na trgu, stroški investicije, pomanjkanje standardov. In spisek verjetno ni zaključen.

Pri vsaki postavitvi skupinske programske opreme ima po besedah anketirancev [4] največjo vlogo najvišje vodstvo in dobro definiran poslovni problem, ki naj se modernizira z uporabo skupinske programske opreme.

KORISTI SKUPINSKE PROGRAMSKE OPREME

Skupinska programska oprema ni le nova faza v razvoju informacijske tehnologije. Je predvsem uporaba tehnoloških sistemov za podporo "naravne" organiziranosti v delovnih sredinah. Namen skupinske programske opreme ni vzpostavljanje nove strukture, temveč povečuje učinkovitosti in uspešnosti obstoječih skupin.

Ponuja kar lep niz možnih koristi. Najmanj kar lahko od nje pričakujemo je hitrejša in učinkovitejša izvajanje poslovnih postopkov. Na drugi strani pa obeta popolno prestrukturiranje poslovnih procesov, ki jih izvajajo dinamične, funkcijsko prepletene delovne skupine. Kaj je "bolje" je seveda prepuščeno definiciji uporabnikov, skupinska programska oprema pa je tista, ki mora omogočiti realizacijo te definicije.

Širok spekter potencialnih koristi otežuje definiranje realističnih pričakovanj. Pri postavitvi je težko definirati oprijemljive potrebe in postaviti dosegljive cilje. In seveda je skoraj nemogoče meriti performanse rešitve skupinske programske opreme. Pri vzpostavitvi pravega odnosa med potrebami organizacije in ustrežno rešitvijo skupinske programske opreme nam lahko pomaga razumevanje principov dinamike v delovnih skupinah: komunikacije, sodelovanja in vsklajevanja.

Eden glavnih pogojev je, da je rešitev enostavno

prilagodljiva stalno se spreminjajočim poslovnim zahtevam. In kot v vsaki organizaciji obstajajo delovne skupine, tako obstajajo tudi delovni postopki, ki jih le-te uporabljajo pri izvajanju delovnih nalog. Delovne skupine (že brez skupinske programske opreme) med seboj komunicirajo, sodelujejo in vsklajujejo naloge na osnovi formalnih in tudi neformalnih postopkov. Posebno ti neformalni mehanizmi so tista področja, kjer največ pridobimo s tehnologijo skupinske programske opreme. Pri postavitvi rešitve moramo sodelovati z uporabniki, da lahko dobro razumemo vse te procese in ugotovimo, kaj lahko z uporabo nove tehnologije izboljšamo.

Pri proučevanju procesov, ki jih želimo modernizirati, moramo razumeti osnovne principe dinamike delovnih skupin za vsak posloven proces. Produkt, ki pokriva le enega teh mehanizmov (npr. komuniciranje, kot ga nudi elektronska pošta), še ni popolno okolje skupinske programske opreme. Zahteve po komunikaciji, sodelovanju, vsklajevanju (koordinaciji) so seveda v medsebojni odvisnosti.

Komuniciranje pomeni posredovanje podatkov in informacij. Komunikaciji zadostimo v prvi vrsti z možnostjo pošiljanja sporočil v delovni sredini oz. v podjetju. Za to ni dovolj le investicija v različno računalniško opremo in telefonijo. Tehnične nepovezljivosti, razlike pri postavitvi in celo upravljanje teh sistemov so pogosto vzrok nepovezanosti le-teh. Pri razmišljanju o skupinski programski opremi moramo ugotoviti, kateri komunikacijski kanali so kritični za osnovno funkcioniranje, kateri dajejo pomembne možnosti in kateri so le opcijski. Pri tem se moramo zavedati, da mora biti možnost pošiljanja sporočil zanesljiva, dostopna, enostavna za končnega uporabnika in predvideti moramo vse oblike komuniciranja sodelavcev v skupini.

Kako preverimo te kriterije? Za zanesljivo lahko imamo komunikacijo, ki deluje določen procent časa glede na zahtevani čas (reactive approach). Lahko pa se odločimo za preventivni pristop in skušamo z rednim vzdrževanjem preprečiti kakršnokoli napako v komunikacijah. Ta način je seveda bistveno zahtevnejši (tudi dražji) in zahteva podrobno načrtovanje in višji nivo infrastrukture.

Delovne skupine in organizacije so povezane z različnimi načini komuniciranja. Pri prehodu na tehnologijo skupinske programske opreme moramo zagotoviti, da so vsi ti načini del integriranega sistema. Vsi uporabniki, ki so vključeni v poslovni proces, morajo imeti dostop do sistema.

V poslovnem svetu se informacije prenašajo kot podatki, dokumenti, glas. Komunikacijska infrastruktura mora omogočiti dostop do pravočasnih, točnih informacij, če naj služi svojemu namenu. Pri postavitvi sistema moramo zato poznati vire podatkov, njihovo

pomembnost in ugotoviti, če zanje že imamo računalniško podporo. In predvsem tista za katere le-te še ni, je verjetno prva na listi kandidatov za nov tehnološki pristop. Seveda ne smemo pozabiti, da morajo biti obstoječe rešitve vključene v novo tehnološko rešitev.

Skupinska programska oprema lahko bistveno nadgradi zanesljiv sistem sporočanja, ki je osnovni komunikacijski nivo. Programski dodatki lahko avtomatsko začnejo določene aktivnosti brez ali le z malo človekove pomoči. Kot primera omenimo usmerjanje le ustrezne pošte do uporabnika in avtomatizacijo usmerjanja dokumentov skozi poslovni proces.

Sodelovanje je nadgradnja komunikacijskega nivoja in mora biti prilagojeno poslovnemu procesu ter uporabnikom delovne skupine. Organizaciji omogoča, da se opre na svoje (in tuje) intelektualne vire. Glavna prednost računalniško podprtega sodelovanja je njena "brezčasovnost" in "brezmejnost". Delavci v različnih krajih, v različnih časovnih pasovih med seboj delujejo kot člani skupine.

Računalniško podprto sodelovanje je prineslo še eno prednost. Pri običajnih sodelovanjih, sestankih ipd. le redko lahko znanje iz teh procesov uporabimo kasneje. Razni "prebliski" in izkušnje so po končanem projektu običajno izgubljene in drugi posamezniki bodo ponovno "izumljali kolo". S produkti skupinske programske opreme lahko te izkušnje shranimo, uredimo, delimo rezultate z drugimi v organizaciji in so na razpolago kasneje. Učinkoviti pripomočki za iskanje in posredovanje znanja pomenijo bistveno spremembo v upravljanju znanja.

Vsklajevanje (koordinacija) sinhronizira napore delovne skupine na najučinkovitejši način. Omogoča delitev uporabe virov, naporov in odgovornosti. V računalniško nepodprtih delovnih sredinah na osnovi predpostavk, da lahko predvidimo okoliščine, ki vplivajo na napredek dela, planiramo vsklajevanje, izjeme upoštevamo le potem, ko je do njih prišlo. Edini način spremljanja napredka so poročanja v časovnih razmikih.

V okolju skupinske programske opreme lahko večino merjenj napredka avtomatiziramo ter omogočimo podrobnejše in nenehno nadziranje. Lahko se bolje posvetimo izjemam, saj jih prej odkrijemo.

Produkti, ki podpirajo vsklajevanje, so različni. To so produkti za razporejanje časa (scheduling), produkti za vodenje projektov (project management) in produkti delovnih poti (workflow). Predvsem produkti delovnih poti so že kar poznani. Funkcije so namreč dobro definirane, koristi lahko opišemo in določimo razumne cilje uporabnikov. Običajno delovne poti ločimo v tri vrste: produkcijsko, administrativno in ad hoc - trenutno delovno pot. Primer prvega je zahteva za zavarovanje, primer drugega je poročilo o stroških in tretjega podpora vsklajevanja dela delovne skupine na projektu.

Delovne sredine se običajno organizirajo glede na potrebe dela in niso togo podrejene formalni organiziranosti. Tudi programska orodja, ki naj bodo uspešna, morajo omogočiti svobodo prilagajanja spremembam delovnih zahtev in poslovnih pogojev. Na drugi strani pa morajo slediti ciljem celotne organizacije.

Razmislek o komunikaciji, sodelovanju in vsklajevanju v organizaciji

- Vsako okolje delovne skupine zahteva komunikacijo, sodelovanje in vsklajevanje.
- Komunikacije in sporočanje morajo biti popolne in zanesljive, saj so osnova za povečanje učinkovitosti in uspešnosti skupine.
- Delovne skupine morajo imeti možnost širitve obstoječe komunikacijske infrastrukture, kot to narekujejo potrebe. Z ustrezno namestitvijo komunikacijskih kanalov postavijo najprimernejši način sodelovanja.
- Tehnične rešitve, ki jih definirajo sodelavci delovne skupine, morajo podpirati potrebe po sodelovanju. Shraniti morajo intelektualne rezultate dela delovne skupine za kasnejšo uporabo.
- Delovne skupine morajo imeti možnost nadzora mehanizmov vsklajevanja. Podobnosti v postopkih, dodelitev nalog, delitev odgovornosti morajo biti dostopne skupnemu nadzoru.
- Rešitve skupinske programske opreme moramo nenehno spremljati ob spreminjajočih se poslovnih pogojih.

PORAZDELJENI OBJEKTI KOT DEL INFRASTRUKTURE SKUPINSKE PROGRAMSKE OPREME

Pogosto so aplikacije skupinske programske opreme primerne le za delo posameznih skupin in se ne posvečajo problemom firme. Pred kratkim je Aberdeen v svoji raziskavi identificiral novo kategorijo skupinske programske opreme, ki temelji na porazdeljenih objektih, imenuje jo *skupinska programska oprema podjetja* (Enterprise Groupware). Ta tehnologija že dolgo ni le stvar raziskovalnih laboratorijev, je stvarnost na trgu. To je integracijsko ogrodje, ki vključuje različne odjemalce, strežnike in aplikacije.

Kaj skupinska programska oprema podjetja nudi in česa ne, je prikazano na sliki 4, povzeti po Aberdeen Group.

Skupinska programska oprema podjetja ima naslednje lastnosti:

- rešitve združujemo z vlečenjem miške (drag/drop),
- rešitev lahko postane osnovna komponenta, ki jo lahko uporabimo v drugih komponentah,
- delovno okolje omogoča izdelavo, testiranje in distribuiranje novih komponent,

- sprememba v komponenti se takoj odraža v vsaki rešitvi, kjer je uporabljena,
- podedovane (legacy) aplikacije in podatki lahko postanejo komponente,
- uporabniki in skupine uporabnikov so objekti sistema s specifičnimi lastnostmi,
- podatki, aplikacije in uporabniki so neodvisni od kraja,
- vse elemente lahko poiščemo in dostopamo do njih in to kot uporabniki ali kot aplikacija.

Večina današnjih programskih rešitev skupinske programske opreme nima navedenih lastnosti, ker niso uvedene v objektno orientiranem okolju. So bolj standardna razvojna okolja z različnimi razvojnimi pripomočki, vsak od le-teh za drugo nalogo. Skupinska programska oprema podjetja pa je tako fleksibilna, da lahko vključi že uvedeni proces prve generacije skupinske programske opreme kot komponento sistema skupinske programske opreme podjetja.

Informacijski sistemi in skupinska programska oprema

V vsakem podjetju obstaja bolj ali manj uspešen informacijski sistem, običajno vsaj delno podprt z računalniškimi aplikacijami. Tega uporabljajo različne skupine uporabnikov pri svojem vsakodnevnem operativnem delu in verjetno tudi upravljavci teh sistemov. Te rešitve želimo uporabiti tudi v novem okolju in skupinska programska oprema *podjetja* nam to omogoča. Je odprta v zelo širokem pomenu te besede. Temelji na distribuiranih objektih in odprtih podatkovnih zbirkah. Je integracijsko okolje različnih odjemalcev, strežnikov in aplikacij. Z vključevanjem (encapsulating) aplikacij omogoči uporabnikom preprost dostop do vseh skupnih podatkov, tudi tistih v podedovanih podatkovnih (legacy) sistemih, na strežnikih ali odjemalcih. Ker so že v samem ogrodju podatki o vseh uporabnikih, je dostop do podatkov zelo strogo nadzorovan, tudi v aplikacijah, ki sicer niso 'varne'. Ko se uporabnik prijavi, lahko nadzorovano dostopa do vseh podatkovnih objektov in ker so le-ti centralno hranjeni, dostopa lahko iz kateregakoli delovnega mesta.

Ne le, da ima vsak pooblaščen uporabnik zelo preprost dostop do vseh podatkov s poljubnega delovnega mesta, v ta dostop je vključena tudi delovna pot, ki omogoča pravočasno in nadzorovano izpeljavo vseh aktivnosti nekega informacijskega sistema.

Aberdeen Group v svoji raziskavi pravi, da se kot tak produkt danes lahko pohvali Digitalov LinkWorks. Uporabniki, ki že uporabljajo ta produkt, so potrdili, da jim je uspešno pomagal pri vpeljavi pomembnih aplikacij v rekordnem času. Ob tem so imeli polno podporo tako uporabnikov kot tudi oddelčnih vodij in upravljalcev.

KORISTI ZA POSAMEZNE SKUPINE UPORABNIKOV

Ohlapna definicija skupinske programske opreme, da je to vsaka programska oprema, ki nam omogoča izvajanje posamezne naloge ali sodelovanje v skupini z uporabo aplikacij, omrežja in komunikacij, ne upošteva specifičnih nalog, ki jih oddelki želijo realizirati s pomočjo skupinske programske opreme. Ker ima vsak oddelk drugačne cilje, so zato tudi njihove zahteve različne. In zato tudi taka raznolikost ponudbe produktov. Na primer: upravljalec informacijskih sistemov želi skupinsko programsko opremo uporabiti za upravljanje, nadzor in standardizacijo dostopa do podatkov, izvajati reinženiring in mogoče celo avtomatizirati podporo aplikacijam podjetja tipa odjemalec-strežnik. Želi imeti okolje za hiter razvoj, uporabo in spreminjanje aplikacij za različne odjemalce in strežnike in pri tem nadzirati dostop uporabnikov do skupnih podatkov in aplikacij. Če je le mogoče, želi uporabiti obstoječo podatkovno zbirko in infrastrukturo. Seveda so pomembni tudi preprosta zaščita (backup in recovery) in drugi pripomočki, ki omogočajo nemoteno delo.

Na drugi strani so zahteve uporabnikov in oddelčnih vodij. Ti želijo preprost dostop do informacij, podporo za skupinska dela, skupne dokumente, projekte in pri tem uporabiti že poznana orodja. Glavna kvaliteta skupinske programske opreme je preprostost sodelovanja skupine in avtomatizacija opravil. Pomembna je

Kaj nudi

Vključevanje podatkov, aplikacij in uporabniških profilov.
 Grafični pripomoček za izdelavo aplikacij z uporabo objektov.
 Mehanizem za združevanje komponent programske opreme v pakete in preprosto distribucijo le-teh.
 Osnovno funkcionalnost za vključene (encapsulated) aplikacije, npr. varnost, osnovna sporočanja, skupni objekti, centralno shranjevanje in nadzor verzij.

Česa ne nudi

Konverzije aplikativnih podatkov.
 Standardnih podatkovnih zbirk in zelo specialnih aplikacij.
 Standardnih funkcij tiskanja.
 Generiranje objektno orientirane aplikacijske kode za razvijalce.
 Izdelave funkcionalnosti, ki je zunaj okolja skupinske programske opreme.

Slika 4. Preglednica možnosti objektno orientirane skupinske programske opreme, vir AberdeenGroup, september 1995

tudi možnost iskanja po skupni shrambi objektov, identifikacija lastnikov in uporabnikov teh objektov, podatki o tem kdo je spreminjal objekt in kakšne so te spremembe.

Seveda to pomeni, da sta računalniški center in uporabnik pogosto na nasprotnih bregovih. Oddelki želijo imeti svoje lokalne podatkovne zbirke, obrazce, skupni informacijski center pa se bori za konsolidacijo podatkovnih zbirk in en sam pogled na obstoječe podatke. Posebej še s pohodom Interneta, postaja tudi problem varnosti in zaščite podatkov tako dostopnih podatkov vse pomembnejši.

Skupinska programska oprema podjetja temelji na distribuiranih objektih. Izvaja se tako na strani odjemalca kot na strežnikih, upravlja in integrira vse aplikacije, podatke in uporabnike. Okolje je konsistentno za večino odjemalcev in strežnikov, najdemo lahko katerikoli podatek v tem okolju, ne glede na aplikacijo, ki ga je ' naredila', platformo na kateri se nahaja ali kraj. Uporabniki lahko uporabljajo skupne podatke oz. objekte s sodelavci v projektni skupini ali podjetju in lahko avtomatizirajo izdelavo, spremembe in distribucijo podatkov. V istem okolju uporabljajo poznane obstoječe aplikacije na odjemalcih, strežnikih. Z uporabo obstoječih aplikacij so stroški administriranja in usposabljanja uporabnikov seveda mnogo nižji.

Ta programska oprema mora imeti vse tiste funkci-

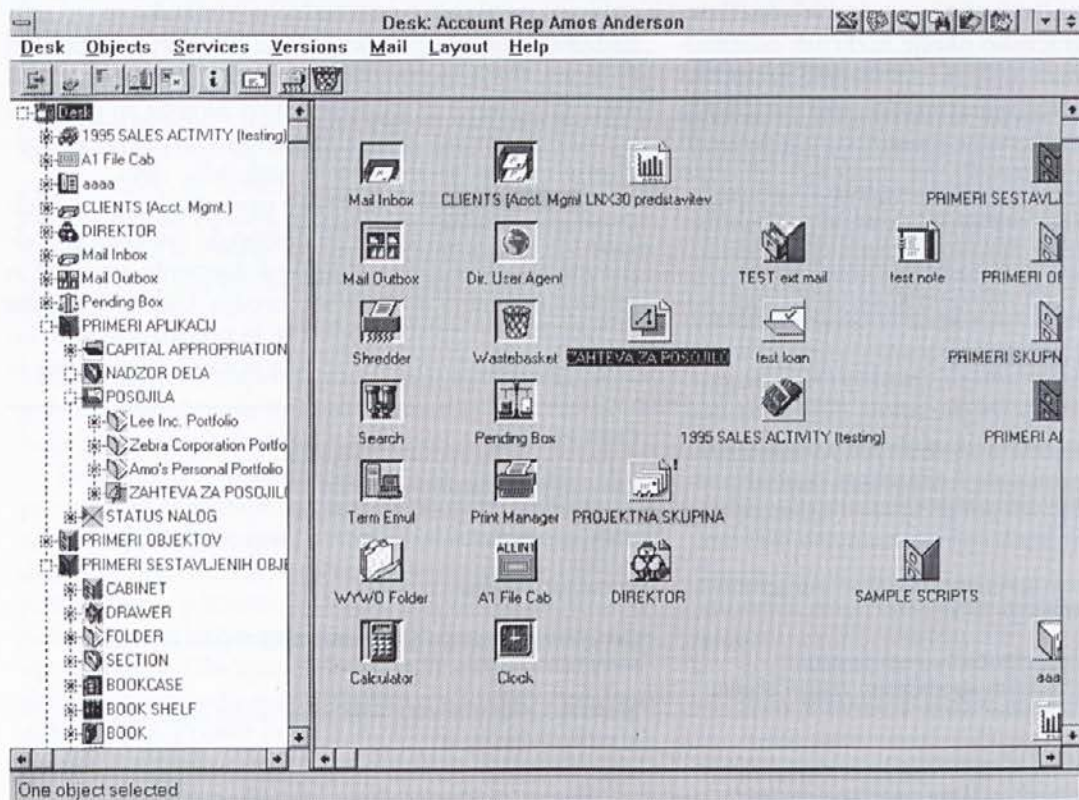
je, ki jih običajno pričakujemo od skupinske programske opreme: skupni dokumenti in objekti, elektronska pošta, upravljanje dokumentov, delovni tok; poleg tega še vključevanje obstoječih aplikacij v okolje skupinske programske opreme.

