



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

FIZIKOS KATEDRA

Asta Aleknaite

**ŽINIŲ BAZĖS INTEGRAVIMAS Į PROJEKTŲ VALDYMO SISTEMAS
KNOWLEDGE BASE INTEGRATION INTO PROJECT MANAGEMENT
SYSTEMS**

Baigiamasis magistro darbas

Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111

Fizinių ir techninių tyrimų kompiuterizacijos specializacija

Informatikos studijų kryptis

Vilnius, 2011

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
FIZIKOS KATEDRA

TVIRTINU

Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Asta Aleknaitė

ŽINIŲ BAZĖS INTEGRAVIMAS Į PROJEKTŲ VALDYMO SISTEMAS
KNOWLEDGE BASE INTEGRATION INTO PROJECT MANAGEMENT
SYSTEMS

Baigiamasis magistro darbas

Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111

Fizinių ir techninių tyrimų kompiuterizacijos specializacija

Informatikos studijų kryptis

Vadovas

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Konsultantas

(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2011

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
FIZIKOS KATEDRA

Informatikos studijų kryptis

Inžinerinės informatikos studijų studijų programa, valstybinis kodas 62409P111

Fizinių bei techninių tyrimų kompiuterizavimo specializacija

TVIRTINU

Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

BAIGIAMOJO MAGISTRO DARBO

UŽDUOTIS

2009-11-11Nr. 1

Vilnius

Studentei **Astai Aleknaitė**

Baigiamojo darbo tema:

Žinių bazės integravimas į projektų valdymo sistemas

patvirtinta 2009m. lapkričio 11 d. dekanų potvarkiu Nr.10-502

Baigiamojo darbo užbaigimo terminas 2011m. birželio 1 d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Išanalizuoti projektų valdymo procesus.
2. Susipažinti su egzistuojančiomis projektų valdymo sistemomis.
3. Išsiaiškinti žinių ir žinių vadybos sąvokas ir svarbą įmonės lygyje.
4. Susipažinti su žinių bazėmis, jų sintakse ir struktūra.
5. Išsiaiškinti projektų valdymo sistemos su integruota žinių baze poreikį.
6. Pasirinkti projektų valdymo sistemą ir integruoti žinių bazę.

Baigiamojo darbo rengimo konsultantai:

.....
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vadovas

.....
(Parašas)

.....
(Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau

.....
(Parašas)

Asta Aleknaitė

(Vardas Pavardė)

2009-11-11

(Data)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
FIZIKOS KATEDRA

Fizinių mokslų sritis

Informatikos mokslo kryptis

Informatikos studijų kryptis

Inžinerinės informatikos studijų studijų programa, valstybinis kodas 62409P111

Fizinių ir techninių tyrimų kompiuterizavimo specializacija

PAŽYMA
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ
.....Nr.
Vilnius

Studentas (ė) Asta Aleknaitė

Studento (ės) studijų svertinis įvertinimų vidurkis 9,13 balo.

Baigiamojo darbo tema: Žinių bazės integravimas į projektų valdymo sistemas.

Baigiamasis darbas peržiūrėtas ir studentui (ei) Astai Aleknaitei leidžiama ginti šį baigiamąjį darbą magistro laipsnio suteikimo komisijoje.

Katedros vedėjas

.....
(Parašas)

.....
(Moksl. laipsnis, vardas, pavardė)

VADOVO ATSLIEPIMAS
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vadovas

.....
(Parašas)

.....
(Moksl. laipsnis, vardas, pavardė)

(Moksl. laipsnis, vards, pavardė)

RECENZIJA APIE BAIGIAMĄJĮ MAGISTRO DARBĄ

Vilnius

RECENZIJÄ

[illegible]

.....
(Parašas)

(Moksl. laipsnis, vards, pavardē)

Vilniaus Gedimino technikos
universiteto egzaminų sesijų ir
baigiamųjų darbų rengimo bei gynimo
organizavimo tvarkos aprašo 2010-
2011 m. m.
6 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Asta Aleknaitė, 20053032

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Inžinerinė informatika, FTKfm-09

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA**

2011 m. birželio 1 d.
(Data)

Patvirtinu, kad mano baigiamasis darbas (projektas) tema Žinių bazės integravimas į
projektų valdymo sistemas
patvirtintas 2009 m. lapkričio mėn. 11 d. dekanu potvarkiu Nr. 10-502, yra
savarankiškai parašytas. Šiame darbe (projekte) pateikta medžiaga
nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų
šaltinių citatos pažymėtos literatūros nuorodose.
Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą
(projektą), mane konsultavo
mokslininkai ir specialistai:

Mano darbo (projekto) vadovas prof. dr. P. Miškinis

Kitų asmenų indėlio į parengtą baigiamąjį darbą (projektą)
nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam
nesu mokėjęs(-usi).

(Parašas)

(Vardas ir pavardė)

Vilniaus Gedimino technikos universitetas

Fundamentinių mokslų fakultetas

Fizikos katedra

ISBN ISSN

Egz. sk.

Data-....-....

Informatikos studijų programos baigiamasis magistro darbas

Pavadinimas **Žinių bazės integravimas į projektų valdymo sistemas**

Autorius **Asta Aleknaitė**

Vadovas prof. dr. **Paulius Miškinis**

Kalba

☒ **X**

lietuvių

☐

užsienio

Anotacija

Baigiamajame magistro darbe nagrinėjama žinių bei žinių vadybos įtaka organizacijos tikslų siekimui. Išanalizuoti pagrindiniai žinių, žiniomis pagrįstų sistemų principai bei žinių išgavimo ir vaizdavimo metodai. Aptariami projektų valdymo procesai ir projektų valdymo sistemos. Pateikiama žinių bazės projektavimo bei integravimo metodika bei veikianti projektų valdymo sistema su integruota žinių baze. Išnagrinėjus teorinius žinių, žinių bazių ir žiniomis paremtų sistemų aspektus bei pristačius projekto valdymo sistemą su integruota žinių baze veikiantį modelį, pateiktos baigiamojo darbo išvados ir siūlymai.

Darbą sudaro 8 dalys: įvadas, žinių valdymas, žinių bazės integravimas, projektų valdymas, žinių bazių panaudojimas įmonėse, praktinė dalis, išvados ir pasiūlymai, literatūros sąrašas.

