

## Verslo tikslų transformavimas į programų sistemos reikalavimus\*

Albertas ČAPLINSKAS (MII)

el. paštas: [alcapl@ktl.mii.lt](mailto:alcapl@ktl.mii.lt)

**Reziumė.** Straipsnyje pasiūlyta metodika, aprašanti kaip verslo tikslus transformuoti į tą verslą palaikančios programų sistemos reikalavimus. Metodika grindžiama Zachmano metodinio karkaso idėjomis. Tačiau šioje metodikoje Zachmano lentelės eilutės ir stulpeliai yra traktuojami kitaip. Siūlomoje metodikoje kairysis stulpelis yra „Kodėl“ stulpelis. Pirmojoje lentelės eilutėje šiame stulpelyje aprašomi verslo tikslai. Kiti stulpeliai yra taip interpretuojami, kad juose formuluojami reikalavimai būtų palaipsniui transformuojami į detalius, matuojamus programų sistemos reikalavimus. Pavyzdžiui, „Kas“ stulpelis susieja verslo tikslus su vaidmenimis, prisidedančiais prie tų tikslų įgyvendinimą ir kiekvienam vaidmeniui aprašomos verslo užduotys, už kurių vykdymą jis atsako. Iš šios informacijos išvedami vartotojo interfeisų ir sistemos panaudojamumo reikalavimai. Svarbiausias siūlomos metodikos privalumas yra tai, kad ji padeda sumažinti diegiamų informacinių ir ryšio technologijų atotrūkį nuo realių kompiuterizuojamo verslo poreikių.

*Raktiniai žodžiai:* IRT atotrūkis, reikalavimų inžinerija, reikalavimų nuleidimas žemyn.

### Ivadas

Dauguma pagal užsakymus kuriamų programų sistemų skirta veiklos ar verslo procesams palaikyti. Tokių sistemų reikalavimai turi būti tiesiogiai susieti su veiklos ar verslo tobulinimo strategija ir prisidėti prie tos strategijos įgyvendinimo [1]. Laikai, kuomet programų sistemos buvo traktuojamos tik kaip duomenų apdorojimo priemonės, gebančios pateikti tiesiog darbo vietoje reikiamą informaciją reikiamu laiku jau praėjo. Šiandien norima gerokai daugiau. Tuo pačiu keičiasi ir programų kuriančių kolektyvų darbo metodai. Kuriant tradicines programų sistemas, analitikai eidavo pas užsakovus, aiškino ji reikalavimus bei verslo taisykles, visa tai perteikdavo projektuotojams bei programuotojams, o pastarieji kurdavo tuos reikalavimus bei taisykles tenkinančias sistemas. Dirbant pagal tokią schemą, strateginio planavimo, reikalavimų inžinerijos ir programų sistemų inžinerijos veiklos buvo atskirtos. Strateginiu planavimu užsiiminėjo verslo specialistai, organizacijų informacinės sistemos arba, griežčiau kalbant organizacijų informacijos apdorojimo sistemos, apskritai nebuvo projektuojamos, klostėsi savaime, o informatikų uždavinys buvo kompiuterizuoti esamus informacijos apdorojimo procesus. Keisti informacijos apdorojimo procesų logiką jie praktiškai negalėjo ir juo labiau negalėjo daryti kokių nors poveikių sprendimams, priimamiems strateginio planavimo metu. Mažai kas pakito ir tada, kada

---

\*Darbas pagal sutartį NR. T-96/07 „Heterogeninių komponentų integravimas šiuolaikinėse informacinėse sistemose“ yra remiamas Lietuvos valstybinio mokslo ir studijų fondo.