Velika dodatna prednost je, da se ob naslednji priključitvi na sistem, s katerega koli delovnega mesta, ki je lahko kjerkoli na svetu, vrnemo v okolje kakršno smo pustili na 'mizi', ko smo se odjavili od sistema, ob predpostavki, da nam administrativna pravila to dovoljujejo.

Ker ima veliko uporabnikov namizne računalnike, je pomembna tesna integracija z okni in OLE avtomatizacija za klic akcij na namiznem računalniku, ki to podpirajo.

In kakšne so koristi za informacijske centre? Razvijalci gradijo aplikacije v objektno orientiranem okolju s 'komponentami'. Zelo zahtevne rešitve lahko izdelajo na osnovi že obstoječih skupnih razredov in komponent. Novo funkcijo izdelajo s preprosto spremembo konfiguracije ali spremembo obstoječe komponente, lahko pa dodajo tudi nove komponente ali razrede. Vsako komponento, ki jo dodajo, lahko uporabijo v novih rešitvah. Sprememba komponente pomeni tudi avtomatično spremembo vseh rešitev, v katere je ta komponenta vključena.

Ponavljajoče se naloge uporabnika lahko avtomatizirajo z uporabo 'script' jezika tako, da v njem napišejo



Slika 5.

Uporabnik ima na mizi dostopne aplikacije, dokumente, shranjene na način, ki mu najbolj ustreza

majhno aplikacijo. Ta je lahko objekt, ki se lahko uporabi na katerem koli odjemalcu ali strežniku.

Zaradi kompleksnosti objektno orientiranih orodij razvijalci pogosto ne dosegaajo vseh prednosti, ki jim jih nudi objektno okolje. Ključ do uspešne izdelave objektnih aplikacij je zelo strukturirano okolje, ki omogoča tvoriti dovolj splošne objektno razrede za preprosto večkratno uporabo. Razvijalci morajo imeti možnost uporabiti prednosti objektnega okolja, ne da bi se jim bilo treba spustiti globoko na tehnični nivo.

Ponudnikov objektno orientiranosti razvoja je mogo, vendar je le malo 'pravih'. Osnovni principi objektnega razvoja ne pomenijo že tudi fleksibilnosti in preproste vzdrževanja. Kot dejansko v celoti objektno okolje omenja Aberdeen Digitalov LinkWorks. Ta res lahko vključuje obstoječe aplikacije, tvori nove neodvisne programske module iz starih in novih komponent, funkcionalnost pa deli še z aplikacijami znotraj okolja LinkWorks.

Objektno orientirani sistem LinkWorks se izvaja na strežniku in se uporablja za izdelavo, porazdelitev in upravljanje vseh objektov in komponent (aplikacije, podatki, atributi, metode). Vsak objekt ima globalno - svetovno enolično oznako. Za vsako vrednost (instance) objekta imamo podatke o tem, kdo je avtor spremembe, kdaj in kakšna je ta sprememba.

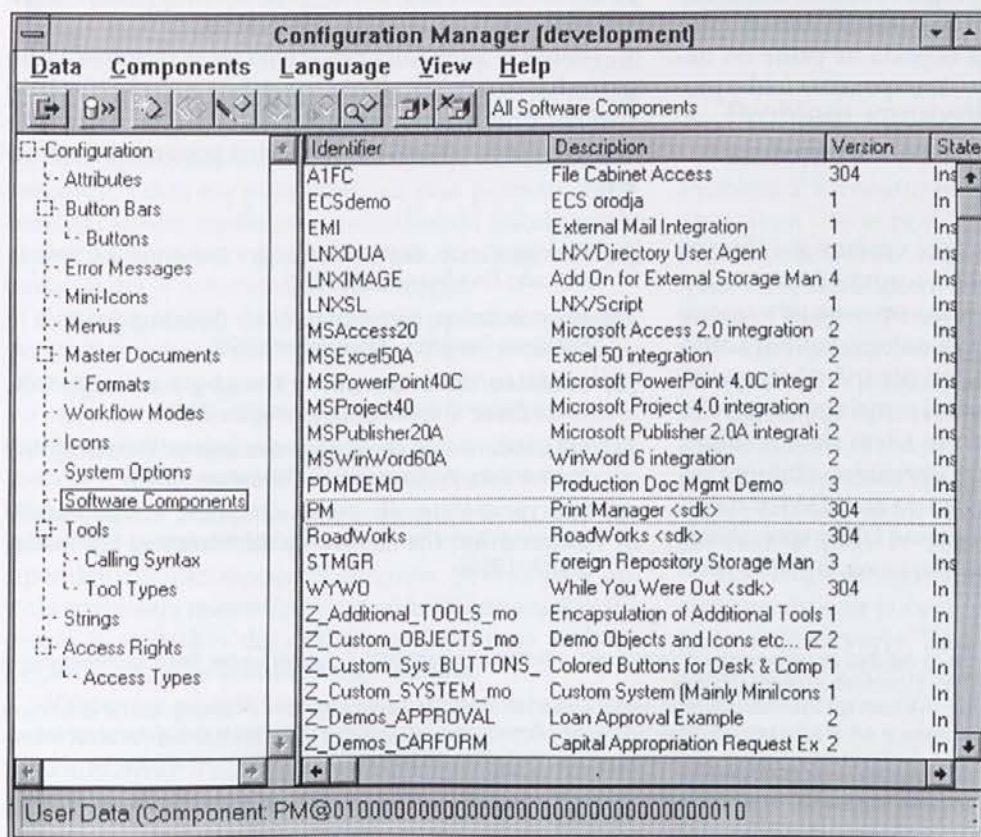
Prednost objektno orientiranosti za razvijalce je hitra

izdelava in uporabnost aplikacij, predvsem pa takojšna uporabnost sprememb, preprostost izvedbe manjših sprememb za različne oddelke in skupine uporabnikov.

Aplikacije in podatke integrirajo v sistem le enkrat in vsem v podjetju so na voljo. Vključevanje obstoječih sistemov seveda zahteva poznavanje tako objektov in možnosti APO (Application Plus Objects) kot tudi obstoječih aplikacij. Vendar se napor obrestuje. Tako lahko obstoječe operativne sisteme vključimo v rešitev skupinske programske opreme in s tem upravičimo investicijo. Na drugi strani pa tudi 'stari' programerji postanejo tako strokovnjaki v objektnem okolju.

Ko rešimo vse te probleme dobimo

- popolno varnost v skupinskem okolju,
- integracijo obstoječih uporabnikov pošte v robusten sistem sporočanja,
- objektni repozitorij vseh podatkov in virov s preprostim varovanjem (backup, recovery),
- možnost kompleksnih poizvedovanj čez vse poslovne objekte v mreži,
- večjo uporabnost obstoječih delovnih postaj in znanja o aplikacijah,
- vzdrževanje objektov ne glede na podatkovne zbirke in aplikacije,
- možnost centralnega upravljanja večine performansnih lastnosti okolja,



Slika 6.

Programske
komponente
v LinkWorks sistemu

- vključitev OLE avtomatizacije v distribuirano okolje.

Razvijalci imajo razvojno okolje, ki je

- objektivno izrazito strukturirano in spodbuja dobro izdelavo kode in učinkovit razvoj,
- ogrodje, ki se ga preprosto dopolnjuje z različnimi prilagodljivimi metodami,
- dobro definiran programski vmesnik APO (Application Plus Objects) za učinkovito integracijo obstoječih aplikacij,
- modularne programske komponente, ki jih lahko tvorijo in združujejo v nove aplikacije in tako skrijejo kompleksnost programske opreme,
- programske komponente so samostojne in prenosljive za instalacijo na drugih sistemih.

Prednosti strežno-orientirane arhitekture

V primeru tri nivojske arhitekture odjemalec-strežnik lahko strežnik uglašujemo (tunning) za različne platforme. Okolje sistema vodimo s centralnega mesta, vendar je padec katerega koli strežnika le lokalnega pomena in ne vpliva na delo ostalih uporabnikov.

Za zanesljivo varnost je pomembno, da moramo vsakemu objektu že ob tvorbi dodeliti dostopne pravice: kaj lahko kdo z njim počne. Administrator sistema pa določi vrste dostopnih pravic za posamezno skupino uporabnikov iz obstoječih možnosti oz. doda nove pravice. Aplikacije in vzorce lahko združujemo v komponente, ki jih lahko na preprost način instaliramo oz. deinstaliramo na drugih strežnikih. Zaradi enoličnega označevanja vsakega objekta ne pride do neželjenega podvajanja objektov. Komponente lahko prenesemo prek medija ali pošte.

ZAKLJUČEK

Postavitev skupinske programske opreme je zahtevna naloga. Vodstvo podjetja zahteva upravičenost investicije in če skupinska programska oprema ni v uporabi oziroma se slabo uporablja za naloge, katerih koristnost težko ocenimo, ne moremo (finančno) upravičiti investicije. Problem je lahko tudi samo slabo upravljanje nove programske opreme in s tem slabi rezultati. Največje koristi te programske opreme so običajno neprijemljive in jih težko merimo na standarden način. Kot primer povejmo, da s to opremo lahko spremljamo trajanje posameznih aktivnosti in se na ta način zave-

damo tudi pomena stroškov teh aktivnosti in to tako izvajalci aktivnosti kot tudi uporabniki - odjemalci teh aktivnosti.

Za uspeh pri postavitvi ustreznih rešitev ni dovolj le tehnologija. Največji problem, ki ga srečujejo uvajalci novih rešitev, je to, da podjetja ne želijo spremeniti načina svojega dela.

Storitvene, svetovalne hiše in tudi dobavitelji skupinske programske opreme nudijo raznoliko paleto storitev. Ta vključuje planiranje, svetovanje, postavitve, izdelavo specifičnih aplikacij za stranko, svetovanje o organizacijski strukturi, spremembe upravljanja in zasnove poslovnih procesov, organizacijo sestankov. Dobavitelji nudijo predvsem pomoč pri postavitvi in izdelavi specifičnih aplikacij, največkrat zaželena pomoč pa je pomoč pri planiranju. Pri prilagoditvah, organizacijskih in upravljaljskih spremembah so svetovalne hiše pogosto prava pot do uspeha. Najpomembnejša pa je seveda, tako kot pri vsaki novosti v podjetju, popolna podpora v najvišjem vodstvu.

Koristi od skupinske programske opreme podjetja imajo računalniški centri in uporabniki. Računalniški centri imajo koristi, ker so izdelava, uporaba in upravljanje aplikacij v objektu zelo hitre in preproste. Uporabniki pa lahko razvijajo rešitve z uporabo orodij namiznega računalnika, ki jih že dobro poznajo. Z vključevanjem aplikacij in podatkov v *skupinsko programsko opremo podjetja* dobi uporabnik dostop do vseh podatkov firme, tudi tistih v operativnih podatkovnih sistemih, na strežnikih ali odjemalcih.

Morda se vam zdijo zagotovila, da je skupinska programska oprema podjetja uporabna na tako širokem spektru problemov, nerealna. Pozabite na dvome, ta tehnologija je realnost in je vredna pozornega razmisleka.

- [1] AberdeenGroup, *Digital LinkWorks - Delivering Solutions to MIS and End-Users*, July 1995
- [2] AberdeenGroup, *Digital LinkWorks - Delivering Solutions to Solutions Providers*, September 1995
- [3] COLEMAN, David: *Groupware: Changing Business for '90s*, White Paper Collaborative Strategies, 1994
- [4] COLEMAN, David: *Groupware: Groupware: Technology and Applications, An Overview of Groupware*, 1995
- [5] The Yankee Group: *Communication, Collaboration, Coordination: The "Three Cs" of Workgroup Computing*, March 1995

Nevenka Gorenšček je diplomirala na fakulteti za naravoslovje in tehnologijo - Odsek za matematiko v Ljubljani - smer Tehnična matematika in magistrirala iz operacijskih raziskav na Ekonomski fakulteti v Ljubljani.

V informatiki in računalništvu je najprej delala kot sistemski inženir in nato kot sistemski analitik ter vodja projektov. Pri svojem delu je uporabljala sodobne metode strukturiranega pristopa, kar je s pridom uporabila tudi v kasnejšem procesu izobraževanja. Delala je kot inštruktor ter kasneje kot vodja področja izobraževanja s področja informacijske tehnologije ter za svoje delo napisala nekaj priročnikov ter programov tečajev s področja informatike.

V EuroComputer Systems, kjer je sedaj zaposlena, se je ukvarjala z Digitalnimi produkti informacijske tehnologije (baza, orodja, CASE), v zadnjem času pa dela predvsem na sodobnem objektu orodja skupinske programske opreme LinkWorks.

ORGANIZACIJSKI VIDIKI VAROVANJA INFORMACIJSKIH SISTEMOV (I)

Tomaž Poštuvan

Povzetek

Prispevek o organizacijskih vidikih varovanja informacijskih sistemov obravnava področje, ki je tudi pri nas vedno bolj pomembno. Vsaka organizacija ima premoženje, ki je lahko običajen denar ali pa so to naslovi strank oz. informacije o dogajanjih na trgu in borzi. Če so kakšni od teh podatkov nenadoma nedostopni ali ukradeni, lahko to povzroči neprijetnosti, izgubo denarja ali pa celo bankrot. Zato je podatke potrebno varovati.

Abstract

The paper discusses an area, which is of outmost importance to all the organizations. Every organization possesses assets that include money, customer adresses and marketing informations. If some or all of those assets are either unavailable for use or stolen, the organization might suffer inconvenience, loss of money or maybe even go out of business. Therefore, data must be secure.



UVOD

Vseprisotnost računalniških informacijskih sistemov v današnji družbi močno izpostavlja vprašanje njihove varnosti oziroma zaščite pred malomarnostjo ali nepooblaščenimi uporabniki. Omenjeni sistemi imajo veliko materialno in drugačno vrednost, zaradi česar obstaja tudi temu primerno tveganje izgub zaradi malomarnosti ali protizakonitih dejanj. Prav zaradi pomembnosti vprašanja je varovanje in zaščita podatkov danes posebna stroka, ki se na sistematičen način ukvarja s področjem vsestranske zaščite informacijskih sistemov. Takoj naj pojasnimo, da pod pojmom informacijski sistem mislimo na računalniški informacijski sistem oziroma informacijski sistem, katerega bistveni sestavni del je računalniška tehnologija.

Velja poudariti, da je varovanje podatkov pomembno tudi iz splošno civilizacijskih razlogov in ne le zaradi materialne vrednosti, ki jo imajo informacijski sistemi oziroma shranjeni podatki. Danes je zasebnost uveljavljena civilizacijska vrednota, ki je v večini držav na ustrezen način tudi ustavno in zakonsko varovana. Zaradi tega je primerno varovanje podatkov, ki se nanašajo na posameznike, obveznost vseh lastnikov in uporabnikov informacijskih sistemov. Seveda pa je obveza varovanja materialne vrednosti informacijskih sistemov in podatkov del običajnega delokroga vodstva organizacije, ki je lastnik takega sistema.

Varovanje podatkov oziroma njihova varnost ima več vidikov, od katerih tu omenjamo dva: tehnični in organizacijski. Tehnična stran obsega vse tehnične probleme zagotavljanja varnosti in načine njihovega reševanja, organizacijska pa zbirko metod, načel, na-

potkov in podobno, ki zagotavlja, da se varovanje podatkov kot ena od nalog podjetja ali organizacije izvaja zanesljivo in učinkovito. V tem sestavku se ukvarjamo predvsem z drugo, organizacijsko stranjo zaščite podatkov, čeprav bomo nekaj prostora namenili tudi tehničnim vprašanjem.

1. Problem varnosti informacijskih sistemov

Problem z varnostjo informacijskih sistemov je zelo enostaven - to je problem ljudi. Rešitev problema je prav tako zelo enostavna - ljudje naj kontrolirajo ostale ljudi v računalniškem okolju. Misel "Računalniki ne kradejo, ljudje pa" stoji na trdni osnovi. Eden od ljudi, ki se ukvarjajo z varnostjo v enem večjih računalniških podjetij v ZDA je rekel, da še nikoli v svojih tridesetletnih izkušnjah nadzora in preiskovanja računalniških goljufij ni naletel na primer, ko za prevaro niso bili odgovorni določeni ljudje [3, stran 3]. Program varnosti podatkov mora večino naporov usmerjati v nadzorovanje ljudi, če želi biti uspešen.

Eden od kriterijev pomembnosti varnosti informacijskega sistema je tudi narava posla, s katerim se organizacija ukvarja. Zadnje čase, ko se organizacije vključujejo v informacijske tokove, se zavedajo nujnosti zaščite informacij. V nekaterih organizacijah sicer še vedno mislijo, da je programska oprema strošek, ne pa premoženje organizacije, vendar so na srečo taka v manjšini. Zapisani ali ne, podatki (in z njimi programska oprema) so glavno premoženje organizacije.

Svet varnosti informacijskih sistemov je velik in še raste. Varnost organizaciji sicer ne bo prinesla denarja v klasičnem smislu, vendar bo povečala posredni dobiček zaradi zmanjšanja stroškov in preprečevanja odtekanja informacij.

1.1 Kako razmišljati o varnem sistemu ?

Odgovornost za varnost podatkov na vsak način nosi vodstvo organizacije in sicer za varovanje premoženja, izdelavo sistemov za notranjo kontrolo, ki varujejo premoženje ter za ugotavljanje, če premoženje sploh še obstaja. Bolj pomembni kot so podatki, večja je potreba po izpolnitvi te vrste odgovornosti.

Nekdo se bo vprašal : "Kaj vodstvo organizacije dela, da bi izpolnilo svojo odgovornost do varnosti?". Tudi vodstvo se mora vprašati, kaj dela za varnost sistema, preden začne kritizirati program varnosti kot neučinkovit.

Poglejmo nekaj vprašanj, ki se nanašajo na varnost in lahko vodstvo organizacije spravijo v zadrego :

- **Ali vodstvo aktivno sodeluje v programu varnosti?** Aktivna udeležba pomeni veliko vložene časa, določanje strategije, povpraševanje ostalih delavcev o njihovem vložku ... Samo podpora in odobravanje nista dovolj.
- **Ali je vodstvo spoznalo njegove glavne šibke točke?** Če se šibke točke ne odkrijejo, to avtomatično pomeni, da se strategija varnosti ne more primerno sestaviti.
- **Če so šibke točke znane, ali poznamo njihov relativni pomen?** Velikost sredstev, ki bodo namenjena določeni šibki točki, je odvisna od možnosti, da pride do incidenta ravno pri njej. Sistem je tako varen, kot je varen njegov najšibkejši člen.
- **Ali uporabljene rešitve delujejo?** Vodstvo ima za nalogo tudi ocenitev učinkovitosti programa, sicer se lahko denarna sredstva, ki so za varnost zelo velika, trošijo nenamensko.