Darbo apimtis – 65 p. teksto be priedų, 29 paveikslai, 1 lentelė, 23 bibliografinių šaltinių.

Prasminiai žodžiai: žinios, žiniomis pagrįsta sistema, žinių bazė, žinių vaizdavimas.

Vilnius Gediminas Technical University

Fundamental sciences faculty

Physics department

ISBN ISSN

Copies No.

Date-....-....

Informatics study programme bachelor (master) thesis.

Title: **Knowledge Base Integration Into Project Management Systems.**

Author **Asta Aleknaitė** Academic supervisor **prof. dr. P. Miškinis**

Thesis language

☒ Lithuanian

☐ Foreign (English)

Annotation

The thesis examined the knowledge and knowledge management influence for organization objectives.

Analytical part of work examined the main knowledge of the principles of knowledge-based system and knowledge extraction and visualization methods. An examination of theoretical knowledge, knowledge bases and knowledge-based systems and delivery aspects of the knowledge base of design and integration methodology, the conclusions and recommendations represented.

The practical part of this work introduces a project management system with integrated knowledge base.

The work consists of eight parts: introduction, knowledge management, knowledge base integration, project management, knowledge base in organization, practical part, summary and suggestions, references.

Thesis consist of: 65 p. text without appendixes, 29 pictures, 1 table, 23 bibliographical entries.

Keywords: knowledge, knowledge base, knowledge based system, knowledge representation.

TURINYS

Paveikslų sąrašas	12
Lentelių sąrašas.....	13
Santrumpos	14
Įvadas.....	15
1. Žinių valdymas	17
1.1. Žinių vadyba	17
1.2. Žinios	18
1.3. Žiniomis pagrįsta sistema	20
1.4. Žiniomis pagrįstos sistemos sandara	22
1.5. Žinių bazė	23
1.6. Žinių vaizdavimas	24
1.7. Žinių bazių sandara.....	29
1.8. Žinių bazių tipai.....	30
1.9. Žinių bazės projektavimas	32
2. Žinių bazės integravimas	37
3. Projektų valdymas	41
3.1. IT projektų valdymas.....	41
3.2. Projektų valdymo sistemos.....	45
3.2.1. Redmine.....	46
3.2.2. Opengoo	47
3.2.3. Factory Nova	48
3.2.4. MS Project	49
4. Žinių bazių pritaikymas įmonėse.....	51
4.1. „Kuoni“ kelionių agentūra.....	51
4.2. „Siemens AG“	52
5. Praktinė dalis	54
5.1. Teamwork.....	54
5.2. Žinių bazės integravimas į Teamwork.....	54
5.3. Naujas užduočių sprendimo procesas.....	55
5.4. Pakeitimai Teamwork duomenų bazėje.....	57

5.5.	Projekto biudžeto vertinimas	58
5.6.	Projekto laiko vertinimas.....	60
5.7.	Darbuotojų rekomendavimas.....	61
5.8.	Žinių bazės panaudojimas užduočių sprendimui.....	62
5.9.	Paieška žinių bazėje.....	65
5.10.	Populiariausi problemų sprendimai	66
6.	Išvados ir pasiūlymai	67
	Literatūros sąrašas	68
	Priedai	70

PAVEIKSLŲ SĄRAŠAS

1 pav. Žinių sąryšis su duomenimis, informacija ir patirtimi ar kompetencija (Akerkar <i>et al.</i> 2009)	19
2 pav. Žinių išgavimo modelis (Akerkar <i>et al.</i> 2009).....	19
3 pav. Žiniomis pagrįstos sistemos architektūra (Akerkar <i>et al.</i> 2009).....	22
4 pav. Apibrėžiamojo tinklo pavyzdys	24
5 pav. Tvirtinančiojo tinklo pavyzdys (Akerkar <i>et al.</i> 2009)	25
6 pav. Išvadų tinklas (Akerkar <i>et al.</i> 2009)	25
7 pav. Freimo pavyzdys.....	27
8 pav. Žinių bazė (Akerkar <i>et al.</i> 2009).....	29
9 pav. Žinių bazė (Tzafestas 1997).....	29
10 pav. Žinių bazės integravimas	37
11 pav. Žinių integravimas į duomenų bazines	38
12 pav. Blokinė žinių bazės išsidėstymo schema.....	40
13 pav. Projekto procesų ryšys (Neverauskas <i>et al.</i> 2007).....	42
14 pav. Geležinis trikampis (Vasiliauskienė 2011).....	43
15 pav. Veiksnių įtaka projektui (Vasiliauskienė 2011).....	44
16 pav. Redmine projektų valdymo sistema (Redmine 2011).....	47
17 pav. OpenGoo projektų valdymo sistema (Feng Office 2011).....	48
18 pav. Factory Nova (StuffedGuys 2011).....	49
19 pav. Microsoft Project (Microsoft Project 2011).....	50
20 pav. Užduoties sprendimo procesas	56
21 pav. Papildyta Teamwork sistemos duomenų bazė.....	57
22 pav. Panašių projektų biudžetų langas.....	59
23 pav. Panašių projektų laiko įvertinimo langas.....	60
24 pav. Panašiuose projektuose dirbusių įmonės darbuotojų langas.....	62
25 pav. HD langas	63
26 pav. Užduoties priežasties ir išsprendimo langas	63
27 pav. Užduoties priežasties ir sprendimo įvedimo langas.....	64
28 pav. Paieška žinių bazėje.....	65
29 pav. „Top resolutions“ VSB	66

LENTELIŲ SĄRAŠAS

1 lentelė. Tradicinės ir žiniomis pagrįstos sistemų palyginimas	21
---	----

SANTRUMPOS

IT (angl. *Information Technology*) – informacinės technologijos

HD (angl. *Helpdesk*) – IT pagalbos tarnyba

HTTP (angl. *HyperText Transfer Protocol*) – pagrindinis metodas pasiekti informaciją pasauliniame tinkle (*www*)

HTTPS (angl. *Hypertext Transfer Protocol Secure*) – saugus HTTP protokolas, kurį UNIX sistemoje valdo SSL

LDAP (angl. *Lightweight Directory Access Protocol*) – centralizuotai (serveryje) laikoma adresų duomenų bazė

SSL (angl. *Secure Socket Layer*) – kriptografinis protokolas, skirtas informacijos perdavimo internete šifravimui

UNIX – operacinė sistema

VSF – vartotojo sąsajos sudėtinis blokas (angl. *portlet*)

IVADAS

2008 m. Jungtinių Amerikos Valstijų (toliau - JAV) atliktų tyrimų duomenimis, net 60% visų JAV darbuotojų yra žinių darbuotojai, o 4/5 darbo vietų yra sukurtos pramonės šakose, kurios intensyviai naudoja žinias (Probst *et al.* 2006). Vystantis informacinių technologijų (toliau – IT) sektoriui, buvo sukurta nemažai produktų, skirtų pagrindiniams organizacijos darbams atlikti. Tuo tarpu dėmesys į žinių valdymo sistemų kūrimą buvo atkreiptas sąlyginai neseniai.