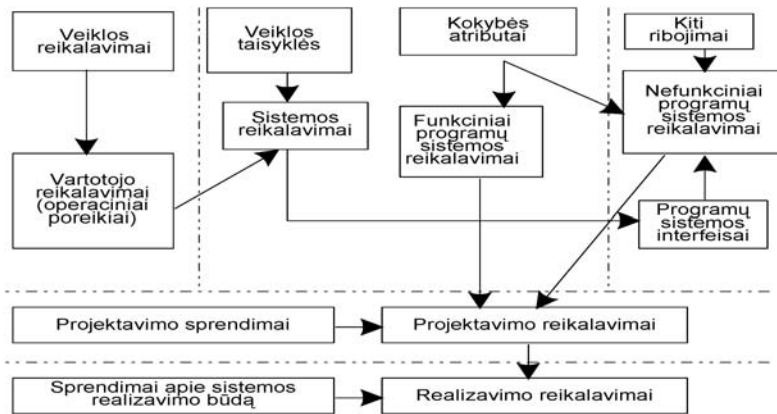
organizacijos savo funkcinių darbo stilių pradėjo keisti procesiniu bei pradėjo diegti vadinamąjį vieno langelio principą, kitaip tariant, pradėjo projektuoti savo informacines sistemas. Programų sistemų reikalavimus pradėta formuluoti remiantis ne vien subjektyvia užsakovų nuomone, bet ir esamų veiklos ar verslo procesų objektyvios analizės rezultatais. Jie tapo objektyvesni, Tačiau strateginio planavimo, informacijos apdorojimo procesų projektavimo ir programų sistemų kūrimo veiklos toliau liko atskirtos vienos nuo kitų. Esant tokiai padėčiai programų sistemos neišvengiamai atitrūkdavo nuo realių verslo tikslų, verslo kompiuterizacija duodavo daug mažesnę efektą, negu buvo tikėtasi. Dėl to buvo pereita prie integruotų organizacijos sistemų inžinerijos, t.y. lygiagretaus verslo procesų, informacijos apdorojimo procesų ir programinės įrangos projektavimo, supinant tuos procesus tarpusavyje ir siekiant sukurti tokią organizacijoje vykstančių procesų visumą, kuri būtų maksimaliai efektyvi jos siekiamų tikslų požiūriu.

### **Reikalavimų lygmenys ir reikalavimų nuleidimas žemyn**

Norint tiesiogiai susieti programų sistemos reikalavimus su veiklos ar verslo tobulinimo strategija, iškyla klausimas, kokius tarpinius reikalavimų lygmenis reikia nagrinėti ir kaip iš vieno lygmens reikalavimų gauti kito, žemesnio lygmens reikalavimus, galimai išvengiant jų atotrūkio nuo aukštesnio lygmens reikalavimų. Kitaip tariant, iškyla reikalavimų nuleidimo žemyn metodikos klausimai.

Reikalavimų nuleidimo žemyn metodikos esmę dar 1990 m. savo darbuose aptarė žymus amerikiečių reikalavimų inžinerijos specialistas Merlin Dorfman [2]. Tai buvo vienas iš pirmųjų darbų šiuo klausimu. Ten išdėstytos idėjos gerai žinomos, įėjo į universitetuose dėstomus reikalavimų inžinerijos kursus. Tačiau Dorfman pasiūlyta metodika pritaikyta tradicinėms programų sistemoms kurti. Ji skirta nuleisti žemyn programų sistemos reikalavimus, t.y. transformuoti juos į projektavimo ir realizavimo reikalavimus. Metodika nenagrinėja nei reikalavimų lygmenų, nei to, kaip iš verslo strateginio lygmens reikalavimų gauti informacijos apdorojimo reikalavimus, o po to iš pastarųjų – programų sistemos reikalavimus.