Začetek kateregakoli programa za varnost je ocena trenutnega stanja. Dokler ne vemo, kje smo, ne bomo nikoli vedeli, kam moramo iti. Obstajata dve poti za ugotovitev, kako varen je informacijski sistem. Prvi pristop je natančna, v globino usmerjena ocena, ki opozori na konkretne grožnje in na primerne protiukrepe za zmanjšanje teh groženj. Ta pristop zahteva strokovno usposobljeno ekipo, ki natančno pozna informacijski sistem v organizaciji. Drugi pristop je bolj površen - pokazati na tista področja, kjer nevarnost obstaja in za varnost ni dovolj dobro poskrbljeno.

Prvi pristop je ekvivalenten natančnemu pregledu kardiovaskularnega sistema z EKG, stresnimi testi, merjenjem količine holesterola itd., medtem ko drugi pristop ustreza ogledu karakteristik, ki vodijo k problemu s srcem - diete, visokega pritiska, kajenja itd. Obe

metodi sta napovedovalca srčnih težav, nobena ni popolnoma zanesljiva, vendar je prva bolj natančna in je verjetnost pravilne napovedi večja.

Vodstvo organizacije nosi osnovno odgovornost za varnost sistema, da pa bo informacijski sistem tudi resnično varen, mora pomagati sleherni zaposleni. Moramo razlikovati med odgovornostjo vodstva za program varnosti in odgovornostjo delavcev za posamezne aktivnosti v okviru programa.

2. Kako velik je problem varnosti ?

Vsak program varnosti se mora začeti z ugotovitvijo trenutnega stanja v organizaciji. Le-to nam predstavlja neke vrste začetno točko za njegov izboljšanje.

2.1 Trenutno stanje v organizaciji

Trenutno stanje v organizaciji je natančen pregled programa varnosti informacijskega sistema v določenem trenutku. Namenjeno je odgovoru na dve vprašanji : (1) Kaj delamo za varnost podatkov ? (2) Kako učinkovit je naš program varnosti ?

Pregled stanja organizacije naj bi napravili neodvisni ljudje, saj bi s tem odpadli vsi predsodki, ki se lahko pojavljajo v organizaciji, če bi pregled delali notranji sodelavci.

Obstajata dve kategoriji informacij, ki jih je treba zbrati: prva je vezana na sam proces varnosti in vključuje strategije, metode, procedure in tehnike, ki jih potrebujemo za varovanje računalniških sredstev. Druga kategorija pa vključuje varnostne dokumente - informacije o poskusih vdora oz. samih vdorih v sistem. Če je mogoče, naj bi bile izgube tudi ocenjene.

Kakšna je potreba po pregledu stanja organizacije? Delavci, ki delajo v organizaciji daljše obdobje, s časom dobijo določene izkušnje in jih je zelo težko prepričati, da so možna tudi druga dejstva, ki so bolj pravilna. Neki strokovnjak za varnost je dejal, da v skoraj vseh organizacijah velja ti. **dvajsetletno pravilo**, ki pravi, da če se neki neprijeten dogodek ni zgodil v zadnjih dvajsetih letih, potem je verjetnost, da se tudi v bodoče ne bo zgodil, enaka ena [3, stran 22]. To pa seveda ni nujno. Edina rešitev, ki vodstvo prepriča v nasprotno, je prikaz dejstev. V svetu varnosti je tak pristop nujen.

2.2 Ključni koraki zbiranja informacij o stanju organizacije

Zbiranje informacij o stanju podatkov ni nujno časovno potratna stvar. Cilj je zbrati tisto, kar lahko zberemo razmeroma enostavno. Trije ključni vidiki so :

- Kaj zbirati
- Od koga bomo dobili informacijo
- Točnost zbranih informacij

Zbiranje informacij o stanju v organizaciji mora biti

končano v enem tednu (razen v velikih organizacijah, kjer se lahko zbiranje malce zavleče). Postopek je sestavljen iz naslednjih šestih točk:

1. Določitev skupine za zbiranje
2. Določitev zahtev in ciljev zbiranja
3. Načrt zbiranja informacij
4. Učenje članov skupine
5. Zbiranje informacij
6. Analiza in poročilo o stanju v organizaciji

Določitev skupine za zbiranje

Za člane skupine morajo biti izbrani tisti delavci, ki se zavedajo problema varnosti in bodo lahko rezultate uporabili za njeno izboljšanje. Vodstvo organizacije mora poudariti važnost in cilje zbiranja informacij, da tudi ostali začnejo na varnost gledati z drugačnimi očmi.

Idealna ekipa je sestavljena iz treh do sedmih ljudi, število članov pa naj bo zaradi morebitnega preglašovanja liho.

Določitev zahtev in ciljev zbiranja

Na prvem sestanku skupine se morajo določiti zahteve in cilji zbiranja. Dve vrsti ciljev sta posebej pomembni: (1) zbrati informacije o trenutnih ukrepih za varnost in (2) zbrati informacije o učinkovitosti odkrivanja in preprečevanja varnostnih incidentov. Zahteve morajo odgovoriti na vprašanja Zakaj, Kaj, Kdo, Kdaj, Kje in Kako lahko pride do napadov na informacijski sistem.

Načrt zbiranja informacij

Pri postavitvi varnostnega sistema je potrebno poskrbeti za to, da se informacije o njegovi kvaliteti zbirajo med njegovim rednim delovanjem in da niso potrebni naknadni in dodatni ukrepi za analizo njegovega delovanja. Če pa se zgodi, da povratni mehanizmi še ne delujejo, se gleda na eni strani dejansko stanje (kdo je odgovoren, kateri viri so varovani, katere metode se uporabljajo), po drugi strani pa tudi, kakšno je stališče zaposlenih. Če se nevarnosti incidenta zavedajo, potem je verjetnost, da bo do njega prišlo, manjša.

Učenje članov skupine

Skupina, ki bo zbirala informacije, in tudi ljudje, ki bodo informacije nudili, bi morali biti seznanjeni s tem, kaj se od njih pričakuje. Najmanj, kar morajo vedeti, je, da se zbiranje sploh opravlja, kako zbrati potrebne informacije in jih ločiti od nepotrebnih ter kakšna točnost zbiranja je potrebna.

Zbiranje informacij

Zbiranje mora biti končano v najkrajšem možnem času, pomeni, v nekaj delovnih dneh. Odgovornost, da je zbiranje v tem času dejansko končano, pade na člane skupine.

Analiza in poročilo o stanju v organizaciji

Rezultat analize stanja v organizaciji mora biti pregled, kateri elementi varnosti v organizaciji pomenijo največjo potencialno nevarnost. Najboljše analize so tiste, ki potrjujejo mnenja članov skupine, ki so jih dobili ob samem zbiranju. Rezultati, ki nasprotujejo mnenju naročnika študije, bodo zelo težko dobili zeleno luč. Zato je bolje, da so usmerjeni k nadgradnji obstoječega sistema kot njegovi zamenjavi.

Ko smo informacije zbrali, imamo podlago za izboljšanje programa - trenutno stanje lahko služi kot primerjava. Študija služi dvema namenoma: (1) Program za varnost usmerja od besed k dejanjem. Čeprav nekatera dejstva temeljijo na odnosu ljudi, so statistična podlaga za nadaljnje študije. (2) Meri učinkovitost sprememb. Če se koristnost sprememb ne da izmeriti, nikoli ne vemo, ali je bila sprememba dovolj učinkovita.

2.3 Ovrednotenje stroškov varovanja informacijskih sistemov

Ocenjene izgube, povezane z varnostjo podatkov in informacijskih sistemov v celoti, so zelo velike (v ZDA so ocenjene na več kot 40 milijard USD letno (povzeto po [1])). Glavna komponenta cene varnosti je cena nadzornih ukrepov in v zvezi s tem se pojavljata dve vprašanji: (1) Ali bi bilo ceneje, če bi izgubili del podatkov kot pa plačali za nadzor in (2) Ali so stroški, ki nastajajo zaradi varnosti, upravičeni - ali se varnost izboljšuje ali ne?

Strošek varnosti v organizaciji je ponavadi neznan. En del gre v projekte, en del v skupno učenje delavcev, nekaj za sprotno delo ... Vsebuje pa tri specifične komponente: (1) preventivne stroške, (2) stroške odkrivanja incidentov in (3) stroške izgub. Vsota vseh treh komponent je skupen strošek organizacije.

Preventivni stroški

Preventivni stroški so stroški, zaradi katerih se incident sploh ne zgodi. Vsebujejo postavitev odgovorne osebe za varnost, šolanje zaposlenih, preverjanje varnosti, razvoj varnih programov, shranjevanje računalniških medijev in nabavo sistema za varnost podatkov. Cilj te vrste stroškov je oteževanje možnega incidenta.

Stroški odkrivanja incidentov

Ti stroški so namenjeni odkritju incidenta (ko se je ta že zgodil) preden lahko pride do kakšne večje škode. Ukrepi vključujejo postavljanje gesel, preverjanje, če je zahtevana akcija dovoljena, opremo za nadzor, kontrolo dostopov in nenazadnje tudi fizično kontrolo. Kljub incidentu so varnostni ukrepi še vedno dokaj učinkoviti, saj denimo brez gesla vdiralec ne more priti nikamor. Ponavadi so preventivni stroški in stroški odkrivanja incidentov povezani med seboj.

Na primer, mnogi terminali so nastavljeni tako, da

se po nekaj napačnih vnosih gesla zaklenejo in nadaljnji dostop do sistema nekaj časa ni več možen.

Stroški izgub

Izgube podatkov so posledica nepravilnih oz. nezadostnih preventivnih ukrepov in ukrepov odkrivanja. Izgube so dveh vrst: fizične izgube (npr. izguba traku, CD-ROMa) in programske izgube (izguba podatkov, manipulacija s transakcijami).

2.4 Izračun stroškov varnosti

Koncept stroškov varnosti ponavadi zajema čim cenejše ukrepe, ki bi preprečili čimveč možnih različic vdorov v sistem, se pravi zmanjšanje verjetnosti vdora na določeni točki.

Eden takih zelo enostavnih in učinkovitih ukrepov je na primer zmanjšanje števila zaposlenih, ki imajo dostop ali pa celo zmanjšanje vstopnih točk v sistem. Ko je koncept določen, ostane končna odločitev na vodstvu, ki zanjo tudi odgovarja.

Stroški varnosti vsebujejo stroške izgub in stroške nadzora. Odločitev o nakupu sistema varnosti postane zelo lahka, ko so stroški enkrat znani. Večina organizacij napravi bilanco enkrat letno, zato se enkrat letno tudi odloča o velikosti proračuna. Takrat je idealen trenutek za določitev stroškov varnosti informacijskega sistema. Stroške protiukrepev lahko izračunamo precej enostavneje kot pa stroške morebitnih groženj. Izziv osebju je torej čim točnejša ocena stroškov morebitnih varnostnih incidentov. Na sliki 1 so trije primeri, ki

ponazarjajo razmerje med izgubami zaradi vdorov in stroški za protiukrepe.

Primer A kaže, da so stroški izgub precej manjši, kot pa bi znašali stroški kontrole aktivnosti. V tem primeru, tudi če smo se pri oceni izgub zmotili za 50 %, bomo še vedno raje utrpeli izgubo, ker so stroški kontrole previsoki. Primer B je nasprotje primera A. Tudi v tem primeru pa velja, da če smo se lahko zmotili za 50 %, je odločitev še vedno nedvoumna. Zadnji primer odgovorne sili v zadrego, saj se stroški izgub in kontrole med seboj prekrivajo. Če so stroški izgub previsoko in stroški kontrole prenizko ocenjeni, potem bi bila kontrola ekonomsko upravičena, v obratnem primeru pa ne.

Ti trije primeri kažejo, da se morajo stroški izračunati v dveh korakih. V prvem koraku naj bo izračun bolj površen (dovolj dober za primera A in B), če pa pride do primera C, naj se v drugem koraku natančnost poveča. Edini pogoj za izračun na tem nivoju pa je, da je natančnost pri obeh izračunih enaka, ne da se stroški kontrole izračunajo do zadnjega tolarja, stroški izgub pa ostanejo v milijonih.

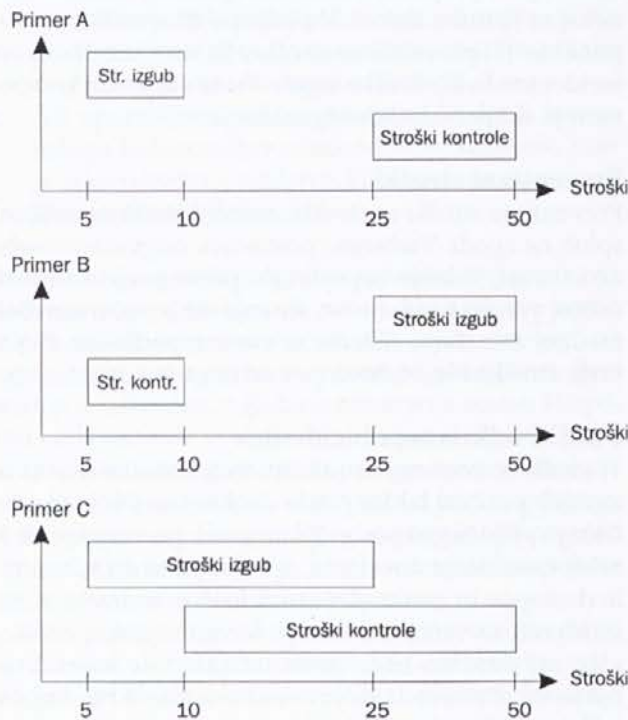
2.4 Stroški izgub in protiukrepi

Tehnologijo izračuna potencialnih izgub zaradi varnosti že stoletja uporabljajo zavarovalnice. Spremenljivki računa sta frekvenca pojavljanja izgube in enkratna izguba (izguba ob enem vdoru). Rezultat bo pričakovana letna izguba in se izračuna kot:

$$\begin{aligned} \text{Pričakovana letna izguba} &= \\ &= \text{frekvenca pojavljanja} \times \text{enkratna izguba} \end{aligned}$$

Kako si s to formulo lahko pomagamo? Denimo, da smo banka, ki posluje s kreditnimi karticami. Ena od možnih izgub je nesolventnost stranke - stranka kartico uporabi za nakup, vendar nima pokritja. Izgube lahko nastanejo tudi zaradi ponarejenih ali ukradenih kartic. Izračuna se po zgornji formuli, za vrednosti spremenljivk pa se uporabijo podatki o prejšnjih, recimo ukradenih karticah. Povprečna izguba pri teh karticah je bila 10.000 SIT in če predvidevamo, da se bo to zgodilo 5000-krat letno, je pričakovana letna izguba 50 milijonov SIT.

Izračun pričakovane letne izgube je orodje, s katerim si lahko pomagamo pri oceni stroškov za varnost in ekonomičnost nakupa. Po drugi strani pa je zelo težko pravilno oceniti izgube, ki nastanejo zaradi kraje računalniških programov in podatkov. Vrednost programa se ponavadi grobo izračuna kot 30\$ - 50\$ na programsko vrstico izvorne kode, vrednost podatkov pa bodisi kot strošek za nadomestitev ukradenih podatkov (npr. ponovno vnašanje imen in priimkov strank) bodisi kot vrednost izgube posla, če je škoda na podatkih nepopravljiva. V drugem primeru je vrednost izgubljenih podatkov seveda precej višja.



Slika 1. Stroški izgub proti stroškom kontrole

Za zmanjšanje frekvence pojavljanja izgube obstajata dve metodi: (1) zmanjšanje priložnosti izgube z omejevanjem dostopa do podatkov ljudem, ki jih ne potrebujejo in (2) zmanjšanje verjetnosti izgube, denimo s preverjanjem vnosa ključev - s tem se verjetnost izgube zmanjša na kakovost programa za preverjanje ključa. Tudi enkratna izguba se lahko zmanjša in to bodisi z zavarovanjem, ki izgubo v celoti pokrije, bodisi z decentralizacijo, na primer razdelitvijo računalniškega centra na dve enoti, tako da katastrofa v eni enoti ne vpliva na delo v drugi.

Stroški preventivnih in kurativnih protiukrepev morajo biti v sorazmerju s stroški izgub. To so stroški za njihov razvoj ter sami operativni stroški. Prvi so enkratni (zato jih je tudi lažje izračunati), medtem ko je treba na druge računati vsakič sproti in so odvisni tudi od števila poskusov vdora.

V tem poglavju sem opisal proces ocene izgub v primerjavi z verjetnostjo vdora v sistem in posledic kontrole aktivnosti, ki izgube zmanjšujejo. Proces se lahko ponavlja toliko časa, da se dobi optimalna mešanica in so kontrole najučinkovitejše. Podrobneje je postopek ugotavljanja velikosti problema varnosti opisan v [3].

3. Kako napraviti sistem varen?

Program za varnost informacijskega sistema naj bo, če je le mogoče, zaključen del širšega sistema, ki varuje celotno organizacijo. Nima namreč smisla upravljati velike vsote denarja za zaščito enega dela, če je na drugi strani organizacije do informacij možno priti brez posebnega napa, ker v nekem oddelku zaščite sploh nimajo. Mnogo vodilnih oseb trpi za ti. "bankirskim sindromom". Ko bankir odpre novo banko, zgradi v kleti trezor, zaščiten z debelim jeklenim zidom, vhod vanj je možen samo na en način, pa še za to je treba poznati več kombinacij varnostnih kod. Ta trezor pa pomeni le 0.5 % celotnega premoženja, 99.5 % pa je v računalniku, do katerega je možen dostop prek terminalov. Kriminalci ta sindrom poznajo in si rečejo: "Zakaj bi tvegali življenje za drobiž iz trezorja, če lahko to precej lažje naredim s pritiskom na nekaj tipk".

Varovanje organizacije se mora vedno začeti na vrhu, nato pa se polagoma spuščati do posameznih enot. Varovanje računalniških virov sicer jamči za več kot povprečno zaščito, vendar naj bi to vseeno ne bil edini vir, vreden pozornosti.

3.1 Strategija vodenja in ključni faktorji uspeha

Prvi korak k programu varnosti je izjava vodstva o nameri in potrebi. Nato je dobro, če vprašamo o njihovih vidikih tudi zaposlene, kajti program ne bo dosti pomagal, če zaposleni sprejemajo varnost kot nebodi-

gatreba in vsakič ko pride do posla, ki ga je treba napraviti hitro, pozabijo na varnost. Šele takrat, ko so morebitni spori z zaposlenimi rešeni, pride trenutek za izdelavo strategije.

Strategija seveda ne more biti enaka za vse organizacije; razlikuje se glede na dejavnost, s katero se organizacija ukvarja, želje vodstva in nenazadnje tudi na zakone, ki se nanašajo na varnost. Predlagana strategija varnosti ne sme ščititi samo računalniških virov, ampak celotno organizacijo (varnostni ukrepi v trezorju so še vedno nujni).

Kriteriji, s pomočjo katerih vodstvo lahko oceni pravilnost strategije, se imenujejo ključni faktorji uspeha (Critical Success Factors). Ključni faktorji uspeha za program računalniške varnosti organizacije se morajo osredotočiti na naslednja štiri področja:

- Uporabnost sistema glede na velikost in porazdeljenost sistema viri vključujejo strojno in programsko opremo, podatke in dokumentacijo, komunikacije, okolje in vzdrževanje. Izguba podatkov napravi sistem v celoti neuporaben, medtem ko pri izgubi komunikacij sistem še vedno deluje, le oddaljeni računalniki ostanejo mrtvi.
- Celovitost sistema. Izraz "celovitost sistema" se najpogosteje uporablja za podatke, na katerih sistem deluje. Zmanjšanje celovitosti ima lahko različne posledice, od napačne akcije zaradi nepravilnih podatkov pa celo do zaustavitve delovanja.
- Zaupanje v sistem. Izguba zaupanja je najresnejša posledica nezanesljivega sistema. Do tega pride zaradi naključne ali namerne prekinitve delovanja ali zaradi nepooblaščenega dostopa. Te vrste izgub so ponavadi obravnavane prioritarno, saj rešitev tega problema večkrat reši tudi kakšnega od ostalih.