Žinios yra brangiausias organizacijos nematerialusis turtas, pagrįstas augančios grąžos dėsniu: kuo daugiau žinios yra naudojamos, tuo didesnę vertę jos įgauna. Veiksmingai valdomas ir naudojamas organizacijos nematerialusis turtas įgyja nuo kelių iki kelių šimtų kartų didesnę vertę nei materialusis turtas.

Organizacijos neretai susiduria su problema, kai darbuotojai savarankiškai mokosi ir įgauna įgūdžius, kuriuos jau turi kiti kolegos. Taip dubliuojamas darbas, daromos jau anksčiau padarytos klaidos ir randami jau anksčiau rasti sprendimai (Kirvaitis 2003).

Efektyvus sukauptų žinių panaudojimas leidžia greičiau ir kokybiškiau atlikti darbus, perteikti patirtį kitiems organizacijos darbuotojams bei nepamiršti praeityje įgytų pamokų. Nuolatinė konkurencija ir iššūkiai skatina organizacijas siūlyti naujas paslaugas ar produktus, ieškoti būdų įgyti pranašumą prieš konkurentus. Todėl organizacijų vadovai vis daugiau pastangų skiria darbui su žiniomis kaip su valdomu ištekliumi ir vis aktyviau bando išnaudoti savo darbuotojų žinių potencialą. Tam daug investuojama į žinių valdymą bei žinių valdymo sistemas, paremtas žinių bazėmis, kurios leidžia išgauti iš sukauptų žinių maksimalią naudą.

Šiame darbe nagrinėjama, kaip pasinaudojant sukauptomis žiniomis ir patirtimi organizacijos projektų valdymo sistemoje galima efektyviai ir kokybiškai atlikti darbą bei sutaupyti laiko, pinigų ir išvengti galimų rizikų.

Baigiamojo magistro darbo tikslas – pateikti siūlymus projektų valdymo efektyvumui gerinti bei integruoti žinių bazę į pasirinktą projektų valdymo sistemą ir pristatyti veikiančią programos modelį.

Šiam tikslui atlikti keliami tokie uždaviniai:

1. Išanalizuoti projektų valdymo procesus.
2. Susipažinti su egzistuojančiomis projektų valdymo sistemomis.
3. Išsiaiškinti žinių ir žinių vadybos sąvokas ir svarbą įmonės lygyje.
4. Susipažinti su žinių bazėmis, jų sintakse ir struktūra.
5. Išsiaiškinti projektų valdymo sistemos su integruota žinių baze poreikį.

6. Pasirinkti projektų valdymo sistemą ir integruoti žinių bazę.

Praktinė vertė – projektų valdymo sistema su integruota žinių baze leidžia mokytis iš anksčiau įgytos patirties, todėl gerėja atliekamų projektų mažesniais kaštais.

Lietuvos 14-oje Jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“ (Vilnius, Vilniaus Gedimino technikos universitetas, 2011 m. balandžio 22 d.) buvo pristatytas straipsnis „Žinių bazės integravimas į projektų valdymo sistemas“.

1. ŽINIŲ VALDYMAS

1.1. ŽINIŲ VADYBA

Žinių vadyba – tai strategijų ir praktikų rinkinys, naudojamas organizacijoje išvalgumo ir patirties identifikavimui, kūrimui, platinimui bei priėmimui. Išvalgumas ir patirtis apima žinias, kurias turi asmenys arba yra sukaupotos organizacijos procesuose ir praktikoje. Žinių vadyba populiarumą įgavo paskutiniaisiais XX a. dešimtmečiais, kai žinių vadyba buvo pradėta taikyti žinių bazėse, ekspertinėse sistemose, žinių sandėliuose, intranete. Taip žinių vadyba atrado savo vietą verslo administravimo, informacinių sistemų, vadybos, medijos, kompiuterių mokslų, visuomenės sveikatos ir kitose srityse. Daugelis didžiųjų įmonių ir nekomercinių organizacijų daug dėmesio skiria išteklių vidaus žinių valdymui, kuris dažnai apibrėžiamas kaip įmonės strategijos dalis ar žmogiškųjų išteklių vadyba.

Žinių vadybos tikslas orientuojamas į organizacijos tikslus, tokius kaip įmonės konkurencingumo didinimas, novatoriškumas, patirties kaupimas ir perdavimas. Žinių vadyba yra tiesiogiai susijusi su efektyvumu: valdomos žinios padeda tvarkytis su šiandieninėmis situacijomis bei efektyviai numatyti ir kurti jų ateitį. Jei žinios nėra valdomos, kiekviena situacija yra grindžiama asmens ar asmenų grupės veiksmiais. Jei žinios yra valdomos – kiekviena situacija yra grindžiama asmens ar asmenų grupės patirtimi, įgauta susidūrus su panašia situacija praeityje.

Žinių vadybos tikslai:

- 1) normatyviniai – bendrieji: sukurti palankią vidinę kultūrą, t.y. siekiama, kad sukaupta patirtimi ir žiniomis būtų dalinamasi, o kilus problemai jos būtų pasiūlytos;
- 2) strateginiai – kompetencijos išsaugojimo ir kėlimo sistema: tik kvalifikuota komanda gali imtis ir sėkmingai įgyvendinti sudėtingus ir brangius projektus;
- 3) operatyviniai – normatyvinių ir strateginių tikslų transformavimas į konkrečius uždavinius.