Karl Wiegars pasiūlė [4] integruotose organizacijų sistemose išskirti tokius reikalavimų lygmenis (1 pav.): veiklos arba verslo lygmens reikalavimus, aprašančius verslo strateginius tikslus; veiklos arba verslo transakcijas vykdančių darbuotojų operacinius poreikius; informacinės sistemos reikalavimus; programų sistemos funkcinius reikalavimus; funkcinių reikalavimų įgyvendinimą ribojančius programų sistemos nefunkcinius reikalavimus; programų sistemos komponentų projektavimo reikalavimus ir programų sistemos komponentų realizavimo (programavimo technologijos) reikalavimus. Karl Wiegars vienas iš pirmųjų parodė, kad vadinamosios veiklos arba verslo taisyklės neaprašo verslo strateginių tikslų ir dėl to pradėti formuluoti reikalavimus nuo jų negalima. Šiomis taisyklėmis reikia pradėti domėtis tik formuluojant trečiojo lygmens reikalavimus, informacinės sistemos reikalavimus, ir galbūt jas keisti, atsižvelgiant į aukštesniųjų lygmenų reikalavimus.



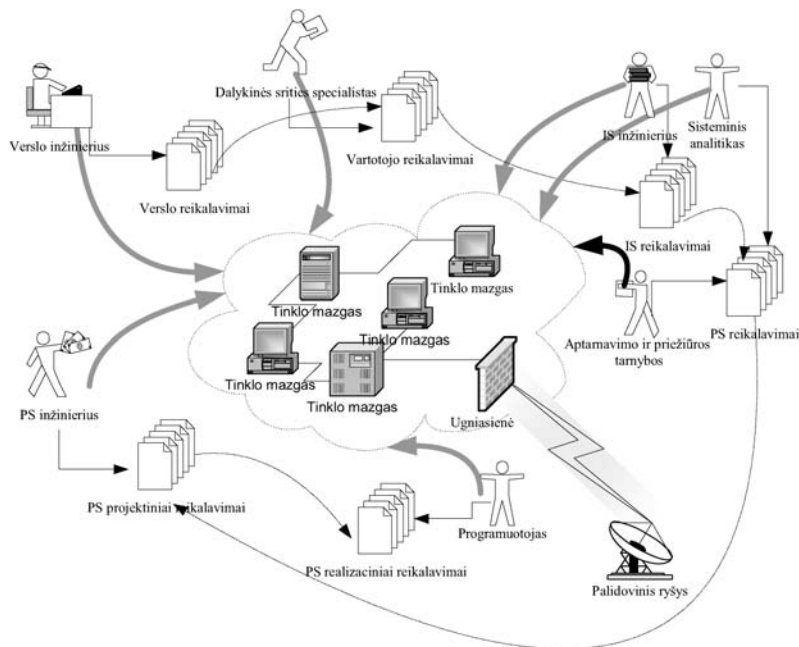
1 pav. Reikalavimų lygmenys.

### Zachmano metodinis karkasas ir jo panaudojimas verslo tikslams į programų sistemos reikalavimus transformuoti

Gerokai anksčiau negu Karl Wieggers, dar 1987 m., organizacijos integruotos sistemos architektūros lygmenis aptarė John Zachman [5]. Jis pasiūlė vadinamąjį metodinį karkasą, kuris išskiria šešis architektūros lygmenis, atspindinčius skirtingų projekto dalyvių (organizacijos vadovybės, operatyvinio lygmens darbuotojų, informacinių sistemų inžinierių, programų sistemų inžinierių, programuotojų, programų sistemos aptarnavimo bei priežiūros tarnybų) turimas kuriamos sistemos vizijas. Aišku, kad Wieggers pasiūlytus reikalavimų lygmenis ir Zachman pasiūlytus architektūros lygmenis nesunkiai galima susieti (2 pav.) ir šitaip gauti organizacijos integruotos sistemos reikalavimų formulavimo koncepcinį modelį, aprašantį kaip iš vieno lygmens reikalavimų reikia gauti kito lygmens reikalavimus ir kas tai turi daryti. Tačiau jokio atsakymo į klausimą, kaip susieti programų sistemų reikalavimus su verslo strategijomis Zachman metodinis karkasas neduoda. Zachman puikiai suprato šio klausimo svarbą, bet sąmoningai sutelkė dėmesį į sistemos architektūros projektavimo klausimus. Jis rašė:

*„Nors informacinių sistemų architektūra yra susijusi su strategija, tiksliau, ir su informacijos apdorojimo strategija ir su verslo strategija, šiame straipsnyje sąmoningai ribojamasi architektūros klausimais, jis nėra skirtas strateginio planavimo metodikai nagrinėti. Verslo strategijų kūrimas ir jų siejimas su informacinių sistemų strategijomis, kurios galų gale pasireiškia architektūriniais sprendimais, yra svarbus nagrinėjimų objektas, bet tai visiškai skirtingas dalykas nuo to, kas yra nagrinėjama šiame darbe – informacinių sistemų architektūros projektuoti skirto metodinio karkaso“ [5].*

Be to Zachman, remdamasis tuo metu vyraujančia nuomone, kad informacinės sistemos pagrindą sudaro duomenų bazės, į pirmą planą iškėlė duomenų struktūrų projektavimo klausimus. Jo metodinio karkaso kiekviename lygmenyje yra išskiriami šeši projektavimo aspektai: *ką* (informacija), *kaip* (informacijos apdorojimo proce-



2 pav. Reikalavimų formulavimas kuriant organizacijos integruotą sistemą.

sai), *kur* (informacijos srautai), *kas* (agentai), *kada* (įvykiai), *kodėl* (loginis pagrindimas). Tai reiškia, kad verslo tikslai nukeliami į paskutinę vietą ir naudojami kitiems architektūriniam sprendimams pagrįsti. Formuluojuojant kuriamos sistemos reikalavimus, mūsų nuomone, reikėtų elgtis atvirkščiai, t.y. reikėtų pradėti nuo verslo tikslų ir iš jų gauti visus kitus reikalavimus.

Organizacijos integruotos sistemos gyvavimo ciklo modelį papildyti strateginio planavimo stadija ir šia stadija pradėti visą ciklą vienas iš pirmųjų pasiūlė David Hay [3]. Jis taip pat pabandė pritaikyti Zachman metodinį karkasą reikalavimams formuluoti. Tačiau Hay nepasiūlė kaip nuleisti reikalavimus žemyn, t.y. nepasiūlė kokius reikalavimus iš kokių konkrečių aukštesnio lygmens reikalavimų reikia gauti ir kaip leidžiantis žemyn padengti visą programų sistemos funkcinių ir nefuncinių reikalavimų spektrą, neatitrūkstant nuo verslo strateginių tikslų. Be to jis paliko Zachman pasiūlytą aspektų išdėstymo tvarką, t.y. taip pat siūlo pradėti formuluoti reikalavimus sistemos apdorojamų duomenų reikalavimais, o ne gauti tuos reikalavimus iš verslo tikslų.

Šiame straipsnyje pasiūlyta metodika, kaip verslo tikslus nuosekliai transformuoti į programų sistemos reikalavimus. Už pagrindą imta Zachman metodinio karkaso idėja, tačiau aspektai išdėstyti visai kita tvarka. Pradedama *kodėl* aspektu, t.y. siūloma visų pirma apgalvoti, kaip norima tobulinti esamą verslą, suformuluoti verslo viziją ir tos vizijos įgyvendinimo strategiją. Taip daroma todėl, kad tikroji programų sistemos vertė priklauso nuo to, kiek ši sistema prisideda prie verslo tobulinimo strategi-