Ključni faktorji uspeha za računalniško varnost morajo imeti štiri osnovne značilnosti. Prvič, uresničevati morajo resnični namen vodstva organizacije. Če vodstvo v faktorje ne verjame, potem ne bo niti podprlo namestitve programa niti ga ne bo zanimalo, če je bil dosežen kakšen uspeh. Drugič, biti morajo dokumentirani, sicer podrejeni ne bodo imeli od njih nobene koristi pri namestitvi. Tretjič, biti morajo merljivi. Če pomena faktorjev ne moremo izmeriti, potem so neuporabni. In četrtič, kriteriji morajo biti dosegljivi. Na primer, faktor, ki pravi, da nikakor ne sme priti do vdora v sistem, je nedosegljiv.

Naloga vodstva, ko so ključni faktorji uspeha podani, je, da jih preuči in napravi plan, ki je ponavadi sestavljen iz petih točk:

1. **Postavitve odgovorne osebe za varnost.** Namen postavitve odgovorne osebe je v tem, da postane odgovoren za varnost posameznik. Veliko organizacij je poskušalo s skupino ljudi, pa se ni obneslo. Naloge odgovorne osebe so: definiranje ključnih

faktorjev uspeha, zagotovitev, da ima vsak oddelek nekoga, ki je odgovoren za varnost, analiza in predlog aktivnosti ter potrebnega denarja za naslednje leto in zagotovitev usposabljanja novih zaposlenih.

2. **Izbira skupine, ki se bo ukvarjala z varnostjo.** Ker je računalniška obdelava podatkov prisotna v celotni organizaciji, bi bilo nerealno pričakovati od ene same osebe, da lahko primerno planira, namešča in nadzoruje potrebe po varnosti na vseh področjih organizacije. Zato se ponavadi določi skupina uporabnikov (liho število), ki skrbijo za celotno organizacijo.
3. **Določitev kriterijev za oceno programa varnosti.** Kriteriji za ocenjevanje so nujni zato, da se lahko preveri, če je informacijski sistem tako varen, kot je potrebno. Kvaliteta programa je odgovornost tistih, ki so program namestili.
4. **Zagotovitev denarja za podporo prvih treh aktivnosti.** Mnogi programi za varnost trpijo pomanjkanje sredstev. Vodstvo ponavadi sicer stoji za programom, pri denarju pa se vse skupaj ustavi. V varnosti pa velja: "Dobiš tisto, kar plačaš." Računalniška varnost mora opravičiti svoj obstoj, ko se to enkrat zgodi, pa se morajo zanjo najti tudi denarna sredstva.
5. **Osebnost angažiranost v programu.** Vodstvo organizacije mora ne samo verjeti v program varnosti, ampak biti vanj tudi osebno vključeno. Če vodilna oseba pride na sestanek o varnosti, prikazuje rezultate delničarjem itd, potem vsi vedo, da je varnost za organizacijo pomembna in se navadno tako tudi obnašajo. Čas je za vodilno osebo najpomembnejši vir in mora biti smotno porabljen, toda ena ura mesečno je nekakšen minimum, ki lahko zaposlene prepriča, da je varnost pomembna.

3.2 Ustvarjanje motiviranosti med delavci.

Nelogično in neupravičeno bi bilo pričakovati, da bodo zaposleni avtomatično pomagali pri vzpostavitvi sistema varnosti, takoj ko se bo vodstvo za to odločilo. Motiviranost pa se s skrbno načrtovanim planom lahko ustvari in s tem doseže namen.

Ustvarjanje motiviranosti vsebuje **princip spremembe vedenja**, pomeni, da bo učenec sposoben napraviti neko novo dejanje.

Ljudje imajo do varnosti različne poglede. Nekateri so skrbni in želijo varovati premoženje delodajalcev, medtem ko drugi mislijo češ, saj so bogati in ne bodo pogrešali nekaj ukradenega premoženja. Znanstveniki so se ukvarjali s spremembo vedenja precej časa, da so prišli do ugotovitve, da velja naslednja formula [3, stran 92]:

$$\begin{aligned} & \text{Sprememba vedenja} = \\ & = \text{sprememba posameznika} + \text{sprememba okolja} \end{aligned}$$

Formula kaže, da je vedenje posameznika odvisno od njegovega odnosa in od vpliva okolja, v katerem se nahaja.

Ustvarjanje motiviranosti je sprememba vedenja. Dosežemo jo lahko tako, da spremenimo posameznika, okolje ali celo oboje. Psihologi trdijo, da je precej lažje spremeniti okolje kot posameznika. Temu pravijo "mafijski sindrom", kjer človeka okolje (množica drugih ljudi, ki delajo isto) naravnost sili k temu, da napravi kriminalno dejanje. Torej bomo najprej poskušali spremeniti okolje tako, da bo naklonjeno varnosti (povabilo delavca na tečaj, kosilo z direktorjem - naj se delavec počuti pomembnega), nato pa še posameznika.

Naštel bom pet principov sprememb vedenja, ki lahko ustvarijo motiviranost ljudi za varnost in jih obravnavajo kot aktivne subjekte:

1. **Delegiranje odgovornosti navzdol.** Delegiranje odgovornosti navzdol ima dva cilja - obogatitev dela podrejenih, da se čutijo pomembnejše in več prostega časa vodstva, ki se lahko posveti drugim stvarim. Metode, ki se jih moramo držati pri delegiranju odgovornosti, so določitev nalog (vse naloge morajo biti neodvisne, da jih lahko opravi en sam človek), vedeti je treba, katera znanja so potrebna za izvršitev naloge (naloge ne sme biti dana človeku glede na položaj v organizaciji) ter kateri človek še ima dovolj znanja za izpolnjevanje nalog varnosti.
2. **Odgovornost posameznika za varnost.** Lastništvo varnosti je ena ključnih lastnosti spremembe vedenja posameznika. Taka odgovornost pomeni, da bodo zaposleni imeli važno vlogo pri določanju, kakšne vrste varnost je potrebna in seveda tudi pri sami namestitvi. Vodstvo se bo sicer še vedno odločilo, kaj je potrebno, zaposleni pa bodo predlagali, kako se bo to tudi dejansko uporabljalo. Ponavadi se sestavijo skupine nekaj ljudi, ki skupaj najdejo rešitev določenega problema in jo predstavijo vodstvu. Le-to si mora vzeti vsaj eno uro tedensko na člana take skupine, sicer pa je odgovornost v celoti na strani skupine.
3. **Osebnostna podpora rezultatov dela.** Ena od težav pri varnosti je, da se nikoli ne ve, ali je bilo kaj narejenega. Recimo, nekdo, ki veliko časa vložil v izboljšave varnosti, je kritiziran, ker porabi preveč procesorskega časa, ne dobi pa nobene podpore svojemu delu s strani vodstva. Za spremembo vedenja pa je zelo pomembno, da posameznik sam ve, da dela nekaj koristnega - v to pa ga mora prepričati nadrejeni s spodbujanjem in svojo podporo rezultatom.
4. **Sistem nagrajevanja.** Ljudje naj bodo nagrajeni, če delajo tisto, kar vodstvo želi. Če je večja varnost v interesu organizacije, potem naj bodo ljudje, ki se z njo ukvarjajo in dosegaajo rezultate, primerno nagrajeni. Nagrade niso nujno denarne, saj je dovolj

nagrad, ki delavcu pokažejo, da je naredil nekaj koristnega za organizacijo, npr. pohvala pred ostalimi delavci, nagradno udeleževanje seminarjev, napredovanje v službi ...

5. **Osební nadzor vodstva.** Prevečkrat so naloge dane delavcem na način, ki pomeni bodisi, da bodo "splavali" (napravili nalogo) ali pa "utonili". Na žalost pa to ni pravi način, ki bi povečeval motiviranost, saj je stalno prisoten strah, kaj pa če naloge ne bom naredil. Zaposlenega je treba najprej naučiti, kaj naloga sploh zahteva in se prepričati, da ima dovolj znanja za njeno rešitev. Tudi potem je treba vsaj na toliko in toliko časa pogledati, kako mu gre od rok in mu morebiti pomagati. Za to je ponavadi odgovoren vodja varnosti.

(se nadaljuje)

Literatura :

- [1] M.A.L. Farr : *Security for Computer Systems*, Hotspur Press, 1974
- [2] Jan Hruska : *Computer Security Solutions*, Blackwell Scientific Publications, 1990
- [3] William E. Perry : *Management Strategies for Computer Security*, Butterworth Publishers, 1985
- [4] D.W. Davies : *Security for Computer Networks*, John Wiley & Sons, 1984
- [5] Paul J. Fortier : *Handbook of LAN Technology*, McGraw-Hill, 1992
- [6] Mario Korva : *Šifriranje - kript algoritmi*, CIFRA d.o.o., 1994
- [7] Roger M. Needham : *Denial of service*, *Communications of the ACM*, November 1994

◆
 Tomaž Poštuvan je diplomiral leta 1993, trenutno pa je zaposlen kot mladi raziskovalec na Fakulteti za računalništvo in informatiko. Njegovo delovno področje so prevajalniki (v tem okviru tudi pripravlja magistrsko delo), sicer pa se je precej ukvarjal tudi z varnostjo oz. zaupnostjo podatkov v računalniških sistemih.
 ◆

Vabilo avtorjem

Uredniški odbor revije *Uporabna informatika* načrtuje razširitev obsega revije oziroma večjo pogostost izhajanja. Rezultati ankete med bralci revije so pokazali, da si želijo med drugim prispevke z naslednjih področij: ocena orodij in različnih rešitev, predstavitev primerov iz prakse, predstavitev rešenih informacijskih problemov, pregled stanja na nekaterih področjih v Sloveniji.

S tem vabilom se posebej obračamo na informatike v praksi, da s članki v naši reviji predstavijo svoje ugotovitve in izkušnje.

Navodila za prispevke objavljamo na zadnji strani revije. Prispevke bomo začeniši z letošnjim letnikom tudi honorirali.

POJMOVNIK

KAKOVOSTI PROCESA PROGRAMSKE OPREME

Pričujoči pojmovnik je nastal v okviru projekta PROCESSUS - Uvajanje ocenitve in dvig kakovosti razvoja informacijskega sistema. Projekt je poleg številnih podjetij finančno podprlo tudi Ministrstvo za znanost in tehnologijo. Osnovni namen projekta je bil izdelati metodologijo za ocenitev in izboljšanje kakovosti procesa razvoja programske opreme. Izdelava pojmovnika je bila nujno potrebna, ker je nedvoumno razumevanje pojmov neobhodno za uspešno komunikacijo med posamezniki, ki so na kakršenkoli način sodelovali v projektu. Služil je kot del osnovne baze znanja, na katerem temeljijo rezultati projekta.

V pojmovniku imamo danes več kot 260 pojmov v angleškem jeziku, uporabljeno slovensko poimenovanje ter ustrezno obširnejšo razlago vsakega izmed pojmov. Zajeta množica pojmov nikakor ne more biti dokončna, vsekakor se bo dopolnjevala v skladu z razvojem stroke, prepričani pa smo, da zajema tisto minimalno množico pojmov, ki omogoča nedvoumno sporazumevanje na področju kakovosti procesa razvoja programske opreme. Seveda bodo poleg dopolnitve s posameznimi pojmi potrebne še kakšne spremembe, predvsem pri slovenskem poimenovanju. Zato naj objava pojmovnika služi kot vzpodbuda tistim, ki so se ali pa se še bodo srečali s problemom slovenskega izrazoslovja na področju kakovosti, da bodo iskali in morda tudi našli ustrežnejšo slovensko poimenovanje za posamezne pojme. Kakršnakoli mnenja in ideje pa so dobrodošli tudi članom delovne skupine, ki so že do sedaj delali pri pričujočem pojmovniku.

Člani delovne skupine: Tomaž Domajnko, József Györkös, Marjan Heričko, Špela Hleb, Marjan Krašna, Ivan Lah, Ivan Rozman, Bruno Stiglic, Romana Vajde Horvat, Tatjana Welzer Družovec, Aleš Živkovič

aktivnost / activity

Vsak opravljeni korak ali izvedena funkcija (tako mentalna kot fizična), potrebna za napredovanje k določenemu cilju. Aktivnosti vključujejo tisto delo vodstva (menedžerjev) in tehničnega osebja, ki je potrebno za izvedbo določene naloge v sklopu projekta ali organizacije. /CMM-93/ (Glej tudi pojem naloga - task, zaradi razjasnitve različnih pomenov obeh pojmov.)

analiza Pareto / Pareto analysis

Analiza hib z razvrščanjem od najpomembnejših do najmanj pomembnih. Temelji na principu, da majhen delež možnih vzrokov povzroči večino možnih posledic (npr. 80% posledic izvira iz 20% možnih vzrokov). /CMM-93/

* Analiza je poimenovana po Vilfredu Paretu, ekonomistu iz 19-ega stoletja.

analiza vzrokov / causal analysis

Analiza hib, s katero določimo prikrite in globoko ukoreninjene vzroke za pojavljanje teh hib. /CMM-93/

aplikacijska domena / application domain

Povezana množica sistemov, ki so med seboj v relaciji (npr. sistemi, ki se nanašajo na določen tip problemov). Razvoj in vzdrževanje v aplikacijski domeni običajno zahteva specialno strokovno znanje in/ali vire. Nekaj primerov aplikacijskih domen: plačilni in kadrovski sistemi, nadzorni sistemi, prevajalniki, ekspertni sistemi. /CMM-93/

arhitektura programske opreme / software architecture

Organizacijska struktura programske opreme ali modula. /IEEE-610/

arhiv o razvoju programske opreme / software development archives

Urejena dokumentacija (lahko v obliki map, zbirk ali avtomatiziranega okolja), ki jo je mogoče uporabiti v skladu z načeli upravljanja konfiguracije. /PROCESSUS/

atribut / attribute

Pri metrikah lastnost ali značilnost entitete, ki jo lahko običajno merimo neposredno. /AMI/

baza podatkov o procesu / process database

Baza podatkov o procesu je repozitorij, v katerem hranimo vse podatke o procesu. To so centralizirani viri, s katerimi upravlja procesna skupina. Centraliziran nadzor nad to bazo zagotavlja, da hranimo in zaščitimo podatke o procesu v vseh projektih. /CMM-93/ (Glej tudi organization's software process database - baza podatkov o procesu v organizaciji)

baza podatkov o procesu razvoja programske opreme v organizaciji / organization's software process database

Baza podatkov za zbiranje in nudenje podatkov o procesu in rezultirajočih delovnih produktih, predvsem tistih, ki se navezujejo na standardiziran proces v organizaciji. Baza podatkov vsebuje tako dejanske podatke meritev kot tudi potrebno znanje za njihovo razumevanje in ocenitev njihove sprejemljivosti in uporabnosti. Primeri podatkov o procesih in delovnih produktih: ocenitev obsega programske opreme, truda in stroškov, podatki o produktivnosti, podatki o internih pregledih, število in natančen opis hib, najdenih v programski kodi. /CMM-93/

cilj / goal

Povzetek ključnih postopkov na ključnih procesnih področjih, s katerimi lahko določimo, ali je organizacija učinkovito izvedla ključno procesno področje. Cilji označujejo obseg, meje ter namen vsakega ključnega procesnega področja. /CMM-93/ Cilj je merljiva tarča, ki jo postavi organizacija. /AMI/. Metoda AMI postavlja primarne cilje in podcilje. Primarni cilji so visokonivojski, podcilje pa razvijemo na podlagi analize visokonivojskih ciljev. Cilje razvrstimo glede na njihov namen: cilji za "znanje in vire" (opazovanje, ovrednotenje, predikcija) in za "spremembo ali uspeh (dosežek)" (povečanje, zmanjšanje, stabilizacija)

cilj kakovosti programske opreme / software quality goal

Kvantitativna definicija ciljev za kakovost delovnih produktov programske opreme. /CMM-93/

ciljni računalnik / target computer

Računalnik, na katerem bo delovala dobavljena programska oprema. /CMM-93/ (Glej glavni računalnik za razliko.)

ciljno okolje / target environment

Okolje, v katerem bo sistem oz. proizvod programske opreme deloval po instalaciji. Srečamo tudi sinonime: okolje delovanja - operational environment, uporabnikovo okolje - user environment. /PROCESSUS/

CMM / CMM

Akronim za *Capability Maturity Model* - zmožnostno zrelostni model.

člen akcije / action item

1. Enota v seznamu, ki je bila skupini ali posamezniku dodeljena za izvedbo. /CMM-93/
2. Sprejeti predlog akcije. /CMM-93/

definirani proces (razvoja programske opreme) projekta / project's defined software process

Operativna definicija procesa, uporabljenega pri projektu. Definirani proces razvoja programske opreme je dobro opisan in razumljiv proces ter je v skladu s standardi programske opreme, postopkov, orodij in metod. Razvit je na podlagi prireditve standardov organizacije, opravljene tako, da se prilega karakteristikam projekta. (Glej tudi standardni proces v organizaciji - organization's standard process, učinkovit proces - effective process in dobro definiran proces - well-defined process.) /CMM-93/

dejanska napaka / actual error

Napaka, ki se pojavi ob izvajanju programa. /PROCESSUS - samo za SEI/

delujoča programska oprema / operational software

Programska oprema, ki je namenjena za uporabo in delovanje na uporabnikovem sistemu po dostavi in vgradnji v predvideno okolje. /CMM-93/

diagram komuniciranja / communication diagram

Risba, ki predstavlja tok informacij med ljudmi v nekem delovnem okolju. Diagram je uporaben v različnih korakih metode AMI za preverjanje primernosti izbranih ciljev in metrik v smislu informacijskih tokov v obstoječem okolju ter za pomoč pri identificiranju problemov, ki bi se kasneje lahko pojavili. /AMI/

discipliniran proces / disciplined process

Proces, ki je natančno predpisan in ki je izvajan v skladu s temi predpisi. /PROCESSUS/

dobro definirani proces / well defined process

Proces, ki vključuje kriterije pripravljenosti za izvajanje aktivnosti, vhode, standarde in postopke za izvajanje dela, mehanizme verifikacije (kot npr. interni pregledi), izhode in kriterije dokončanja. /CMM-93/ (Glej tudi učinkoviti proces - effective process.)

dodelitev dela / work assignment

Določitev oseba za izvajanje določenih nalog in aktivnosti. /PROCESSUS/

dodeljene zahteve / allocated requirements

(Glej sistemske zahteve dodeljene programski opremi - system requirements allocated to software.)

dogodkovno vodeni pregled/aktivnost / event driven review/activity

Pregled ali aktivnost, ki se izvaja glede na pojavljanje posameznih dogodkov znotraj projekta (npr. formalni pregled ali dokončanje določene stopnje v življenjskem ciklu). /CMM-93/ (Glej periodični pregled/aktivnost - periodic review/activity za razliko.)

dovolilnica za izvzetje / concession; waiver

Pisno dovoljenje za uporabo ali dobavo določene količine proizvedenega materiala, sestavnih delov ali zalog, ki niso v skladu s postavljenimi zahtevami. /SLS ISO 8402 : 1993/ V SEI CMM nastopa dovolilnica za izvzetje iz obveznega usposabljanja - glej oprostitev usposabljanja.

dovolilnica za odstopanje / production permit; deviation permit

Pisno dovoljenje za odstopanje od specificiranih zahtev, ki se izda pred začetkom proizvodnje ali pred izvedbo storitve in ki velja za določeno količino ali za določeno dobo. /SLS ISO 8402: 1993/

drevo ciljev / goal tree

Vizualni pripomoček za prikaz ciljev in podciljev. /AMI/

element konfiguracije / configuration item

Skupek strojne ali programske opreme ali kombinacije obojega, ki je vključen v upravljanju konfiguracije in je obravnavan kot samostojna entiteta v procesu upravljanja konfiguracije. /IEEE-610/

element odvisnosti / dependency item

Produkt, akcija, informacija ipd., ki jo mora določen posameznik ali skupina zagotoviti drugemu posamezniku ali skupini, če naj le-ta dokonča planirano nalogo. /CMM-93/

element procesa razvoja programske opreme / software process element

Sestavni element opisa procesa. Vsak element pokriva dobro definirano, povezano množico nalog, ki so hkrati v tesni medsebojni relaciji (npr. element ocenjevanja programske opreme, element načrtovanja programske opreme, element kodiranja, element internih pregledov). /CMM-93/ V PROCESSUSU je enakovredni termin standardni postopek.

element programske opreme / software item

Vsak prepoznaven del proizvoda programske opreme v vmesni ali končni fazi razvoja. /ISO 9000-3/

element sistema kakovosti / quality system element

Področja, ki jih je pri vodenju kakovosti potrebno zajeti. (npr. odgovornost vodstva, načela sistema kakovosti, presoja sistema kakovosti, ekonomika - obravnavanje stroškov kakovosti, pregled pogodbe, itd.) /PROCESSUS/

enota / unit

1. Element, ki ga lahko ločeno testiramo in je specificiran v načrtu računalniških programskih komponent.
2. Logično ločljiv del programske opreme.
3. Komponenta programske opreme, ki ni dalje razdeljena na podkomponente. /IEEE-610/

entiteta / entity

Dobesedno je to "stvar", produkt ali aktivnost. V viru /AM/ so entitete razvrščene kot viri, procesi ali produkti. Vir je osredotočen na razvojni postopek, vendar obstaja še niz drugih pomembnih entitet, kot so proces končne uporabe, specifikacija zahtev, inženirsko učenje in drugo.