Vienas pagrindinių žinių vadybos uždavinių yra išreikštinių žinių išgavimas iš įvairių informacijos šaltinių ir sukaupytų žinių kaupimas bei dalinimasis. Dalinimasis sukauptomis žiniomis leidžia individualią patirtį paversti kolektyvine ir pakelti bendrą organizacijos darbuotojų kompetencijos lygį. Dar vienas svarbus uždavinys – neišreikštinių žinių transformavimas į išreikštines žinias, t. y. žinių išgavimas iš darbuotojų minčių ir konvertavimas į struktūras. Teigiama, kad net 85% organizacijos darbuotojų žinių yra nepatalpintos duomenų bazėse, o laikomos el. laiškų, įvairių dokumentų, darbuotojų asmeninių dokumentuose.

I organizacijų vykdomus projektus ekspertai įtraukiami tik kompleksinėms problemoms spręsti. Taip yra dėl to, kad ekspertų žinios ir sukaupta patirtis brangiai kainuoja, todėl jų „pirkimas“ tiesiogiai įtakoja projekto kainą. Taigi, organizacija, žinodama, kad sukauptos žinios daro įtaką jos sėkmei, savo veiklos srityje skatina savo darbuotojus plėtoti ir dalintis žinias tarpusavyje. Geriausia praktika – tai projektai, turėję pasisekimą rinkoje, nes jų vykdymo metu sukaupiama patirtis kaip efektyviai naudoti išteklius, laiku ir kokybiškai įvykdyti užsibrėžtus tikslus ir rezultatus.

Priklausomai nuo organizacijos veiklos, skatinamas tos srities žinių plėtojimas. Žinių kaupimui naudojami įvairūs žinių platinimo būdai. Tai gali būti įvairios hibridinės – žmogaus ir IT derinio – sistemos, intelektualų išplanavimas, kai vadovai nusprendžia kokie padaliniai turi kontaktuoti ir netiesiogiai juos apjungia, ar priverstinis komandų sudarymas. Organizacijos kaupiamos žinios turi būti nuolat plečiamos jas papildant naujai išmoktomis „pamokomis“ bei keičiantis jomis su darbuotojais, kitomis organizacijomis ir t. t.

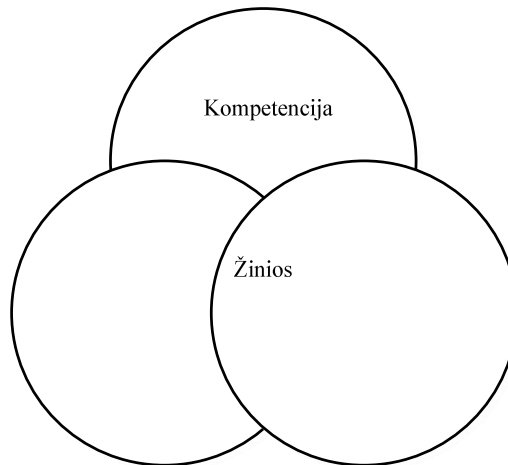
Visgi, neretai organizacijos susiduria su kultūrinėmis problemomis, kai žmonės nenori dalintis arba naudotis jau sukauptomis žiniomis. Taip nutinka kai organizacijos darbuotojai pervertina savo įgūdžius ir galimybes vykdant užduotis, nuvertina sukauptas organizacijos žinias dėl netinkamai žinių pateikimo struktūros ar tiesiog sumenkina jų autoritetą (sindromas „išrasta ne čia“). Tuomet organizacijos sukauptos ir siūlomos žinios darbuotojui atrodo varganos ir jis mieliau ieško galimų uždavinio sprendimo variantų savarankiškai.

Susidūrus su panašiomis problemomis reikalingi ryžtingi ir konkretūs vadybiniai veiksmai, verčiantys pakeisti darbuotojų sąmoningą nusistatymą prieš. Vienas geriausių konkretaus kultūrinio pokyčio pavyzdys aprašytas 4.2 skyriuje.

1.2. ŽINIOS

Žinios – tai informacija, paremta įžvalgomis ir patirtimi, kuria remiantis priimami sprendimai ir veiksmai (Levesque 2000) (1 pav.).

Žinios yra pažinimo ir įgūdžių, kuriuos asmenys vartoja sprenddami problemas, visuma, ir vienintelis išteklius, kuris kaupiasi jį naudojant. Jos apima tiek teoriją, tiek praktines, kasdienes taisykles bei instrukcijas. Žinios yra pagrįstos duomenimis ir informacija, tačiau skirtingai nuo pastarųjų, jos visuomet yra susijusios su asmenimis ir jų kompetencija (Nikolipoulou 1997).



1 pav. Žinių sąryšis su duomenimis, informacija ir patirtimi ar kompetencija (Akerkar *et al.* 2009)

Duomenys – tai daugybė išmatuojamų arba apibrėžiamų simbolių: skaičiai, raidės, paveikslai, signalai ir kt. Tuo tarpu informacija yra neformali abstrakcija, kuri yra žmogaus galvoje minčių pavidalu ir turi tam žmogui reikšmingą prasmę. Kompetencija yra žmogaus gebėjimas atlikti užduotį realioje aplinkoje, t. y. žmogus turi įgūdžių, kurie buvo įgauti ir panaudoti anksčiau atliktuose darbuose (Setzer 2011).

Dažniausiai apibrėžiamas žinių išgavimo modelis pateiktas 2 pav. Čia duomenys išgaunami iš simbolių, juos apdorojus filtruojant, formatuojant, atliekant aritmetines operacijas ir pan. Iš duomenų gautą informaciją žmogus vertina, interpretuoja ir priima sprendimą.



2 pav. Žinių išgavimo modelis (Akerkar *et al.* 2009)

Organizacijos žinių bazę sudaro individualus ir kolektyvinis žinių turtas, kurį organizacija gali naudoti vykdydama užduotis. Žinių bazė apima ir duomenis bei informaciją, kuriomis remiantis kuriamos individualios ir organizacinės žinios. Būtina paminėti, kad žinių bazėje vyksta įprasti ir nuolatiniai pokyčiai, kurie sudaro organizacinį mokymąsi. Organizacinis mokymasis – tai organizacijos žinių bazės pokyčiai, kolektyvinės koordinacinių sistemų sukūrimas ir augantis organizacijos gebėjimas veikti bei spręsti problemas (Probst *et al.* 2006).