	Kodėl? (motyvacija)	Kaip? (veiklos)	Ką? (apdorojami objektai)	Kas? (funkciniai vienetai)	Kur? (vieta)	Kada? (laikas)
<b>Verslo reikalavimai</b> ( <i>verslo inžinieriaus požiūris</i> )	Misija, verslo sėkmės matavimų rezultatai, problemos, gresmės, neišnaudotos galimybės, rinkos strategija, verslo vykdymo strategija, organizacijos veiklos strategijos, verslo mūštos ir verslo taisyklės, vizija	Viziją įgyvendinant iš verslo tikslų medžiagos ir jo ribojimai.	Verslo tikslams pasiekti reikalingų verslo objektų reikalavimai	Verslo tikslų susiejimo su vykdytojais reikalavimai, vykdytojų įgaliojimų reikalavimai	Verslo tikslų susiejimo su darbo vietomis reikalavimai	Verslo našumo reikalavimai
<b>Vartotojo reikalavimai</b> ( <i>dalykinės sritys specialisto požiūris</i> )	Patobulintų verslo procesų reikalavimai	Informaciniai, skaičiavimo, komunikavimo ir kiti operaciniai poreikiai bei jų ribojimai.	Koncepciniai verslo objektus modeliuojančių informacinių objektų reikalavimai, įskaitant verslo duomenų apsaugos reikalavimus	Kvalifaciniai vykdytojų reikalavimai, preliminariniai panaudojamo reikalavimai	Verslo sistemos darbo vietų reikalavimai	Verslo užduočių vykdymo našumo reikalavimai
<b>IS reikalavimai</b> ( <i>IS inžinieriaus požiūris</i> )	IS vizija, IS galimybių medis, Organizacijos darbo su informacija taisyklės	IS architektūros reikalavimai, IS komponentų funkciniai, saugos ir patikimumo reikalavimai	Informacijos saugyklų ir jose saugojamų informacinių objektų bei jų apsaugos reikalavimai	IS interfeisų ir panaudojamo reikalavimai	IS darbo vietų reikalavimai (įskaitant TI ir PI)	Informacijos apdoravimo užduočių vykdymo našumo reikalavimai
<b>PS reikalavimai</b> ( <i>sisteminio analitiko požiūris</i> )	Ekonominiai, politiniai ir teisiniai ribojimai, PS kokybės vertinimo kriterijai	PS funkciniai, saugos, patikimumo, diegimo, aptarnavimo ir priežiūros reikalavimai	PS apdorojamų duomenų, įskaitant jų apsaugą, reikalavimai	Vartotojo interfeisų ir PS panaudojamo reikalavimai	PS veikti reikalingos TI ir sisteminės PI reikalavimai, TI išdėstymo reikalavimai	PS našumo reikalavimai
<b>Projektiniai PS reikalavimai</b> ( <i>PS inžinieriaus požiūris</i> )	Eskizinio lygmens PS architektūros reikalavimai	PS komponentų funkciniai, saugos, robustiškumo, patikimumo, diegimo, aptarnavimo ir priežiūros reikalavimai	Duomenų bazių ir kitų duomenų saugyklų loginio lygmens reikalavimai	PS komponentų interfeisų reikalavimai	PS komponentų išdėstymo kompiuterių tinkle reikalavimai	PS komponentų našumo reikalavimai
<b>Realizaciniai PS reikalavimai</b> ( <i>programuotojo požiūris</i> )	Detaliosios PS architektūros reikalavimai	PS komponentų realizavimo ir testavimo reikalavimai	Duomenų bazių ir kitų duomenų saugyklų fizinio lygmens reikalavimai	Komponentų vidinių dalių (procedūrų, klasių ir kt.) interfeisų reikalavimai	Skirtinguose tinklo mazguose veikiančių komponentų sąveikos reikalavimai	Algoritmų efektyvumo reikalavimai

3 pav. Metodinis karkasas, pritaikytas verslo tikslams transformuoti į programų sistemos reikalavimus.

jos įgyvendinimo. Neturint gerai apgalvotos verslo plėtros ir tobulinimo strategijos, suformuluoti pagrįstus programų sistemos reikalavimus praktiškai yra neįmanoma. Suformulavus strategiją, siūloma pereiti prie *kaip* aspekto ir suformuluoti strategijos įgyvendinimo reikalavimus, išreiškiant tą strategiją atitinkamu tikslų medžiu. Tada pradedami formuluoti *ką*, *kas*, *kur*, *kada* aspektų reikalavimai (3 pav.). Siūlomoje metodikoje tie aspektai traktuojami kitaip kaip Zachman karkase. Jau viršutiniame lygmenyje visų aspektų reikalavimai taip formuluojami, kad žemutiniame lygmenyje jie virstų atitinkamais programų sistemos reikalavimais ir padengtų visą tų reikalavimų spektrą. Pavyzdžiui, formuluojant *kas* aspekto reikalavimus verslo lygmenyje, verslo tikslai yra susiejami su vykdytojais ir aprašoma, *ką* tie vykdytojai turi teisę daryti.