ESPRIT / ESPRIT

"European Strategic Program for Research and Development in Information Technology".

faza (življenjskega cikla) / life-cycle phase

Skupek logično povezanih aktivnosti, ki jih je potrebno izvesti v okviru razvoja programske opreme (npr. aktivnosti, povezane s testiranjem so združene v fazi testiranja). Zaporedje izvajanja faz je odvisno od uporabljenega življenjskega cikla.

/PROCESSUS/

formalni pregled / formal review

Formalna seja, na kateri uporabnikom, stranki ali drugim zainteresiranim predstavimo izdelek ter sprejemamo komentarje in odobritve. Pregledamo lahko menedžment, tehnične aktivnosti ali napredovanje projekta. /CMM-93/

formalni proces / formal process

Dokumentiran niz korakov z navodili za uporabo. /CMM-93/

funkcija / function

Množica akcij, ki so v medsebojni povezavi in ki jih prevzamejo posamezniki (ali orodja), ki so specifično določeni (oz. prirejeni) za svoje vloge. /CMM-93/

funkcijsko testiranje / functional testing

Je vrsta testiranja, pri katerem program obravnavamo kot črno skrinjo. Objekt testiranja so funkcije programa. /DOG-93/

glavni pogodbenik / prime contractor

Posameznik, partner, podjetje ali družba, ki upravlja podpogodbe o načrtovanju, razvoju in/ali vzdrževanju enega ali več produktov. /CMM-93/

glavni računalnik / host computer

Računalnik, na katerem razvijamo programsko opremo. /CMM-93/

(Glej ciljni računalnik - target computer kot nasprotni primer.)

gostota hib / defect density

Kvocien števila odkritih hib v produktu in velikosti produkta (izraženo v standardnih enotah za ta produkt). /CMM-93/

hiba / defect

1. Pomanjkljivost sistema ali systemske komponente, ki povzroči, da sistem ali komponenta ne izvrši zahtevane funkcije. Hiba, odkrita v času izvajanja, lahko povzroči celo izpad celotnega sistema. /CMM-93/
2. Neizpolnitev zahtev za predvideno uporabo. /SLS ISO 8402 : 1993/

identifikacija konfiguracije / configuration identification

Element upravljanja konfiguracije, ki sestoji iz izbire elementov konfiguracije sistema in iz posnetka njihovih funkcionalnih in fizičnih karakteristik iz tehnične dokumentacije. /IEEE-610/

instalacija / installation

Aktivnost, pri kateri vgradimo programski proizvod v ciljno okolje. /PROCESSUS/

institucionalizacija / institutionalization

Institucionalizacija pomeni graditev takšne infrastrukture in skupne kulture, v katerih so metode, pravila in postopki podprti na tak način, da so del poslovanja organizacije tudi tedaj, ko organizacijo zapustijo tisti, ki so jih prvotno vpeljali ali definirali. /CMM-93/

inšpekcija / inspection

Formalna metoda preverjanja proizvodov razvoja programske opreme, pri kateri skupina, v kateri ne sme biti avtorja proizvoda, natančno pregleduje proizvod. /DOG-93/

integracija / integration

Je postopek združevanja komponent programske in strojne opreme v sistem kot celoto. /IEEE 729/

integracija programske opreme / software integration

Proces združevanja izbranih komponent programske opreme z namenom zagotovitve množice ali specificirane podmnožice možnosti, ki jih bo ponujal končni sistem programske opreme. /CMM-93/

integracijsko testiranje / integration testing

Testiranje modulov, združenih v celoto, z namenom, da se odkrijejo napake na njihovi povezavi. /DOG-93/

integrirano upravljanje programske opreme / integrated software management

Združitev in integracija aktivnosti upravljanja programske opreme in programskega inženirstva v koherenten (razumljiv in jasen) definirani proces. Temelji na standardnem procesu v organizaciji ter na vseh procesih, ki so v povezavi z razvijanim procesom in mu služijo kot opora. /CMM-93/

interne zahteve za verifikacijo / internal verification requirements

Zahteve za verifikacijske aktivnosti, ki si jih interno definira organizacija in ki jih upošteva pri procesu razvoja programske opreme. /PROCESSUS/

interni pregled / internal review

Pregled *programskih produktov*, ki ga po načelu definiranega postopka opravijo proizvajalci produkta z namenom identificiranja *hib* ter izboljšanja produkta. /CMM-93/

interni standard / internal standard

Je dokument, ki v *organizaciji* predpisuje *standarde* in *postopke*, ki jim je potrebno slediti pri izvajanju danega projekta. /PROCESSUS/

inženirska skupina / engineering group

Skupina posameznikov (menedžerjev in tehničnega osebja), ki zastopa določeno inženirsko disciplino. Primeri inženirske discipline so: sistemski inženiring, inženiring strojne opreme, preiskovanje sistemov, *programsko inženirstvo*, *upravljanje konfiguracije* programske opreme in *zagotavljanje kakovosti programske opreme*. /CMM-93/

izdelana programska oprema / software build

Operativna verzija programskega sistema ali komponenta, ki vključuje specifično podmnožico tistih zmožnosti, ki jih bo zagotavljal tudi končni *produkt*. /IEEE-610/

izstopni kriteriji / output criteria

Kriteriji, ki morajo biti izpolnjeni, da lahko zaključimo določeno fazo. /PROCESSUS/

izvedbena zmogljivost procesa / process performance

Meritev dejanskih rezultatov, ki so doseženi s sledenjem procesu. /CMM-93/ (Glej *zmožnost procesa - process capability* za razliko.)

izvedene aktivnosti / activities performed

(Glej *skupne lastnosti - common features*.)

izvedbena obveza / commitment to perform

(Glej *skupne lastnosti - common features*.)

izvor hibe / defect root cause

Skriti razlog (npr. nepopolnost *procesa*), ki omogoči vgraditev *hibe* v *produkt* ali *proces*. /CMM-93/

kakovost / quality

1. Stopnja, s katero *sistem*, komponenta ali *proces* zadovoljuje *specificirane zahteve*.
2. Stopnja, s katero *sistem*, komponenta ali *proces* zadovoljuje uporabnikove ali strankine potrebe oz. pričakovanja. /IEEE-610/
3. Skupek vseh lastnosti in karakteristik proizvoda ali storitve, ki se nanašajo na sposobnost proizvoda ali storitve, da zadovolji izražene ali pričakovane potrebe.

ključni postopki / key practices

Infrastruktura in *aktivnosti*, ki najbolj prispevajo k učinkovitosti izvedbi in institucionalizaciji *ključnega procesnega področja*. /CMM-93/

ključno procesno področje / key process area

Množica med seboj odvisnih *aktivnosti*, s katerimi dosežemo *cilje*, ki so pomembni za *vpeljavo zmožnosti procesa*. Definirane so tako, da se vse *aktivnosti*, ki so v določeno ključno področje *procesa* zajete, nahajajo na istem *zrelostnem nivoju*. To so področja, ki jih je SEI definiriral kot *osnovne verzije*, s katerimi lahko določimo *zmožnost procesa* v *organizaciji* ter

spoznamo *izboljšave*, ki jih je potrebno izvesti na procesu za prehod na *višji zrelostni nivo*. /CMM-93/

Ključna področja procesa za nivoje CMM so:

- **drugi nivo**: upravljanje zahtev, planiranje *projektov*, sledenje in nadzor projektov, upravljanje podpogodbenikov, zagotovitev kakovosti in *upravljanje konfiguracije* programske opreme.
- **tretji nivo**: procesni vidik, definiranje procesa, program usposabljanja, celovito upravljanje programske opreme, programsko inženirstvo, sodelovanje skupin, interni pregledi.
- **četrti nivo**: kvantitativne meritve procesa, *upravljanje kakovosti programske opreme*.
- **peti nivo**: *preprečevanje hib*, upravljanje sprememb tehnologije, upravljanje sprememb procesa.

knjižnica osnovnih verzij programske opreme / software baseline library

Vsebina repozitorija za shranjevanje *elementov konfiguracije* in pripadajočih zapisov. /CMM-93/

končni uporabnik / end user

Posameznik ali *skupina*, ki bo *sistem* uporabljala za opravljanje predvidenih operacij, ko bo *sistem* vgrajen v svoje okolje. /CMM-93/

konfiguracija / configuration

Funkcionalne in fizične karakteristike strojne ali programske opreme, kot so predpisane v tehnični dokumentaciji ali dosežene v *produktu*. /IEEE-610/

konfiguracijska enota / configuration unit

Entiteta najnižjega nivoja *elementa konfiguracije*, ali komponenta, ko ji lahko vstavimo ali izberemo iz *sistema knjižnice* za upravljanje konfiguracije. /CMM-93/

korektivni ukrep / corrective action

Ukrep, ki ga planiramo in izvajamo z namenom izboljšanja določenega stanja. /PROCESSUS/

kritična pot / critical path

Zaporedje medsebojno odvisnih *nalog* v sklopu *projekta*, ki jih je potrebno opraviti natančno znotraj plana, da bo celoten projekt potekal po *terminskem planu*. /CMM-93/

kritični računalniški vir / critical computer resource

Parametri računalniških *virov*, za katere predvidevamo, da bodo *vir tveganja* za *projekt* zaradi potencialne potrebe po večji kapaciteti, kot je dejansko na razpolago. Primer: pomnilnik *ciljnega računalnika*, kapaciteta diska na *glavnem računalniku*. /CMM-93/

kvantitativno vodenje / quantitative control

Vsaka kvantitativna ali statistično-zasnovana tehnika, ki je primerna za analizo *procesa*, identifikacijo posebnih vzrokov za *variiranje izvedbenih zmogljivosti procesa* in za *vmitev izvedbene zmogljivosti procesa* v okvir dobro definiranih omejitev. /CMM-93/

mehanizem / mechanism

Instrument ali tehnika, s katero je zagotovljena izvedba posla, procedure ali postopka. Mehanizem lahko vključuje različne organizacijske elemente in njegova dokumentacija lahko vsebuje kombinacije funkcionalnih izrazov, operativnih načrtov, opisov stanja in/ali *formalnih postopkov*. Dokumentacija definira, kaj naj bi bilo izvedeno, kako naj bi bilo izvedeno in kdo je odgovoren za rezultate. /CMM-93/

mejnik / milestone

Terminsko omejen dogodek, za katerega je zadolžen posameznik in s katerim merimo napredovanje dela. /CMM-93/

meritev / measurement

Ugotavljanje dimenzije, kapacitete, kvantitete ali vrednosti določenega atributa (npr. 300 vrstic izvorne kode, 7 strani dokumentacije, ipd.). /CMM-93/

merilo / measure

Enota, uporabljena pri meritvi (kot npr. število vrstic izvorne kode, ali število strani dokumentacije *načrtovanja*, ipd.). /CMM-93/

meritev procesa / process measurement

Množica definicij, *metod* in *aktivnosti*, ki so uporabljene za izvedbo meritev procesa in njegovih *produktov*. Izvedemo jih z namenom karakterizacije in razumevanja procesa. /CMM-93/

metoda / method

Zaključena celota pravil in kriterijev, ki zagotovijo natančen in ponovljiv način izvajanja določene *naloge* in doseganja želenega rezultata. /CMM-93/

metodologija / methodology

Zbirka *metod*, *postopkov* in *standardov*, ki sestavljajo zaključeno celoto inženirskih pristopov k razvoju produkta. /CMM-93/

metrika / metric

Značilnost produkta ali procesa, izražena npr. v vrsticah kode, ceni projekta ali delovnih urah. Metrika je lahko objektivna ali subjektivna glede na to ali so podatki rezultat štetja ali subjektivne ocene znotraj podane merske lestvice. Alternativni izraz za metriko je meritev (measurement). /PROCESSUS/

nadzor dokumentacije / document control

Aktivnosti, povezane z izdajanjem novih dokumentov, spremljanjem uporabe in hranjenjem obstoječih dokumentov ter odstranjevanjem neprimernih (zastarelih) verzij dokumentov. /PROCESSUS/

nadzor in izboljšava / control and improvement

Bistvo pristopa AMI je kratkoročno zagotoviti nadzor in dolgoročno izboljšavo razvoja programske opreme. Poudarek pri nadzoru je na boljšem razumevanju in planiranju (*terminskega plana* in kakovosti projekta) in problemov (*upravljanje tveganja*). /AMI/

nadzor kakovosti / quality surveillance

Stalno opazovanje in overjanje stanja *postopkov*, *metod*, *pogojev*, izvajanja *procesov*, *produktov* in storitev ter analiza zapisov glede na dane reference, da se izpolnjujejo specificirane zahteve za kakovost. /SLS ISO 8402 : 1993/

nadzor konfiguracije / configuration control

Element *upravljanja konfiguracije*, ki vsebuje vrednotenje, koordinacijo, odobritev ali zavrnitev ter implementacijo sprememb delov *konfiguracije* po njihovi formalni identifikaciji. /CMM-93/

načrtovanje / design

Je postopek definiranja arhitekture programske opreme, komponent, modulov, vmesnikov, pristopov k *testiranju* in podatkov za programski sistem, ki bo zadovoljil zahteve. Rezultat postopka načrtovanja je načrt. /IEEE 729/

najvišje vodstvo / senior management

Vodstvo na najvišjem nivoju in z največjimi pooblastili v organizaciji. /PROCESSUS/

najvišji vodja / senior manager

Visoka vodstvena *vloga* v *organizaciji*, katere osnovni cilj dolgoročno dobrobit organizacije in ne samo kratkoročni *projekt* in pogodbene določbe in pritiski. Običajno je najvišji vodja za inženirstvo odgovoren za več projektov hkrati. /CMM-93/

naloga, opravilo / task

1. Zaporedje navodil, obravnavano kot osnovna enota dela. /IEEE-610Č
2. Dobro definirana enota dela v *procesu za razvoj programske opreme*, ki zagotavlja upravljanje z vidnimi *mejniki* v statusu *projekta*. Naloge imajo kriterije pripravljenosti (predpogoji) in kriterije dokončanja (zaključni pogoji). /CMM-93/(Glej *aktivnost* za razliko.)

napaka / error

Je vzrok za nastop *hibe* (defect).

ne-tehnične zahteve / nontechnical requirements

Dogovori, pogoji in/ali pogodbene določbe, ki določajo ali vplivajo na upravljanje *projektov za razvoj programske opreme*. /CMM-93/

neposredni vodja za programsko opremo / first-line software management

Vodja, ki ima neposredne vodstvene odgovornosti (vključno z zagotavljanjem tehničnih *usmeritev* in vodenjem kadrovske in finančne funkcij) glede *osebja* in *aktivnosti* znotraj določene organizacijske enote (npr. oddelek ali projektni team). /CMM-93/

neskladnost / nonconformity

Neizpolnitev specificiranih zahtev. /SLS ISO 8402 : 1993/

nosilec naloge / task leader

Vodja tehničnega *osebja* za določeno *nalogo*. Nosi odgovornost za tehnične zadeve in zagotavlja tehnične *usmeritve* za *osebje*, ki je na nalogi zaposleno. /CMM-93/

obveza / commitment

Dogovor (ali pogodba) med dvema ali več posamezniki, *skupinami* ali *organizacijami*, ki je sprejet prostovoljno, ki je jasen in se ga morajo držati vse strani. /CMM-93/, /HUM-89/

obvladovanje kakovosti / quality control

Izvajalne tehnike in *aktivnosti*, ki se uporabljajo za izpolnitev zahtev glede kakovosti. /SLS ISO 8402 : 1993/ V okviru metodologije PROCESSUS se uporablja sinonim *vođenje kakovosti*.

ocenitev procesa razvoja programske opreme / software process assesment

Ocenitvena aktivnost izurjenega *teama* strokovnjakov s področja programske opreme. Obsega določitev stanja procesa, določitev najpomembnejših izhodišč, ki se nanašajo na proces razvoja programske opreme ter na vzpostavitev organizacijske podpore za izboljšanje procesa. /CMM-93/ Splošna aktivnost za vpogled v razvoj programske opreme ali poslovnega okolja in ovrednotenje pomembnih izhodišč. /AMI/

odgovornost za proizvod/storitev / product/service liability

Splošni izraz za obveznost proizvajalca ali drugih, da nadomestijo izgubo zaradi osebne poškodbe, škode na imetju ali drugih oblik škode, ki jo povzroči proizvod ali storitev. /SLS ISO 8402 : 1993/

odklon / deviation

Opazno ali izrazito odstopanje od primerne norme, načrta, standarda, postopka, ali parametra, ki ga preverjamo. /CMM-93/

odobritev / approval

Pisno potrdilo odgovornih o strinjanju z vsebino in nadaljnjih ukrepih. /PROCESSUS/

opis dela / statement of work

Opis vsega dela, potrebnega za dokončanje projekta, ki ga predloži stranka. /CMM-93/

opis procesa / process description

Operativna definicija glavnih komponent procesa. Dokumentacija vsebuje popoln, natančen in preverljiv opis zahtev, načrtovanja, obnašanja in ostalih karakteristik procesa. Prav tako lahko dokumentacija vsebuje postopke za določitev v kolikšni meri so ti kriteriji izpolnjeni. Opisi procesov se nahajajo na nivoju nalog, projekta ali organizacije. /CMM-93/

opis procesa razvoja programske opreme / software process description

Operativna definicija glavnih komponent procesa, ki so identificirani v definiranem procesu projekta ali v standardnem procesu organizacije. Dokumentirana zahteva, načrtovanje, obnašanje ali druge karakteristike procesa na popoln, natančen in preverljiv način. /CMM-93/ (Glej tudi opis procesa).