Klasikinių gamybos veiksnių valdymo metodai ir įrankiai buvo nuolat tobulinami, tačiau profesionalių žinių turto valdymo įrankių kūrimas į priekį nepasistūmėjo. Dėl to organizacijos pernelyg retai naudojasi turimais intelektualiais išteklių: nepakankamai panaudojami ir neugdomi specifiniai

darbuotojo įgūdžiai ir joms gali nepavykti įgyti konkurencinio pranašumo naudojantis tam tikromis stiprybėmis, pavyzdžiui, puikiu IT išmanymu.

Įmonės suvokia, kad produktų vertė gali išaugti, jei jie bus susiję su intensyviu žinių naudojimu. Pavyzdžiui, produktui gali būti suteikta galimybė pačiam prisitaikyti prie kintančių sąlygų arba rinkti ir saugoti informaciją bei naudoti ją vartotojo labui. Tokia žinių bazėmis paremta sistema ypač aktuali didelėse įmonėse, kur būtų iškart siūlomi galimi problemos sprendimo variantai ar kompetentingiausi darbuotojai spręsti kilusiai problemai. Taip būtų sutaupoma daug laiko neieškant reikalingų darbuotojų ir problemos sprendimo variantų. Taip įmonės, investavusios į žinias, pagrįstus produktus, greitai įgauna svarų konkurencingumą rinkoje.

1.3. ŽINIOMIS PAGRĮSTA SISTEMA

Žiniomis pagrįsta sistema yra vienas iš pagrindinių dirbtinio intelekto (toliau - DI) šeimos narių. Turint lengvesnę galimybę programuoti ir kitus išteklius, dabar dėmesys nukrypo į sudėtingesnes užduotis, kurios gali reikalauti protinių gabumų. Visuomenė ir industrija vis labiau orientuojasi į žinias ir remiasi skirtingais ekspertų sprendimais, kad išspręstų kilusias problemas. Žiniomis pagrįsta sistema yra veiksmingas įrankis, kuris siūlo kolektyvines vieno ar daugiau ekspertų žinias ir gali veikti kaip ekspertas bet kur ir bet kada. Be to, tokia sistema leidžia sutaupyti pinigų, teikia vartotojams aukšto lygio funkcionalumą ir užtikrina neprieštarinamumą.

Žiniomis pagrįstos sistemos pagrindas – gerai sudaryta jos duomenų bazės struktūra ir surinktos žinios bei faktai. Tam geriausiai tinka reliacinė duomenų bazės struktūra, kurioje pateikiami duomenų ryšiai tarp parametrų, įrenginių ir vykdytojų galimybių bei apdirbamų detalių parametrų. Tuomet formuluojamos tikrinimo taisyklės ir išvados.

Skirtumai tarp tradicinės kompiuterinės informacinės sistemos ir žiniomis pagrįstos sistemos pateikti 1 lentelėje.

1 lentelė. Tradicinės ir žiniomis pagrįstos sistemų palyginimas

Tradicinė kompiuterinė sistema	Žiniomis pagrįsta sistema
Garantuotai pateikia sprendimą ir koncentruojasi į efektyvumą.	Prideda jėgos prie sprendimo ir koncentruojasi į efektyvumą (neteikdama garantijos sprendiniui).
Duomenų ir (arba) informacijos apdorojimas.	Žinių ir (arba) sprendimų apdorojimas.
Padedą sprendimų priėmimui, teikia reikalingą informaciją.	Kompetencijos perdavimas. Priima žiniomis pagrįstą sprendimą, jį paaiškina ir, jei reikia, atnaujina.
Naudojamas skaitinis ir algoritminis metodas.	Naudojamas simbolinis ir nealgoritminis manipuliacijos metodas.
Šios sistemos nedaro klaidų.	Šios sistemos mokosi iš klaidų.
Reikia visos informacijos ar duomenų.	Tinka dalinė ir nepatikima informacija, duomenys ar žinios.
Veikia kompleksinėse, integruotose ir plačiose srityse.	Veikia siaurose srityse.

Žiniomis pagrįsta sistema yra ekspertinių sistemų pagrindas. Galima teigti, kad žiniomis pagrįsta sistema yra kompiuteriu paremtos sistemos, kurios naudoja ir generuoja žinias iš duomenų, informacijos ir žinių. Šios sistemos gali suprasti informaciją, kurią apdoroja, ir gali padaryti nuo jos priklausomą sprendimą. Tuo tarpu tradicinė kompiuterinė sistema nežino arba nesupranta apdorojamos informacijos ir duomenų.

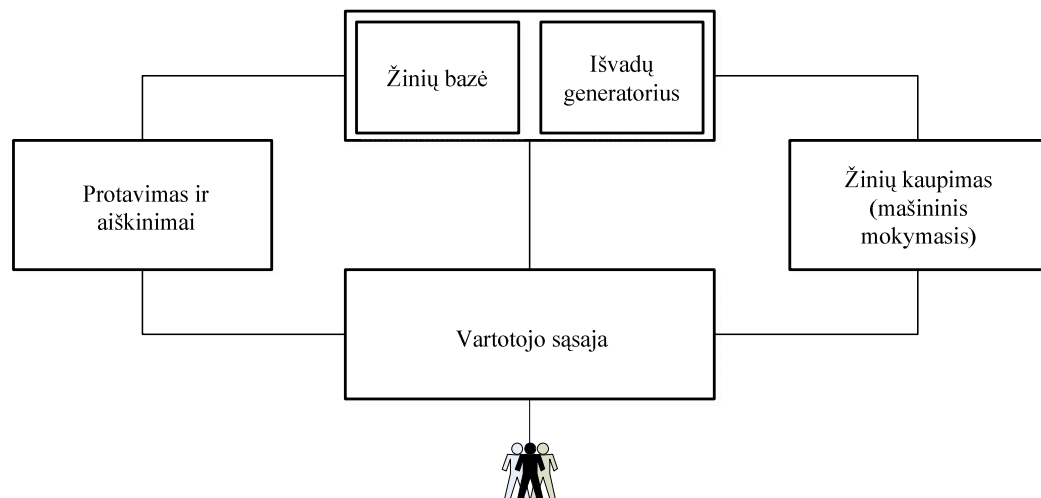
Žiniomis pagrįsta sistema sprendžia problemas siauroje srityje ir negali būti universalus problemų sprendėjas. Pavyzdžiui, žiniomis pagrįsta sistema neturėtų stengtis spręsti problemas plačioje plotmėje kaip „medicinos sritis“. Tokia žiniomis pagrįsta sistema turėtų specializuotis kurioje nors iš medicinos šakų, pavyzdžiui, „širdies ligos“.