Nuleidžiant *kas* reikalavimus žemyn galų gale yra gaunami vartotojų interfeisų ir programų sistemos panaudojamumo reikalavimai.

Pagal siūlomą metodiką reikalavimų nuleidimas vyksta dviem kryptimis – iš kairės į dešinę (pagal aspektus) ir iš viršaus žemyn (pagal lygmenis). Taip sudaromos galimybės, viena vertus, rinkti papildomą reikalavimams patikslinti reikalingą informaciją ir, kita vertus, nuolat analizuojant tiek skirtingų lygmenų tiek ir skirtingų aspektų reikalavimų darną, tikrinti surinktą informaciją ir šitaip išvengti programų sistemos reikalavimų atotrūkio nuo realių verslo poreikių.

### Išvados

Straipsnio apimtis neleidžia aptarti siūlomą metodiką išsamiau. Jos esmė matosi iš 3 pav. pateiktos lentelės. Lentelės langeliuose nurodyta, kokius reikalavimus reikia suformuluoti, konkrečiame lygmenyje nagrinėjant konkretų sistemos aspektą. Pagrindinis šios metodikos privalumas yra galimybė nuosekliai pereiti nuo verslo tikslų prie detalių, matuojamų programų sistemos reikalavimų. Judama dviem kryptim: iš kairės į dešinę ir iš viršaus žemyn. Judant iš kairės į dešinę nuo tikslų pereinama prie atitinkamo lygmens funkcinių reikalavimų, formuluojamų „Kaip“ stulpelyje, o po to – prie nefuncinių reikalavimų bei kitų ribojimų, formuluojamų kituose lentelės stulpeliuose. Judant iš viršaus žemyn, kiekviename stulpelyje reikalavimai nuleidžiami iš aukštesnio lygmens ir panaudojami patikrinti, ar einant iš kairės į dešinę nebuvo prarastas ryšys su aukštesniojo lygmens reikalavimais.

### Literatūra

1. S.J. Bleistein, A. Aurum, K. Cox, P.K. Ray, Strategy-oriented alignment in requirements engineering: linking business strategy to requirements of e-business systems using the SOARE approach, *Journal of Research and Practice in Information Technology*, **36**(4), 259–276 (2004).
2. M. Dorfman, System and software requirements engineering, in: R.H. Thayer, M. Dorfman (Eds.), *System and Software Requirements Engineering*, IEEE CS Press, Los Alamitos, Calif. (1990), pp. 4–16.
3. D.C. Hay, *Requirements Analysis, From Business Views to Architecture*, Pearson Education (2003).
4. K.E. Wiegers, *Software Requirements*, 2nd edition, Microsoft Press (2003).
5. J. Zachman, A framework for information systems architecture, *IBM Systems Journal*, **26**(3), 276–292 (1987).

### SUMMARY

#### *A. Čaplinskas. A methodology to transform business goals and objectives into software requirements*

The paper proposes a methodology how to transform business goals and objectives into supporting software requirements. The proposed approach is based on ideas beyond the Zachman's framework. However both lines and columns of the Zachman's framework are interpreted in quite different way. The left column of the proposed framework is Why column. At the top level it describes business goals and objectives. Other columns are interpreted in the way allowing formulating requirements that step-by-step can be flowdown into detailed and measurable software requirements. For example, the Who column associates business goals and subgoals with the particular roles (positions) that in some way contribute to their implementation and for each role describes performing business tasks. User interface requirements and usability requirements are derived later from this information. The main advantage of the proposed methodology is the possibility to ensure better strategic alignment of software requirements.

*Keywords:* strategic alignment, requirements engineering, requirements flowdown.