oprostitev usposabljanja / training waiver

Pisno dovoljenje za oprostitev posameznika od usposabljanja, ki je bilo določeno kot obvezno za specifično vlogo. Oprostitev je lahko odobrena, ker je bilo objektivno ocenjeno, da posameznik že obvlada potrebne veščine za izvedbo vloge. /CMM-93/

optimizacijski nivo / optimizing level

(Glej zrelostni nivo - maturity level.) /CMM-93/

organizacija / organization

Enota v podjetju ali drugi entiteti (npr. vladna služba), znotraj katere se odvija več projektov in ki je upravljana kot celota. Vsi projekti znotraj organizacije imajo skupno vodstvo na najvišjem nivoju in skupno politiko delovanja. /CMM-93/

V modelu ISO nastopa v istem pomenu izraz *dobavitelj* (supplier). /ISO 9000-3/

orodje / tool

Programska oprema, namenjena za podporo pri izvajanju določene aktivnosti. /PROCESSUS/

osebje / staff

Posamezniki, vključno z nosilci nalog, ki so odgovorni za izvršitev dodeljene naloge (kot je npr. razvoj programske opreme, upravljanje konfiguracije programske opreme), vendar niso vodje. /CMM-93/

osebje za programsko inženirstvo / software engineering staff

Tehnično osebje (analitiki, programerji in inženirji), vključno z vodji programskih nalog, ki izvajajo aktivnosti razvoja in vzdrževanja programske opreme, vendar niso v vodstvu (niso menedžerji). /CMM-93/

osnovna verzija / baseline

Specifikacija ali produkt, ki je bil formalno preverjen in določen za podlago nadaljnega razvoja. Izvedba morebitnih sprememb v osnovni verziji je dovoljena samo preko formalnih postopkov za nadzor sprememb. /CMM-93/

osnovna verzija izvedbene zmogljivosti procesa / process performance baseline

Dokumentiran opis dejansko doseženih rezultatov pri sledenju procesa, ki je uporabljen kot primerjalni test za primerjavo dejanske izvedbene zmogljivosti procesa s pričakovano. Funkcionalna osnovna verzija procesa je tipično vzpostavljena na nivoju projekta, čeprav bo začetna funkcionalna osnovna verzija procesa običajno izpeljana iz osnovne verzije zmogljivosti procesa. /CMM-93/ (Glej osnovna verzija zmogljivosti procesa - process capability baseline za razliko.)

osnovna verzija zmogljivosti procesa / process capability baseline

Dokumentiran opis razpona pričakovanih rezultatov, ki bi bili običajno doseženi, če bi specifični proces tekel pod tipično določenimi okoliščinami. Osnovne verzije zmogljivosti procesa se običajno vzpostavljajo na organizacijskem nivoju. /CMM-93/ (Glej osnovna verzija izvedbenih zmogljivosti procesa - process performance baselines za razliko.)

periodični pregled/aktivnost / periodic review/activity

Pregled ali aktivnost, ki se pojavlja v natančno določenih časovnih intervalih. /CMM-93/ (Glej dogodkovno vodeni pregled/aktivnost - event-driven review/activity za razliko.)

plan izboljšanja procesa razvoja programske opreme / software process improvement plan

Plan, izpeljan iz priporočil, dobljenih po ocenitvi procesa. Identificira specifične akcije, ki jih je potrebno izvesti, za izboljšavo procesa, ter plan za izvedbo teh akcij. /CMM-93/ Sinonim: plan akcij.

plan kakovosti / quality plan

Dokument, ki opredeljuje specifične postopke, sredstva in zaporedje ukrepov za kakovost za določen proizvod, storitev, pogodbo ali projekt. /SLS ISO 8402 : 1993/

plan razvoja programske opreme / software development plan

Zbirka planov, ki opisujejo aktivnosti, ki jih je potrebno izvesti v sklopu programskega projekta. Vpliva na upravljanje aktivnosti, ki jih izvaja inženirska skupina programskega projekta. Ni omejen na področje uporabe kateregakoli standarda (kot npr. DOD-STD-2167A in IEEE-STD-1058), ki uporablja podobno terminologijo. /CMM-93/

plan upravljanja tveganja / risk management plan

Zbirka planov z opisom tistih aktivnosti za upravljanje tveganja, ki jih je pri projektu potrebno izvesti. /CMM-93/

plan vzdrževanja / maintenance plan

Plan, v katerem so zajeti postopki, aktivnosti ter odgovorne skupine za izvajanje vzdrževanja. /PROCESSUS/

plani za programsko opremo / software plans

Zbirka planov, formalnih in neformalnih, ki določajo, kako bodo izvedene aktivnosti razvoja in/ali vzdrževanja programske opreme. Primeri planov, ki so lahko vključeni: *plan razvoja programske opreme*, *plan zagotavljanja kakovosti programske opreme*, *plan upravljanja konfiguracije programske opreme*, *plan testiranja programske opreme*, *plan upravljanja tveganja* in *plan izboljšanja procesa*. /CMM-93/

podatki / data

Obravnavamo v smislu zbranih podatkov in izvedenih meritvenih podatkov. /AMI/

podpogodba / subcontract

Pogodba, ki jo organizacija sklene z zunanjim dobaviteljem glede izdelave ali dobave programske oz. strojne opreme, ki bo vključena v končni produkt določenega projekta v organizaciji. /PROCESSUS/ Glej *pogodba* za razliko.

podpogodbenuk / subcontractor

Posameznik, partner, korporacija ali združenje, ki z organizacijo sklene pogodbo da bo oblikovalo, razvilo oz. vzdrževalo enega ali več produktov. /CMM-93/

pogodba / contract

Pravno veljavni dokument, ki ga glede določenega projekta skleneta naročnik projekta in organizacija kot izvajalec. /PROCESSUS/

pogodbena določila in pogoji / contract terms and conditions

Specificirani (postavljeni) pravni, finančni in administrativni vidiki pogodbe. /CMM-93/

pokritost (testiranja) / coverage (test ...)

Je razmerje med vsaj enkrat uporabljenimi strukturnimi elementi in vsemi strukturnimi elementi, ki so iste vrste (npr. število najmanj enkrat klicanih rutin proti številu vseh rutin). /DOG-93/

politika / policy

Vodilni princip, ki ga v večini primerov vzpostavi najvišje vodstvo, in ki je v organizaciji ali projektu privzet kot izhodišče za sprejemanje odločitev. /CMM-93/

politika kakovosti / quality policy

Splošne usmeritve in cilji organizacije glede kakovosti, ki jih formalno določi najvišje vodstvo. /SLS ISO 8402 : 1993/

ponovljivi nivo / repeatable level

(Glej *zrelostni nivo - maturity level*.)

ponudnik / bidder

Posameznik, družba, korporacija ali združenje, ki je predložilo ponudbo in je kandidat za dodelitev pogodbe o načrtovanju, razvoju oz. vzdrževanju enega ali več produktov. /CMM-93/

posebni vzroki za nastanek hibe / special causes of a defect

Vzrok za hibo, ki je specifičen za trenutne okoliščine in ni neločljiv del procesa. Posebni vzroki povzročajo naključno variiranje (šume) v izvedbeni zmogljivosti procesa. /CMM-93/

poslovnik kakovosti / quality manual

Osnovni dokument upravljanja kakovosti v organizaciji. V njem so opisani struktura organizacije ter vsi postopki, ki jih organizacija izvaja z namenom zagotavljanja kakovosti. /PROCESSUS/

postopek / procedure

1. Zapisan opis poteka akcije, ki jo je potrebno izvesti, da bo dana naloga ustrezno opravljena. /IEEE-610/
2. V programih: ime dela programa, ki opravi določeno nalogo. /HUM-89/

predlog ukrepa / action proposal

Dokumentiran predlog za spremembo procesa oz. postavke, ki se na proces nanaša. Uvedene spremembe bodo v prihodnosti proces obvarovale pred hibami, ki bi se sicer pojavile zaradi napačnih aktivnosti za preprečevanje programskih hib. /CMM-93/ (Glej tudi *predlog za izboljšanje procesa razvoja programske opreme - software process improvement proposal*.)

predlog za izboljšanje procesa razvoja programske opreme / software process improvement proposal

Dokumentiran predlog za spremembo procesa ali elementov, ki se na proces nanašajo, na podlagi česar se bo izboljšala izvedbena zmogljivost in zmožnost procesa. /CMM-93/

predpis / regulation

Pravilo, zakon ali ukaz, običajno osnovan od zakonodajnih ali urejevalnih teles s sankcijami v primeru neupoštevanja. /PROCESSUS/

predstavniki končnih uporabnikov / end user representatives

Izbran vzorec končnih uporabnikov, ki zastopa celotno populacijo končnih uporabnikov. /CMM-93/

pregled / review

Formalna, dokumentirana in sistematična preiskava določenega proizvoda, aktivnosti ali dokumenta, pri kateri se preverja izpolnjevanje zahtev, identificira probleme in predlaga rešitve. /PROCESSUS/

pregled (načrtovanja, razvoja) / review (design, development)

Formalna, dokumentirana, vseobsegajoča in sistematična preiskava načrtovanja, razvoja, ki se opravi z namenom, da se ovrednotijo zahteve za načrtovanje, razvoj in sposobnost načrtovanja, razvoja, da izpolni te zahteve ter da se ugotovijo problemi in predlagajo rešitve. /SLS ISO 8402 : 1993/

pregled sistema kakovosti / quality system review

Formalno ovrednotenje stanja in primernosti sistema kakovosti, ki ga opravi najvišje vodstvo glede na politiko kakovosti in nove cilje, postavljene zaradi spremenjenih okoliščin. /SLS ISO 8402 : 1993/

preprečevanje hib / defect prevention

Aktivnosti iskanja *hib* ali potencialnih hib in preprečevanje njihove vgraditve v produkt. /CMM-93/

presoja / audit

Neodvisen pregled *delovnih produktov* ali množice delovnih produktov, pri čemer ocenimo njihovo usklajenost s specifikacijami, standardi, pogodbami ali ostalimi kriteriji. /IEEE-610/

presoja kakovosti / quality audit

Sistematična in neodvisna preiskava, katere namen je ugotoviti, ali ukrepi za *kakovost* in njihovi rezultati ustrezajo načrtovanim ureditvam ter ali se te ureditve izvajajo učinkovito in ali so primerne za doseganje ciljev. /SLS ISO 8402 : 1993/

presoja osnovne verzije programske opreme / software baseline audit

Pregled strukture, vsebine in pripomočkov *knjižnice osnovne verzije programske opreme* z namenom *verifikacije skladnosti* z dokumentacijo, ki to verzijo opisuje. /CMM-93/

prevzem / acceptance

Formalni postopek, pri katerem *naročnik* potrdi ustreznost proizvoda in ga prevzame v uporabo. Naročnik potrdi prevzem s podpisom ustreznega dokumenta (npr. poročilo o prevzemu). /PROCESSUS/

prevzemni kriteriji / acceptance criteria

Kriteriji, ki jim mora zadostiti *sistem* ali komponenta sistema, če naj jo uporabniki, *stranke* ali druge pooblaščenke *entitete* sprejmejo. /IEEE-610/

prevzemno testiranje / acceptance testing

Formalno izvedeno *testiranje* z namenom določitve ali *sistem* zadovoljuje prevzemne kriterije ali ne. Po opravljenem prevzemnem testiranju se uporabnik odloči, ali bo sistem prevzel ali ne. /IEEE-610/, /DOG-93/

pridobitve procesa razvoja programske opreme / software process assets

Zbirka entitet, ki jih vzdržuje organizacija z namenom uporabe pri *projektih*, ki se nanašajo na razvoj, *prireditev*, *vzdrževanje* in izvedbo *procesa*.

Pridobitve običajno vključujejo:

- standardni proces v organizaciji,
- opis življenjskega cikla, ki je odobren za uporabo,
- napotke za prireditev standardnega procesa organizacije,
- bazo podatkov o procesu v organizaciji,
- knjižnico dokumentacije, ki je povezana s procesom.

Med pridobitve so lahko vključene tudi katerekoli entitete, ki jih organizacija spozna za koristne pri definiranju in vzdrževanju procesa.

primerjalni test / benchmark

Standard, po katerem lahko opravimo meritve ali primerjave. /IEEE-610/

primernost za testiranje / testability

1. Stopnja, s katero sistem ali njegova komponenta olajša odobritev testnih kriterijev in izvršitev testov z namenom preverjanja izpolnjevanja kriterijev.
2. Stopnja, do katere so postavljene zahteve v smislu dovoljevanja odobritve testnih kriterijev in izvršitev testov z namenom preverjanja izpolnjevanja kriterijev. /IEEE-610/

pripomoček / facility

Programska ali strojna oprema, ki služi kot pomoč pri izvajanju določene aktivnosti. /PROCESSUS/

prireditev (prikrojitev) / tailor

Sprememba *procesa*, *standarda* ali *postopka* tako, da bolje ustreza procesnim zahtevam ali zahtevam programske opreme. /CMM-93/

prireditev procesa / process tailoring

Izdelava *opisa procesa*. Pri tem izpopolnimo, prilagodimo oz. dokončamo vse podrobnosti *elementov procesa* ali drugih nepopolnih specifikacij procesa. V času prireditve procesa običajno upoštevamo tudi vse specifične poslovne potrebe. /CMM-93/

proces / process

1. Zaporedje korakov z namenom doseganja določenega cilja. Primer: proces razvoja programske opreme. /IEEE-610/
2. Sinonim za "proces razvoja programske opreme", razen takrat, kadar je iz konteksta razviden drugačen pomen. /PROCESSUS/

proces razvoja programske opreme / software process

Množica *aktivnosti*, *metod*, *postopkov* in transformacij, ki jih uporabljamo za razvoj in *vzdrževanje* programske opreme in pripadajočih *produktov* (npr. projektni plani, dokumentacija *načrtovanja*, koda, testni primeri, uporabniški priročniki). /CMM-93/ Zaradi pogoste uporabe pogosto uporabljamo sinonim "proces". /PROCESSUS/

procesna dokumentacija / software process-related documentation

Primeri dokumentov ali njihovih delov, ki bodo uporabljeni v naslednjih projektih. Izdelani so kot prirejeni (prikrojeni) deli standardov *procesa* v *organizaciji*. Primeri se nanašajo na subjekte kot je definirani proces projekta, *standardi*, postopki, *plani razvoja programske opreme*, plani meritev, gradiva za *usposabljanje*. /CMM-93/

procesna metrika / process metric

Je metrika, ki jo lahko uporabimo za meritve značilnosti metod, tehnik in orodij, uporabljenih za preučevanje razvoja, verifikacije in delovanja programske opreme.

procesna skupina / process group

Procesno skupino *programskega inženirstva* sestavljajo strokovnjaki, zadolženi za proces razvoja programske opreme, ki se uporablja v organizaciji. Tipična funkcija procesne skupine vključuje definiranje in dokumentiranje procesa, ustalitev in definiranje *procesnih metrik*, podporo zbiranju in analizi podatkov in svetovanje vodstvu na področjih, ki zahtevajo največjo pozornost. Ob četrletjih skupina običajno prireja vodstvene preglede stanja procesa in pri tem imenuje vodje pregledov. /CMM-93/

produkt (proizvod) / product

(Glej *programski izdelek (program)* - *software product* in *delovni produkt programske opreme* - *software work product.*) /CMM-93/

profil / profile

Občasna primerjava (običajno v grafični obliki) planov ali osnutkov z dejanskim stanjem. /CMM-93/

program meritev organizacije / organization's measurement program

Množica medsebojno odvisnih elementov, ki se nanašajo na potrebe meritev v organizaciji. Zajemajo definiranje razsežnosti meritev v organizaciji, metode in postopke za zbiranje podatkov, meritve, metode in postopke za analizo pridobljenih podatkov ter cilje, ki jih organizacija želi doseči z meritvijo. /CMM-93/

program usposabljanja / training program

Množica medsebojno povezanih elementov, ki se osredotočajo na potrebe usposabljanja. Vključuje plan usposabljanja, gradivo za usposabljanje, razvoj usposabljanja, vodenje usposabljanja, pripomočke za usposabljanje, vrednotenje usposabljanja in vzdrževanje zapisov o usposabljanju. /CMM-93/

programska oprema / software

Skupek programov in pripadajoče dokumentacije (uporabniške in tehnične), ki je potrebna za delovanje, uporabo in vzdrževanje programov. /PROCESSUS/

programska oprema za testiranje / test software

Programska oprema, ki je v pomoč pri izvajanju aktivnosti testiranja. /PROCESSUS/

programski delovni produkt / software work product

Vsak izdelek, ki je nastal kot del definiranja, vzdrževanja ali uporabe procesa za razvoj programske opreme, vključno z opisom procesa, načrti, postopki, računalniškimi programi in pripadajočo dokumentacijo, ki je in ki ni namenjena za predajo stranki ali končnemu uporabniku. /CMM-93/ (Glej programski izdelek za razliko.)

programski izdelek / software product

Katerikoli posamezni element množice ali cela množica računalniških programov, postopkov, pripadajoče dokumentacije in podatkov, namenjenih za predajo stranki ali končnemu uporabniku. /IEEE-610/ (Glej programski delovni produkt - software work product za razliko.)

programski projekt / software project

Analiza, specifikacija, načrtovanje, testiranje in vzdrževanje komponent programske opreme in pripadajoče dokumentacije sistema. Projekt lahko nastopa kot del projekta za izgradnjo hardversko/softverskega sistema. /CMM-93/

programsko inženirstvo / software engineering

Je sistematičen pristop k razvoju, uporabi, vzdrževanju in prenehanju obratovanja programske opreme. /IEEE 729/

prva seja za nalogo / task kick-off meeting

Seja ob začetku določene naloge v okviru projekta. Namen seje je priprava vključenih posameznikov za učinkovito izvajanje aktivnosti v okviru naloge. /CMM-93/

razmnoževanje kopij / replication

Postopek pripravljanja več kopij iste verzije programske opreme, kadar je določeno, da dobi naročnik več kot eno kopijo končnega proizvoda. /PROCESSUS/

razvijalec programske opreme / software developer

Oseba, ki sodeluje v katerikoli fazi razvoja programske opreme. /PROCESSUS/

razvoj / development

Vse aktivnosti, ki jih je treba izvesti, da bi ustvarili neki proizvod. /PROCESSUS/

razvoj procesa / process development

Postopek definiranja in opisovanja procesa. Vsebuje lahko planiranje, arhitekturo, načrtovanje, izvedbo in vrednotenje. /CMM-93/

razvoj programske opreme / software development

Vse aktivnosti, ki jih je potrebno izvesti, da bi ustvarili proizvod programske opreme. /ISO 9000-3/

regresijsko testiranje / regression testing

Je ponovno testiranje programa z namenom, da ugotovimo, ali pri odpravljanju napak nismo vnesli nove napake. /DOG-93/

rezultat faze / output from phase

Vsi proizvodi, podatki, dokumentacija, itd., ki so bili v fazi izdelani oz. dopolnjeni in ki jih ob zaključku določene faze dobimo na izhodu. /PROCESSUS/

SCE / SCE

Akronim za Software Capability Evaluation (vrednotenje zmoglosti programske opreme).