Kompiuterinė programa neturi ekspertinės sistemos pavadinimo, nes tam, kad ji veiktų kaip ekspertas tam tikroje srityje, ji turi ne taikyti žinomus algoritmus, o naudotis sukauptais faktais ir taisyklėmis. Todėl žiniomis pagrįsta sistema simuliuoja sprendimo priėmimą ir problemos sprendimo

galimybes, kaip žmogus - ekspertas savo kompetencijos ribose, ir yra tikimasi, kad ji veiks taip pat kaip ir žmogus – ekspertas (Leondes 2005).

1.4. ŽINIOMIS PAGRĮSTOS SISTEMOS SANDARA

Žiniomis pagrįstą sistemą sudaro keturios pagrindinės dalys: žinių kaupimo modelis, žinių bazė, vartotojo sąsaja ir išvadų generatorius. 3 paveikslėlyje pateikiama žiniomis pagrįstos (ekspertinės) sistemos visuotina bendra architektūra:



3 pav. Žiniomis pagrįstos sistemos architektūra (Akerkar *et al.* 2009)

Išvadų generatorius manipuliuoja žinių bazėje esančiais duomenimis ir faktais, kad išspręstų problemą, aprašytą duomenų bazėje esančia informacija. Išvadų generatoriuje yra programinė įranga, kuri daro išvadas iš žinių, esančių žinių bazėje. Žinių bazė gali būti naudojama kaip įvairių formų žinių talpykla. Tai gali apimti tuščią vietą, kurioje bus saugomi laikini rezultatai ir informacijos ar žinių dalys.

Kadangi eksperto jėga priklauso nuo jo paaiškinimo ir priežasčių galimybių, ekspertinės sistemos patikimumas taip pat priklauso nuo paaiškinimų ir veiksmų, lėmusių priimti vienokius ar kitokius sprendimus. Žinių bazei kurti naudojamas žinių įgijimo posistemis, kuris stebi pokyčius aplinkoje ir prireikus atnaujina žinių bazę.

Žmonės gali išmokti naujų dalykų, tačiau pamiršti žinias, kurias naudoja nereguliariai. Toks mokymasis yra esminis žiniomis pagrįstos sistemos komponentas. Žiniomis pagrįstos sistemos gyvavimo trukmė gali labai skirtis priklausomai nuo tokio mokymosi. Žiniomis pagrįstos sistemos gali

būti nuolat papildoma arba automatiškai atnaujinama pačios mašinos (mašininis mokymasis). Idealiu atveju žiniomis pagrįstos sistemos pagrindą retai kada reikia modifikuoti.

Kiekviena žiniomis pagrįsta sistema privalo turėti priimtina vartotojo sąsają. Priešingu atveju rizikuojama, kad vartotojas atsisakys dirbti sistema, nepaisant kitų jos teikiamų privalumų. Vartotojo sąsaja gali turėti natūralios kalbos apdorojimo galimybę, kai vartotojas su sistema bendrauja tekstiniu būdu (Levesque *et al.* 2000).

1.5. ŽINIŲ BAZĖ

Pagrindinis žinių bazių uždavinys – automatizuoti vartotojo pagalbą, kur informacija būtų lengvai randama ir lengvai prieinama. Žinių bazė gali būti naudojama ieškant atsakymų į dažnai pasitaikančius vartotojo klausimus ar problemas ir gali būti integruota į internetinę taikomąją programą, kad kiekvienas vartotojas galėtų laisvai prie jos prieiti ir atlikti konkrečią jam reikalingos informacijos paiešką. Mažų mažiausiai, žinių bazę galima pateikti kaip indeksuotą rinkinį DUK (dažnai užduodami klausimai) su geru paieškos įrenginiu. Gerai pritaikyta ir išvystyta žinių bazė projektų valdymo sistemą paverčia nepamaitomu įrankiu valdant didelį kiekį sudėtingų projektų.

Žinių bazė susideda iš specifinių srities ir kontrolės žinių, kurios yra skirtos padėti išspręsti kilusias problemas srityje. Šios žinios gali būti išgautos iš ekspertų arba jas gali įgyti pati sistema besimokydama. Per žinių įgijimo procesą, žinių inžinierius (sistemos kūrėjas) sukaupia problemos sprendimui reikalingas žinias iš srities ekspertų ir kitų šaltinių, tokių kaip knygos, internetas, ir t. t. Automatizuotai įgydama žinias sistema naudoja mašininio mokymosi technika, kad galėtų tiesiogiai įgyti žinias iš gautos duomenų eilutės.

Konceptualus įgytų žinių modelis yra keičiamas į kompiuteriui valdyti patogesnę formą. Toks procesas vadinamas žinių vaizdavimu. Žinios žinių bazėje gali būti vaizduojamos ir laikomos įvairiomis formomis. Pavyzdžiui, vienas dažniausiai naudojamų būdų atvaizduoti žinioms yra *if - then* taisyklių forma. Sistema, kuri naudoja taisykles žinioms vaizduoti, yra vadinama taisyklėmis paremta žinių sistema. Taisyklė turi formą:

IF (sąlyga yra patenkinama)

THEN (veiksmas ar išvedimas prie naujų išvadų).

Pavyzdžiui,

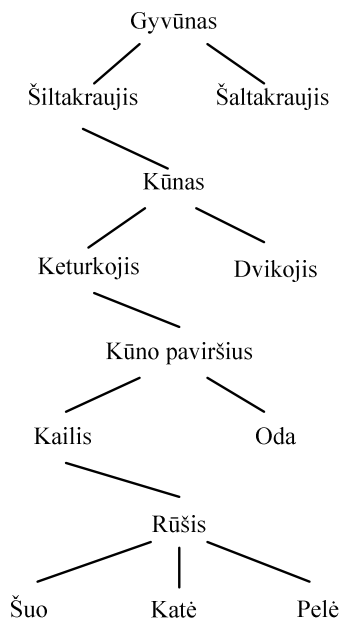
IF (temperatūra yra aukšta) *THEN* (slėgis yra žemas).

Žinios taip pat gali būti atvaizduotos semantiniais tinklais, sąvokų grafais, freimais (angl. frame) ir objektiškai orientuotomis schemomis bei Petri tinklais.

1.6. ŽINIŲ VAIZDAVIMAS

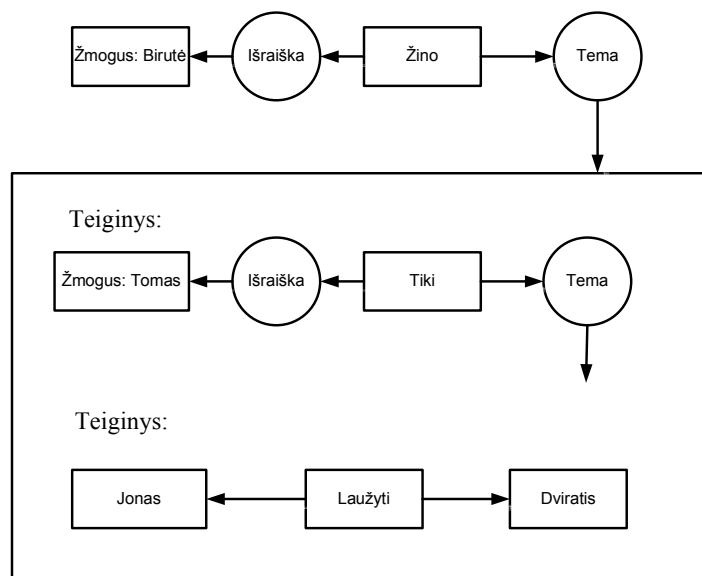
Pirmą kartą semantiniai tinklai buvo panaudoti dirbtinio intelekto sistemose, skirtose kompiuteriniam vertimui. Semantinis tinklas yra duomenų ryšių modelio tiesioginis vaizdavimas, žinias vaizduojantis grafiniu – mazgų ir ryšių tarp jų – būdu. Jame tinklo stereotipas yra naudojamas atskirai apibūdinti kiekvieną realios situacijos atvejį. Semantiniame tinkle mazginiai taškai vaizduoja objektus arba jų požymius, o jungiančios linijos rodo ryšius tarp mazginių taškų. Tinklas yra tiesioginis grafas, kuriame mazginis taškas, nuo kurio prasideda linija, yra vadinamas mazginiu objekto tašku, o taškas, kuriame linija baigiasi, vadinamas mazginiu reikšmės tašku to ryšio atžvilgiu. Kadangi semantinis tinklas yra stereotipinio pobūdžio ir skirtas įvairiems objektams vaizduoti, mazginio objekto taško lizdas gali būti interpretuojamas kaip vieta, numatyta objekto reikšmei saugoti.

Pagrindinis semantinių tinklų bruožas – deklaratyviai atspindimos žinios. Apibrėžiamasis (*definitional*) semantinis tinklas yra seniausiai žinomas semantinis tinklas, apibrėžtas dar III a. 4 pav. pateikiamas apibrėžiamojo tinklo pavyzdys: katė yra šiltakraujis keturkojis gyvūnas, turintis kailį.



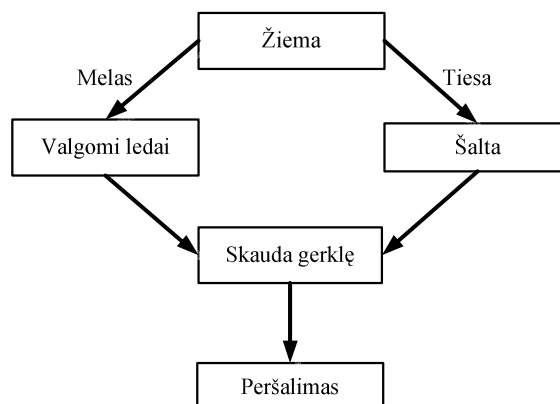
4 pav. Apibrėžiamojo tinklo pavyzdys

Tvirtinamasis (*assertional*) semantinis tinklas yra skirtas teiginių tvirtinimui. Čia teiginiai laikomi teisingais, jeigu nėra pažymėta kitaip. Pavyzdžiui, Birutė žino, kad Tomas tiki, jog jo dviratį sulaužė Jonas (5 pav.).



5 pav. Tvirtinančiojo tinklo pavyzdys (Akerkar *et al.* 2009)

Išvadų (*implicational*) tinklai, priklausomai nuo interpretacijos, gali būti vadinami tikėjimo (*belief*), kauzalinio arba priežastinio (*causal*) ar tiesos patvirtinimo (*truth maintenance*) tinklais.



6 pav. Išvadų tinklas (Akerkar *et al.* 2009)

Tokiam grafui gali būti pritaikyti keli – loginis ir tikimybinis – samprotavimo būdai. Loginis samprotavimas gali būti pradedamas tiek nuo viršaus, tiek nuo apačios – tai įgalina atrasti tinkle slypinčias klaidas. Tuo tarpu tikimybiniame samprotavime pateiktas pavyzdys gali būti papildytas tikimybėmis. Pavyzdžiui, ne visuomet žiemą skauda gerklę, ne visuomet valgomi ledai, kai yra ne žiema.

Vykdomasis (*executable*) tinklas turi mechanizmą, leidžiantį perduoti duomenis, požymius ir atlikti tam tikras duomenų pernešimo, perdavimo, išvadų darymo ar šablonų ir asociacijų paieškos procedūras. Dažniausiai naudojami mechanizmai vykdomuosiuose tinkluose:

- 1) pranešimo perdavimas. Pranešimas gali būti vienas bitas, skaitinis svoris ar net palyginti didelis pranešimas, kuris gali būti perduotas iš vieno mazgo į kitą;
- 2) prijungtos procedūros atlieka jose aprašytus veiksmus su mazgų duomenimis;
- 3) grafo transformacijos leidžia sujungti kelis grafus į vieną ar juos išskaidyti į smulkesnius.

1962 m. K. A. Petri aprašė naują informacinių srautų sistemos modelį. Kuriant modelį, buvo laikytasi prielaidos, kad atskiros sistemos dalys funkcionuoja asinchroniškai ir konkuruoja tarpusavyje. Ryšiai tarp šių atskirų dalių buvo vaizduojami grafu arba tinklu. Dažniausiai Petri tinklai naudojami modeliuojant sistemas, kuriose vienu metu gali įvykti keletas įvykių ir yra apribojimai procesų dažnumui, sekai. Įvykiams yra suteikiami prioriteto įvertinimai. Tai labai patogus būdas informaciniams procesams aprašyti, kuomet valdymo sistemose vyksta konfliktinės situacijos, lygiagretūs atsitiktiniai ir nedeterminuoti procesai.