SCM / SCM

Akronim za Software Configuration Management (upravljanje konfiguracije programske opreme).

seja za analizo vzrokov / causal analysis meeting

Seja, posvečena analizi hib, ki so bile odkrite v času izvajanja naloge. Sejo izvedemo po dokončanju določene naloge. /CMM-93/

sistem / system

Zbirka komponent, organiziranih tako, da izvršijo določeno funkcijo ali množico funkcij. /IEEE-610/

sistem kakovosti / quality system

Organizacijska struktura, odgovornosti, postopki, procesi in viri za izvajanje vodenja kakovosti. /SLS ISO 8402 : 1993/

sistem knjižnice za upravljanje konfiguracije / configuration management library system

Orodja in postopki za dostop do vsebine knjižnice osnovnih verzij programske opreme. /CMM-93/

sistemska zahteva / system requirement

Stanje ali zmožnost, ki mora biti dosežena ali vsebovana v sistemu ali posamezni komponenti sistema, da zadovolji stanje ali zmožnost, potrebno za rešitev (uporabnikovega) problema. /IEEE-610/

sistemske zahteve za programsko opremo / system requirements allocated to software

Podmnožica sistemskih zahtev, ki jih morajo izvesti posamezne komponente sistema. Dodeljene zahteve so osnovni vhodi v plan razvoja programske opreme. Analiza zahtev za programsko opremo izpopolni in izboljša dodeljene zahteve ter vrne dokumentirane zahteve za programsko opremo. /CMM-93/

skladnost / consistency

Stopnja enovitosti, standardizacije in odsotnosti protislovij med dokumenti, deli sistema ali komponentami. /IEEE-610/

skupina / group

Več oddelkov, vodij in posameznikov, ki nosijo odgovornost za množico *nalog* ali *aktivnosti*. Skupina lahko variira od posameznika, zaposlenega le del svojega delovnega časa na teh nalogah, do množice posameznikov (lahko tudi iz različnih oddelkov), ki so zaposleni svoj poln delovni čas na teh nalogah. /CMM-93/

skupina za programsko inženirstvo / software engineering group

Množica posameznikov (vodij in tehničnega osebja), ki niso odgovornost za razvoj in *vzdrževanje* programske opreme (to je: analiza zahtev, *načrtovanje*, kodiranje in *testiranje*) za projekt. *Skupine*, ki sicer opravljajo delo, povezano s programsko opremo (kot je npr. skupina za *zagotavljanje kakovosti programske opreme*, procesna skupina za *programsko inženirstvo*), niso vključene v skupino programskih inženirjev. /CMM-93/

skupina za programsko inženirstvo v procesu / software engineering process group

Skupina specialistov, ki pospeši definiranje, *vzdrževanje* in izboljšanje procesa, ki ga uporablja *organizacija*. V *ključnih postopkih* je ta skupina večinoma imenovana "skupina, odgovorna za aktivnosti, povezane s procesom v organizaciji". /CMM-93/

skupina za sistemsko inženirstvo / system engineering group

Skupina posameznikov (vodij in tehničnega osebja) ki nosijo odgovornost za:

- specifikiranje *sistemskih zahtev*,
- dodeljevanje *sistemskih zahtev* (za zadovoljitev s strojno opremo, programsko opremo ali z drugimi komponentami),
- specifikiranje *vmesnikov* med strojno opremo, programsko opremo in drugimi komponentami,
- nadziranje *načrtovanja* in *razvoja* teh komponent ter njihovega ujemanja s specifikacijami. /CMM-93/

skupina za usposabljanje / training group

Množica ljudi (vodij in *osebja*), ki so odgovorni za usklajevanje in urejanje aktivnosti *usposabljanja* znotraj *organizacije*. Ta skupina običajno pripravlja in vodi večino tečajev usposabljanja ter usklajuje ostala sredstva usposabljanja. /CMM-93/

skupina, povezana s programsko opremo / software related group

Množica posameznikov (vodij in tehničnega osebja), ki nudijo *programsko inženirstvo* za podporo, (vendar ne z neposredno odgovornostjo) za razvoj in/ali *vzdrževanje* programske opreme. Primeri disciplin programskega inženirstva: *zagotavljanje kakovosti programske opreme*, *upravljanje konfiguracije* programske opreme. /CMM-93/

skupne lastnosti / common features

Kategorije nadaljnje delitve *ključnih procesnih področij* modela CMM. To so *atributi*, ki pokažejo ali je izvedba in institucionalizacija ključnega procesnega področja učinkovita, ponovljiva in trajna.

Skupne lastnosti modela CMM so naslednje:

- *izvedbena obveza (commitment to perform)* - Vse aktivnosti, ki jih mora izvesti *organizacija*, da zagotovi izvedbo *procesa*. Običajno obsega vzpostavitev organizacijske *politike* in pridobitev podpore vodstvenih struktur.
- *zmožnost izvedbe (ability to perform)* - Predpogoji, ki morajo v organizaciji obstajati, če naj organizacija kompetentno izvede proces. K tem pogojem običajno spadajo viri, organizacijske strukture ter *usposabljanje*.
- *izvedene aktivnosti (activities performed)* - Opis *vlog* in postopkov, ki so potrebni za izvedbo ključnega procesnega področja. Izvedene aktivnosti običajno vključujejo zasnovane planov in procedur, izvajanje dela, sledenje delu in izvajanje korekcij, če se to izkaže za potrebno.
- *meritve in analize (measurements and analysis)* - Opis nujnosti *meritev procesa* in analize rezultatov, dobljenih z meritvami. Običajno vključujejo primere meritev, ki jih lahko uporabimo za določitev statusa in učinkovitosti izvedenih akcij.
- *verificiranje implementacije (verifying implementation)* - Zajema korake, ki jih je potrebno izvesti za preverjanje ali so bile vse aktivnosti izvedene v skladu s postavljenim procesom. Verificiranje običajno obsega pregledne in *presoje*, ki jih izvajata vodstvo in skupina za zagotavljanje kakovosti. /CMM-93/

skupni vzrok (hibe) / common cause (of a defect)

Vzrok *hibe*, ki je sam po sebi del *procesa* ali *sistema*. Skupni vzroki vplivajo na vsak izhod iz procesa in na vsakogar, ki na procesu dela. /CMM-93/ (Glej *posebni vzrok - special cause*, ki določa nasprotni primer)

sledljivost / traceability

1. Stopnja, do katere je lahko vzpostavljena povezava med dvema ali več produkti razvojnega procesa, posebno med produkti, ki so med seboj v odnosu predhodnik-naslednik ali nadrejeni-podrejeni. /IEEE-610/
2. Sposobnost slediti potek, uporabo ali namestitve predmeta ali dejavnosti oziroma podobnih predmetov ali dejavnosti s pomočjo zapisane identifikacije. /SLS ISO 8402 : 1993/

SPA / SPA

Akronim za Software Process Assessment (*ocenitev procesa razvoja programske opreme*). /CMM-93/

specifikacija / specification

1. Natančen in preverljiv opis karakteristik produkta. Procesna specifikacija podrobno definira metode, postopke ali procese, ki bodo uporabljeni pri izvajanju naloge. Specifikacije izdelata tehnično osebje in sicer pogosto kot del pogodbenega dogovora. /HUM-89/
2. Dokument, ki predpisuje zahteve, s katerimi mora biti proizvod ali storitev v skladu. /SLS ISO 8402 : 1993/

SQA / SQA

Akronim za Software Quality Assurance (*zagotavljanje kakovosti programske opreme*). /CMM-93/

standard / standard

Obvezne zahteve uporabljane in uveljavljene za *predpis* discipliniranega in enotnega pristopa k razvoju programske opreme. /CMM-93/, /IEEE-610/

standardni dokument / standard document

Obrazci in predloge, ki so definirani in veljavni v organizaciji in jih osebe izpolnjuje ob izvajanju posamezne aktivnosti. /PROCESSUS/

standardni postopki / standard procedures

Definirani in uporabljeni postopki za izvajanje aktivnosti v organizaciji. Pojem nastopa kot sinonim za *ključni postopki*. /PROCESSUS/

standardni proces razvoja programske opreme / standard software process

(Glej *standardni proces razvoja programske opreme v organizaciji - organization's standard software process*)

standardni proces razvoja programske opreme v organizaciji / organization's standard software process

Operativna definicija osnovnega procesa, ki bo služil kot vodilo za vzpostavitev splošnega procesa za vse projekte v organizaciji. Opisuje osnovne *elemente procesa*, za katere pričakujemo, da jih bo vključil vsak projekt pri *definiranju svojega procesa*. Prav tako opisuje relacije med posameznimi elementi procesa (npr. vmesniki in vrstni red). /CMM-93/

stopnja / stage

1. Razdelitev truda na dele (primernih velikosti za upravljanje), ki pomenijo pomensko in izmerljivo množico medsebojno odvisnih nalog, izvedenih v sklopu projekta. Stopnja je običajno obravnavana kot faza življenjskega cikla programske opreme in se pogosto konča s formalnim pregledom pred prehodom na naslednjo stopnjo. /CMM-93/
2. Kazalec kategorije ali razred lastnosti ali karakteristik, ki se nanašajo na različne skupine potreb za proizvode ali storitve, namanjene za isto funkcionalno uporabo.

stranka / customer

Posameznik ali organizacija, ki je odgovorna za prevzem produkta in pooblaščen za plačilo organizaciji, ki je produkt razvila. /CMM-93/

V modelu ISO nastopa kot sinonim za stranko izraz *kupec (purchaser)*. /ISO 9000-3/

tabela entitet / entity table

Seznam pomembnih lastnosti produkta ali aktivnosti. Seznam lahko uporabimo tudi kot predlogo (izv. template). /AMI/

team (skupina) / team

Množica ljudi, pogosto članov različnih (vendar medsebojno povezanih) skupin, imenovanih za izvedbo dobro definirane funkcije za organizacijo ali projekt. Člani teama lahko v okviru projekta sodelujejo samo delno in imajo druge primarne zadolžitve. /CMM-93/

tehnična izmenjava / technical interchange

Izmenjava medsebojno usklajenih predpisov o tehničnih značilnostih produktov programske opreme in nanje vezane strojne/periferne opreme. /PROCESSUS/

tehnične zahteve / technical requirements

Zahteve, ki opisujejo, kaj mora programska oprema opraviti, in omejitve pri njenem delovanju. Primeri tehničnih zahtev vključujejo funkcionalne, kakovostne zahteve in zahteve vmesnikov. /CMM-93/

tehnologija / technology

Uporaba znanosti in inženirstva za doseganje določenega rezultata. /CMM-93/

telo za vodenje konfiguracije programske opreme / software configuration control board

Skupina, odgovorna za vrednotenje in odobritev ali zavrnitev predlaganih sprememb *elementov konfiguracije* in za zagotavljanje izvedbe odobrenih sprememb. /CMM-93/

terminski plan / schedule

Definirana časovna razporeditev začetka izvajanja in roka dokončanja posameznih nalog ali akcij znotraj projekta oz. ostalih aktivnosti. /CMM-93/

testiranje / testing

Je izvajanje ali vrednotenje programa (ali samo komponente) z namenom, da preverimo, ali program ustreza zahtevam, oziroma, da se pokaže razlika med pričakovanimi in dejanskimi izhodnimi vrednostmi. Program lahko izvajamo z računalnikom ali pa na kakšen drug način. /DOG-93/

testiranje enot / unit testing

Je takšno testiranje modulov, kjer je modul, ki ga testiramo, popolnoma ločen od ostalih. Kliče ga ustrezen krmilni modul, sam pa kliče nadomestne module. /DOG-93/ Tudi: Izolirano testiranje modulov.

testiranje sistema / system testing

Posebna oblika preverjanja združene programske in strojne opreme z namenom, da se ugotovi skladnost delovanja z zahtevami. /DOG-93/

testirno okolje / test environment

Niz orodij in drugih pripomočkov, integriranih v celoto. /DOG-93/

testirno orodje / testing tool

Sredstva, ki omogočajo izvedbo posebnih testirnih postopkov. /DOG-93/

tveganje / risk

Možnost izgube. /CMM-93/

učinkovit proces / effective process

Proces, ki ga lahko karakteriziramo kot izvajan, dokumentiran, uveljavljen, uveden, merjen in sposoben za izboljšave. /CMM-93/

(Glej tudi *dobro definiran proces - well defined process*.)

učinkovitost pregledov / review efficiency

Delež najdenih napak v postopku pregledovanja. Izračunamo jo kot kvocient vseh napak, najdenih v pregledu, in vseh napak, najdenih v pregledu in testiranju v času izvedbe integracijskega testa produkta in sistema. Ne izključuje napak, najdenih pri prevzemnem testiranju in pri poskusni uporabi. /CMM-93/

ugotovitve / findings

Zaključki, pridobljeni ob *ocenitvi*, vrednotenju, *presoji* ali pregledu. Označujejo najpomembnejše vidike, probleme ali možnosti znotraj področja preiskovanja. /CMM-93/

ukrep / action

Akcija, ki jo je potrebno izvesti glede na določen dogodek ali stanje. /PROCESSUS/

upravljan in voden (proces) / managed and controlled (process)

Identifikacija in definiranje tistih delovnih produktov programske opreme, ki ne spadajo med osnovne verzije in zato niso vključeni v upravljanje konfiguracije, vendar jih je potrebno nadzorovati, če naj projekt napreduje na discipliniran način. To pomeni, da je verzija produkta, ki je v določenem trenutku v uporabi (sedaj ali v preteklosti), natančno poznana (t.j. nadzor verzij), prav tako so nadzorovane tudi vse vgrajene spremembe (t.j. nadzor sprememb). /CMM-93/

upravljanje kakovosti programske opreme / software quality management

Proces definiranja ciljev kakovosti za programsko opremo, izdelave planov za doseganje teh ciljev, ter nadziranja in prilagajanja planov, delovnih produktov programske opreme, aktivnosti in kakovostnih ciljev. Namen procesa je zadovoljitev potreb in zahtev strank in končnih uporabnikov. /CMM-93/

upravljanje konfiguracije / configuration management

Disciplina, ki uporablja tehnične in administrativne usmeritve in njihovo stalno uporabo in sicer za:

- identifikacijo in dokumentiranje funkcionalnih in fizičnih karakteristik določenega elementa konfiguracije,
- nadzor sprememb teh karakteristik,
- posnetek in poročilo o procesiranju sprememb in statusu implementacije,
- verificiranje skladnosti s specifikiranimi zahtevami. /IEEE-610/

upravljanje konfiguracije osnovnih verzij / baseline configuration management

Vzpostavitev osnovnih verzij, ki so formalno pregledane in obstaja soglasje v zvezi z njimi in ki dejansko služijo kot osnova za nadaljnji razvoj. Nekateri delovni produkti programske opreme (npr. načrt in koda) morajo imeti osnovne verzije osnovane na vnaprej določenih točkah in za te elemente mora biti izveden natančen nadzor sprememb. Te osnovne verzije zagotavljajo nadzor in stabilnost pri komuniciranju s stranko. /CMM-93/ (Glej tudi upravljanje osnovne verzije - baseline management.)

upravljanje osnovne verzije / baseline management

(Pri upravljanju konfiguracije) Uporaba tehničnih in administrativnih usmeritev za določitev dokumentov in sprememb v tistih dokumentih, ki formalno identificirajo in uvajajo osnovne verzije v določenem času življenjskega cikla elementa konfiguracije. /IEEE-610/

upravljanje razvojne konfiguracije / developmental configuration management

Uporaba tehničnih in administrativnih usmeritev za določitev in nadzor programske opreme in pripadajoče tehnične dokumentacije, ki definira nastajajočo konfiguracijo delovnih produktov med razvojem programske opreme. Upravljanje razvojne konfiguracije je pod neposrednim nadzorom strokovnjakov za razvoj. Elementi upravljanja razvojne konfiguracije niso osnovne verzije, kljub temu, da jih na določeni razvojni točki njihovega razvoja lahko uvrstimo med osnovne verzije. /CMM-93/

upravljanje tveganja / risk management

Pristop k analizi problemov, pri kateri rangiramo tveganje glede na situacijo z uporabo verjetnosti s ciljem čim natančnejšega razumevanja obstoječega tveganja. Upravljanje tveganja vključuje identifikacijo tveganja, analizo, postavitve prioritete in vodenje. /CMM-93/

uskladitveni faktor / contingency factor

Poravnava (povečanje) velikosti, stroškov ali termenskega plana zaradi običajno prenizke ocenitve teh parametrov. Do prenizkih ocenitev pride zaradi nepopolnih specifikacij, neizkušenosti pri ocenjevanju aplikacijskih domen, ipd. /CMM-93/

usmeritev / guideline

Predlagana praksa, metoda ali postopek, ki jo običajno postavi določena avtoriteta. /HUM-89/

usposabljanje / training

Pripraviti nekoga za uporabo posebnih navodil ali postopkov. /CMM-93/

validacija / validation

1. Proces vrednotenja programske opreme ob koncu njenega razvoja z namenom, da se ugotovi skladnost izdelane programske opreme z zahtevami.
2. Vrednotenje programske opreme z namenom, da se ugotovi usklajenost delovanja in zahtev. /DOG-93/

verifikacija / verification

1. Proces, pri katerem preverjamo ali produkt tekoče faze ustreza zahtevam, postavljenim v predhodni fazi.
2. Formalno dokazovanje pravilnosti programov.
3. Sinonim za vse vrste preverjanj. /DOG-93/

vhod za fazo / input for phase

Vsi potrebni proizvodi, podatki, dokumentacija, orodja, standardi, predpisi, itd., ki so potrebni na vhodu določene faze. /PROCESSUS/

viri / resources

Osebe, programska oprema, strojna oprema in denarna sredstva, ki so potrebna za izvajanje projektov v organizaciji. /PROCESSUS/

visoko-nivojski jezik / high-level language

Programski jezik, ki običajno vključuje značilnosti, kot so: gnezdenje izrazov, uporabniško definiranje podatkovnih tipov in prenos parametrov, česar običajno ne najdemo v nižje-nivojskih jezikih. /IEEE 729/

vključen proizvod / included product

Proizvod, za katerega naročnik zahteva, da je vključen v programski proizvod. Vključen proizvod lahko izdelata naročnik ali ga priskrbi od tretje stranke. /PROCESSUS, ISO 9000-3/

vloga (položaj) / role

Zahtevane odgovornosti (kot enota), ki jih lahko prevzame eden ali več posameznikov. /CMM-93/

vodeni nivo / managed level

(Glej zrelostni nivo - maturity level)

vodenje kakovosti / *quality management*

Vidik celotne funkcije vodenja, ki določa in izvaja *politiko kakovosti*. /SLS ISO 8402 : 1993/

vodja / *manager*

Vloga, ki obsega zagotavljanje tehničnih in administrativnih *usmeritev*, ter nadzor posameznikov in njihovega izvajanja *nalog* ali *aktivnosti*, ki so v sklopu pristojnosti vodje. Običajne funkcije vodje zajemajo planiranje, zagotavljanje virov, *organizacijo*, usmerjanje in vodenje dela znotraj področja njegovih pristojnosti. /CMM-93/ Kot sinonim je v uporabi tudi izraz *menedžer* (samo v projektu PROCESSUS).

vodja internega pregleda / *peer review leader*

Usposobljen in kvalificiran posameznik za *načrtovanje*, *organizacijo* in vodenje *internega pregleda*. /CMM-93/

vodja podpogodb / *subcontract manager*

Vodja v osnovni pogodbeni organizaciji, ki nosi neposredno odgovornost za upravljanje in nadzor ene ali več podpogodb. /CMM-93/

vodja programske opreme v projektu / *project software manager*

Položaj s popolno odgovornostjo za vse aktivnosti znotraj *projekta*, ki se nanašajo na programsko opremo. *Vodja projekta* skupno z vodjem za programsko opremo obravnava vsa vprašanja, povezana z obveznostmi in *viri* glede programske opreme. /CMM-93/

vodja projekta / *project manager*

Položaj s popolno poslovno odgovornostjo za celoten *projekt*; posameznik, ki usmerja, nadzira, upravlja in regulira projekt izgradnje sistema programske ali strojno/programske opreme. *Vodja projekta* je neposredno odgovoren *stranki*. /CMM-93/

vodja za programsko opremo / *software manager*

Katerikoli *vodja* (*menedžer*) na projektne nivoju ali nivoju *organizacije*, ki nosi neposredno odgovornost za razvoj oz. *vzdrževanje* programske opreme. /CMM-93/

vprašalnik o zrelosti / *maturity questionnaire*

Množica vprašanj o procesu, ki povzamejo vzorec *ključnih postopkov* iz vsakega procesnega področja iz CMM. Zrelostni vprašalnik služi kot odskočna deska za ocenitev sposobnosti organizacije ali projekta za zanesljivo izvedbo procesa. /CMM-93/

vrednotenje zmoglosti programske opreme / *software capability evaluation*

Ocenenje, ki jo opravi *team* usposobljenih strokovnjakov z namenom identifikacije pogodbenih partnerjev, ki so usposobljeni za izvedbo dela na programski opremi ali za nadzor stanja procesa, uporabljenega na obstoječih dosežkih programske opreme. /CMM-93/

vzdrževanje / *maintainance*

Proces spreminjanja *sistema* ali njegovih komponent po predaji *naročniku*. Spreminjanje zajema popravljanje *napak*, izboljšanje učinkovitosti ali drugih *atributov*, ali prilagoditev na spremenjeno okolje. /IEEE-610/

začetni nivo / *initial level*

(Glej *zrelostni nivo - maturity level*.)

zagotavljanje kakovosti / *quality assurance*

Vsi načrtovani in sistematični ukrepi, ki so potrebni za doseganje ustreznega zaupanja, da bo *produkt*, *proces* ali storitev izpolnila postavljene zahteve glede kakovosti. /SLS ISO 8042 : 1993/

(Glej *zagotavljanje kakovosti programske opreme - software quality assurance*.)

zagotavljanje kakovosti programske opreme / *software quality assurance*

1. Planirani in sistematični vzorec vseh akcij, potrebnih za pridobitev potrebnega zaupanja, da bodo delovni produkti programske opreme skladni s postavljenimi tehničnimi zahtevami.
2. Množica aktivnosti, oblikovanih za vrednotenje procesa, s katerim razvijamo oz. vzdržujemo delovne produkte programske opreme. /CMM-93/

zahtevano usposabljanje / *required training*

Usposabljanje, ki ga določi *organizacija* kot zahtevano za izvajanje določene *vloge* oz. *položaja*. /CMM-93/

zahteve za programsko opremo / *software requirements*

Stanje oz. *zmoglost*, ki jo mora imeti programska oprema, potrebna za rešitev problema ali doseganje določenega cilja. /IEEE-610/

zanesljivost / *reliability*

Sposobnost proizvoda, da opravlja zahtevano funkcijo pod določenimi pogoji v določenem obdobju. /SLS ISO 8402 : 1993/

zanka kakovosti; spirala kakovosti / *quality loop; quality spiral*

Zasnova modela med seboj odvisnih dejavnosti, ki vplivajo na *kakovost proizvoda* ali storitve v različnih fazah, od opredelitve potreb do ocenitve, ali so te potrebe zadovoljene. /SLS ISO 8402 : 1993/

zmoglost procesa / *process capability*

Razpon pričakovanih rezultatov, ki jih lahko dosežemo, če sledimo *procesu*. /CMM-93/ (Glej *izvedbena zmogljivost procesa - process performance* za razliko).

zmoglost procesa razvoja programske opreme / *software process capability*

(Glej *zmoglost procesa - process capability*.)

zmoglostno-zrelostni model / *Capability Maturity Model (CMM)*

Opis stopenj, ki jih mora preiti *organizacija* pri svojem razvoju, ko definira, izvaja, meri, nadzira in izboljšuje svoj *proces razvoja programske opreme*. Model zagotavlja vodilo za izbiro ustreznih strategij izboljšanja procesa tako, da pospeši določanje trenutnih *zmoglosti procesa* ter identifikacijo najbolj ključnih vidikov glede kakovosti programske opreme in izboljšave procesa. /CMM-93/ Model CMM je razvil SEI (Software Engineering Institute); model definira množico ključnih aktivnosti, ki sestavljajo dobro prakso *programskega inženirstva*. /AMI/

zrelost procesa / software process maturity

Stopnja, na kateri je proces eksplicitno definiran, voden, merjen, nadziran in učinkovit. Zrelost predstavlja potencial za rast zmoglosti ter nakazuje obsežnost procesa v organizaciji in konsistenco, s katero je uporabljen pri projektih v organizaciji. /CMM-93/

zrelostni nivo / maturity level

Natančno definirana raven pri zavzemanju za doseganje zrelega procesa. SEI v CMM definira pet zrelostnih nivojev:

- **začetni** - Proces na tem nivoju je karakteriziran kot "ad hoc", pogosto celo kaotičen. Le malo postopkov je definiranih in uspeh je popolnoma odvisen od napora posameznikov.
- **ponovljivi** - Vzpostavljeno je osnovno upravljanje procesa za sledenje stroškom, terminskemu planu, in izvedbeni zmogljivosti. S takšno organiziranostjo procesa je organizacija sposobna doseči dobre rezultate pri izvajanju podobnih aplikacij, kot jih je že razvila.

- **definirani** - Upravljanje procesa in izvajanje inženirskih aktivnosti sta dokumentirana, standardizirana in integrirana v standardni proces organizacije. Vsi projekti uporabljajo odobren standard za razvoj in vzdrževanje programske opreme (oz. prirejeno verzijo) v organizaciji.
- **vodeni** - Zbrani so podrobni podatki o meritvah procesa ter kvaliteti produktov. Softverski produkt in proces sta kvantitativno ovrednotena in nadzirana.
- **optimizacijski** - Zagotovljeno je neprestano izboljšanje procesa s kvantitativno povratno informacijo od procesa in od inovativnih idej in tehnologij. /CMM-93/

življenjski cikel programske opreme / software life cycle

Časovna perioda, ki se prične ob zasnovi programskega produkta in se konča šele takrat, ko produkt ni nikjer več v uporabi. Življenjski cikel programske opreme zajema fazo zasnove, fazo zahtev, fazo načrtovanja, fazo izvedbe, fazo testiranja, fazo instalacije in odjave, fazo uporabe in vzdrževanja, in včasih še fazo umika iz uporabe. /IEEE-610/

Literatura / references:

- /AMI/ AMI Handbook, A quantitative approach to software management, Metric User's Handbook.
- /CMM-93/ CMM Practices, CMU/SEI-93-TR-25, 1993.
- /DOG-93/ Dogša T., Verifikacija in validacija programske opreme, Tehniška fakulteta Maribor, 1993.
- /HUM-89/ Humprey W., Managing the Software Process, Addison Wesley, 1989.
- /IEEE-610/ IEEE STD 610 Standard Glossary of Software Engineering Terminology, The institute of Electrical and Electronics Engineers, New York, USA, 1990.
- /IEEE-729/ IEEE STD 729 Glossary of Software Engineering Terminology, The institute of Electrical and Electronics Engineers, New York, USA, 1983.
- /ISO 9000-3/ Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, ISO 9000-3 : 1991 (E), International Organization for Standardization, Geneva, Switzerland, 1991
- /PROCESSUS/ Metodologija PROCESSUS, Delovna gradiva seminarjev in delavnic, 1994-1996
- /SLS 8402:1993/ SLS ISO 8402: Kakovost - Slovar (identičen z ISO 8402 : 1986 (E)), Urad za standardizacijo in meroslovje pri Ministrstvu za znanost in tehnologijo, Ljubljana, Slovenija, 1993

Literatura, uporabljena za preverjanje prevoda:

- /TURK-87/ Turk I. s sodelavci, Pojmovnik poslovne informatike, Društvo ekonomistov Ljubljana, 1987.
- /RSLO-93/ Računalniški slovarček, angleško slovenski, slovensko-angleški, tretja, razširjena izdaja, Cankarjeva založba, Ljubljana 1993.
- /PRAV-75/ Slovenski pravopis
- /SLOT-89/ Slovar tujk

BLEJSKA KONFERENCA O ELEKTRONSKEM POSLOVANJU

Od 10. do 12. junija 1996 je bila na Bledu deveta mednarodna konferenca **Povečanje učinkovitosti in uspešnosti trgovine z elektronskim poslovanjem (Electronic Commerce for Trade Efficiency and Effectiveness)**. Znak konference, ki si je tokom let pridela ime **Bled Electronic Commerce Conference**, tvorijo štirje medsebojno povezani krogi, ki predstavljajo povezovanje štirih dejavnikov, pomembnih za uresničitev elektronskega poslovanja: podjetij, vladnih organizacij, ponudnikov informacijske tehnologije in univerz. Blejska konferenca je slovenski prispevek k svetovnim prizadevanjem razvijanja elektronskega poslovanja.

Njena usmeritev je skladna s prizadevanji Evropske zveze za pospešitev informatiziranja družbe, katerega pomembna sestavina je ravno elektronsko poslovanje.

V programu konference so bili poleg vrste slovenskih in mednarodnih referentov v plenarnem delu tudi tematsko usmerjeni paneli o elektronskem poslovanju: trgovina in oskrbovalna veriga, izvozno-uvozni proces (izvoznik, špediter, prevoznik, carina), telekomunikacije, zdravstvo - zavarovalstvo, vladne organizacije - podjetja, plačilni promet, elektronsko poslovanje malih podjetij, raziskovanje elektronskega poslovanja. Predstavljene so bile tehnologije ripa, Interneta in elektronskega poslovanja.

Upoštevanje tega, da v podjetjih in vladnih organizacijah držav Evropske Unije uvajajo elektronsko poslovanje, je za slovenska podjetja velikega pomena. V naslednjih letih bodo namreč morala zagotoviti, da bodo usposobljena za elektronsko poslovanje, s čim manj papirja, zamud in stroškov. V mnogih slovenskih podjetjih so že bili prisiljeni uvesti računalniško izmenjavanje podatkov - rip v poslovanje, ker so to zahtevali njihovi kupci. Zato nekatera podjetja že poslujejo elektronsko, nekatera pa se na elektronsko poslovanje pripravljajo. Vrsta akcij pa je sproženih tudi na ravni vladnih organizacij. Temeljna tehnologija elektronskega poslovanja (electronic commerce) je računalniško izmenjavanje podatkov - rip (Electronic Data Interchange - EDI). Rip omogoča, da organizacije namesto papirnatih listin izmenjujejo elektronska sporočila z uporabo računalnikov, telekomunikacij in standardov listin.

S temi prizadevanji je aktivno povezana tudi Univerza v Mariboru. Na Fakulteti za organizacijske vede Univerze v Mariboru že nekaj let izvajajo raziskovalno-razvojni projekt Računalniško izmenjavanje podatkov in medorganizacijski sistemi. V njem sodelujejo trgovska, proizvodna, špedicijska in transportna podjetja, luka, letališče, Agencija za plačilni promet, banke in zavarovalnice, dve občini, Telekom Slovenije, Zveza računovodij, vladne organizacije, naprimer Carinska uprava. Projekt sofinancirajo Ministrstvo za znanost in tehnologijo Republike Slovenije ter sodelujoče organizacije. Slovenski projekt rip in medorganizacijski sistemi je med drugim prispeval k razširjanju spoznanj o obstoju, uporabnosti pa tudi zapletenosti uvajanja tehnologije ripa in elektronskega poslovanja. S tem je Slovenija dosegla določeno stopnjo osveščenosti o ripu in najbrž pridobila na času nekaj let.

Za podjetja v Sloveniji so prizadevanja v omenjenem slovenskem projektu povsem praktično koristna. Na to kažejo nekateri primeri:

- Izmenjavanje elektronskih sporočil namesto papirnih (naročilo, potrditev naročila, račun) nekaterih večjih slovenskih trgovskih organizacijah z dobavitelji.

- Republiška carinska uprava Republike Slovenije je bila v letu 1993 imenovana za koordinatorico modernizacije carinskega poslovanja v enajstih državah Srednje in Vzhodne Evrope. Sedaj za svoje potrebe pripravlja sistem sprejemanja podatkov enotne carinske listine v elektronski obliki.
- Luka Koper spodbuja izvoz in uvoz blaga prek luke brez papirja.
- Zavod za zdravstveno zavarovanje Slovenije že sprejema elektronske račune izvajalcev zdravstvenih storitev namesto papirnih.
- Banke že nekaj let uporabljajo elektronska sporočila v mednarodnem plačilnem prometu v sistemu S.W.I.F.T., omogočajo pa tudi elektronsko poslovanje s sedeža stranke (home banking).
- Agencija za plačilni promet pripravlja uvedbo elektronskih sporočil v plačilnem prometu (nalog za plačilo, obvestilo o plačilu, obvestilo o stanju na računu).
- Ponudniki telekomunikacijskih storitev zagotavljajo vrsto novih storitev.
- Vsakoletna junijska mednarodna konferenca na Bledu (Bled Electronic Commerce Conference) spodbuja razvoj elektronskega poslovanja v Sloveniji in seznanja svetovno strokovno javnost z razmerami v Sloveniji.
- Konzorcij podiplomskih študentov in izobraževalni simpozij, ki ga po blejski konferenci na Otočcu organizirata Univerza v Mariboru in Krka Novo mesto, povezuje slovenske študente s kolegi, ki na univerzah v Evropi, Avstraliji, Aziji in Severni Ameriki s svojimi profesorji proučujejo elektronsko poslovanje.

Lahko pričakujemo, da bodo mnoga slovenska podjetja prisiljena, da se pospešeno elektronsko povežejo s svojimi partnerji doma in v tujini. V Sloveniji je na voljo informacijska tehnologija, ki omogoča takojšen zagon temeljnih oblik elektronskega poslovanja. Potencialni uporabniki so seznanjeni s prednostmi in težavami uvedbe tehnologije elektronskega poslovanja. Strokovnjaki v organizacijah so usposobljeni za njeno začetno uvedbo. Zato je mogoče priporočiti pospešeno razvijanje in uvajanje elektronskega poslovanja v podjetjih in vladnih organizacijah v Sloveniji. V ta namen pa je potrebno organizirati ustrezne akcije, ki naj pripeljejo do praktično delujočih rešitev. Tovrstni ukrepi lahko prispevajo k temu, da bo Slovenija upoštevana kot država, s katero je mogoče poslovati na sodoben način - elektronsko.

J.G.

■

ISD 96 - 5th International Conference on Information Systems Development
Sopot (Gdansk), Poljska, 24. - 26. september 1996

Organizatorja:
University of Gdansk, Departement of Information Systems
Univerza v Mariboru, Fakulteta za organizacijske vede Kranj

Informacije:
 Jože Zupančič, Fakulteta za organizacijske vede, Kranj, Prešernova 11
 e-pošta: joze.zupancic@fov.uni-mb.si, WWW: <http://panda.bg.univ.gda.pl/~isd>

■

PROCESSUS - Uvajanje sistemov kakovosti za področje softverske dejavnosti
Maribor, 17. in 18. oktober 1996

Organizator:
Univerza v Mariboru
Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko, Maribor

Informacije in prijave:
 dr. I. Rozman, dr. J. Györkös,
 Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, 2000 Maribor
 el. pošta: Gyorkos@uni-mb.si URL: <http://lisa.uni-mb.si/processus/sts/>

Revija Uporabna informatika bo brezplačno objavljala v rubriki Koledar prireditev datume strokovnih srečanj, posvetovanj in drugih prireditev s področja informatike. Obvestila naj vsebujejo naslednje podatke: ime srečanja, datum in kraj prireditve, naziv organizatorja, ime in telefonska številka kontaktne osebe. Pošiljajte jih na naslov: Slovensko društvo Informatika, za revijo Uporabna informatika, rubrika: Koledar prireditev, 61000 Ljubljana, Vožarski pot 12. Objavljali bomo vsa obvestila, ki bodo prispela 30 dni pred objavo revije.

Navodila avtorjem

Prispevke pošiljajte v predpisani obliki na naslov Slovensko društvo Informatika, 61000 Ljubljana, Vožarski pot 12, s pripisom za revijo Uporabna informatika.

Če je možno, naj bo članek lektoriran. V uredništvu bomo opravili korekturo in se po presoji posvetovali z avtorjem, da članek tudi lektoriramo.

Prispevek naj bo v obsegu največ avtorska pola (30.000 znakov) za strokovne članke in približno 2 do 3 tiskane strani za druge prispevke. Vsak strokovni članek naj ima na začetku povzetek v slovenskem in v angleškem jeziku. Na koncu dodajte kratek življenjepis.

Pošljite ga na disketi in odtisnjene na papirju. Napisan naj bo v urejevalniku **WORD 6.0** oziroma v **ASCII** formatu. Na disketi označite, kateri urejevalnik ste uporabili, in ime datoteke. Datoteko imenujte s svojim priimkom, n. pr. Novak.doc ali Novak.txt.

Slike, ki ste jih izdelali z grafičnim programom, označite podobno. Na natisnjem izvodu članka naj bo jasno vidno, kam sodi posamezna slika. Lahko priložite tudi originalne predloge, ki jih na hrbtni strani označite s številkami, tako kot v natisnjem besedilu.

Pišite v razmaku vrstic 1, brez posebnih ali poudarjenih črk ali podčrtovanja, za ločilom na koncu stavka napravite samo en prazen prostor, ne uporabljajte zamika pri odstavkih.

Za vsa vprašanja se obračajte na tehnično urednico Katarino Puc, 61000 Ljubljana, Ulica Gubčeve brigade 120 tel. 1271-579, elektronska pošta Katarina.Puc@uni-lj.si

UPORABNA INFORMATIKA

ISSN 1318-1882

Ustanovitelj in izdajatelj:

Slovensko društvo Informatika, 61000 Ljubljana, Vožarski pot 12

Glavni in odgovorni urednik:

Mirko Vintar

Svet revije:

Ciril Baškovič, Andrej Cetinski, Ljubica Djordjevič, Franc Križaj, Ivan Žerko

Uredniški odbor:

Tomaž Banovec, Vladimir Batagelj, Ivan Vezočnik, Jože Gričar, Janez Grad, Andrej Kovačič, Tomaž Mohorič, Katarina Puc, Vladislav Rajkovič, Ivan Rozman, Niko Schlamberger, Mirko Vintar.

Tehnična urednica: Katarina Puc

Oblikovanje: Zarja Vintar, Dušan Weiss

Naslovnica: Zarja Vintar

Tisk: Prograf

Naklada: 700 izvodov

Revija izhaja četrtletno. Cena posamezne številke je 1.200 SIT.

Letna naročnina za podjetja SIT 6.000, za vsak nadaljnji izvod SIT 4.000.

Letna naročnina za posameznika SIT 4.000, za študente SIT 1.200.