Freimas (angl. *frame*) – tai semantinė duomenų struktūra stereotipinėms situacijoms vaizduoti. Freimo idėją 1975 m. pasiūlė Minsky: freimas sukoncentruoja visas žinias apie objektų klasę su vieninga duomenų struktūra. Taip susiejamos deklaratyvinės ir procedūrinės žinios. Freimų teorijos esmė – žmogus priimamus faktus mintyse tapatina su konkrečiais elementais, jų reikšmėmis, taip pat su tam tikrais „šablonais“ kiekvienos konkrečios sąvokos atveju.

Jei sprendžiamas uždavinys, kompiuteriui nėra svarbi uždavinio prasmė, nes viskas paverčiama dvejetainėmis sekomis. Jei kompiuteris turi imituoti žmogaus mąstymą ir suvokimą, sprendamas intelektualų uždavinį, kompiuterio atmintyje turi būti tam tikros duomenų struktūros, pagal kurias galima būtų lyginti duomenų aprašus ir priskirti juos vienai ar kitai klasei.

Freimas susideda iš lizdų

$\langle (lizdo\ vardas), (lizdo\ reikšmė) \rangle,$

kur lizdo vardas gali būti vien tik vardas-žymuo, bet kokio sudėtingumo tekstas arba kitas lizdo vardas. Sudarant freimus, sudaromos tiek kompozicijos, tiek ir dekompozicijos:

$\langle (freimo\ vardas),$

$(1\ lizdo\ vardas), (1\ lizdo\ reikšmė),$

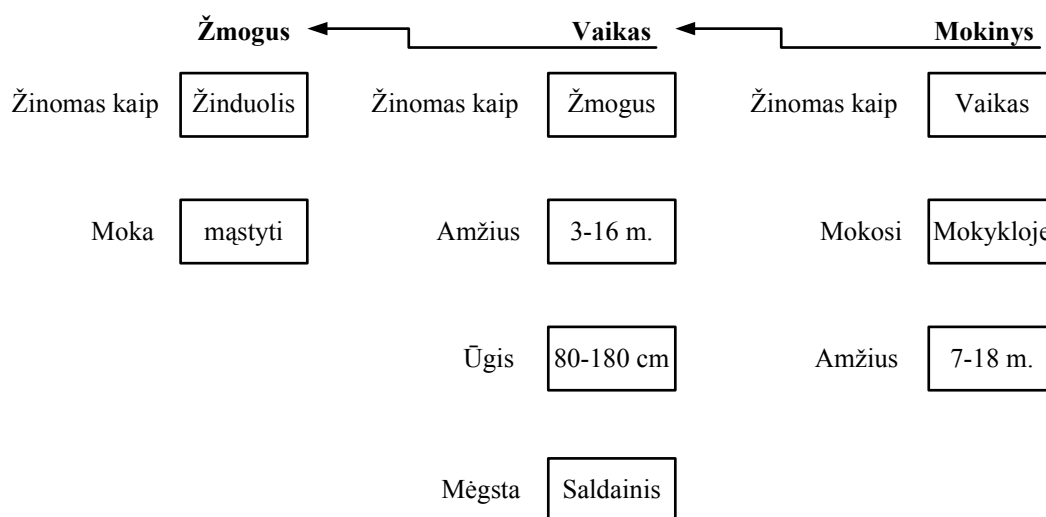
$(2\ lizdo\ vardas), (2\ lizdo\ reikšmė),$

.....

$\rangle$

Su freimais galima atlikti santykio operacijas, jungties, sankirtos operacijas. Freimai taip pat turi ir paveldėjimo savybę.

Pavyzdyje (7 pav.) pateikiamas žmogaus – vaiko – mokinio savybių paveldėjimas. Žmogus, yra žinomas kaip žinduolis, kuris moka mąstyti. Vaikas – tai žmogus (t. y. yra žinduolis ir moka mąstyti), kurio ūgis 80—180 cm, amžius 3—16 m. ir mėgsta saldinius. Mokinys yra vaikas, t. y. jis yra žinduolis, kuris moka mąstyti, 80—180 cm ūgio, patenka į 3—16 m. amžiaus grupę ir mėgsta saldinius; be to, mokosi mokykloje (Laučius 2011).



7 pav. Freimo pavyzdys

Loginių sakinių užrašymas taisyklingai sudarytomis formulėmis vadinama predikatų logika. Pavyzdžiui,

$$sninga \rightarrow - karšta$$

reiškia, kad jeigu sninga, tuomet nėra karšta.

Pagrindinė predikatų logikos idėja yra ta, kad sakiniai išreiškia santykius tarp objektų taip pat, kaip ir objektų rūšis ar atributus. Objektai gali būti žmonės, kiti fiziniai objektai ar sąvokos. Tokio tipo santykiai arba atributai vadinami predikatais. Objektai vadinami predikato argumentais arba termiais. Predikatai turi teisingumo reikšmę, bet, skirtingai nuo teiginių, predikatų teisingumas priklauso nuo termų. Predikatas gali būti teisingas vienam termų rinkiniui ir neteisingas kitam.

Predikatų logika tik remiasi argumentacijos metodais. Tokios argumentacijos yra vadinamos predikatų logikos išvedimo taisyklėmis. Jei turime rinkinį teiginių, kuriuos laikome teisingais, tai šios taisyklės garantuoja, kad gautos išvados yra teisingos. Predikatai gali būti panaudoti veiksmų vaizdavimui ar veiksmo santykiams tarp dviejų objektų. Galima apibrėžti predikatus, kurie turi

abstraktų ryšį su jo termiais, ir atlikti loginius veiksmus: konjunkciją, disjunkciją, neigimą, sekimą, tapatingumą. Taisyklingai sudarytų formų predikatų pavyzdžiai:

Klasiokai(Petras, Jonas)

kaimynai(Petras, Marytė)

Jei predikato argumentai yra faktai, tada šie atitinkami predikatai kaip termai gali būti kintamieji kitose predikatinėse funkcijose. Termo pavyzdys, kur argumentas išreikštas predikatu:

kaimynai(Klasiokas (Juozas), Klasiokas (Antanas))

Šie argumentai yra funkcinės išraiškos, kuriose atsispindi priklausomybė nuo predikato parametrų.

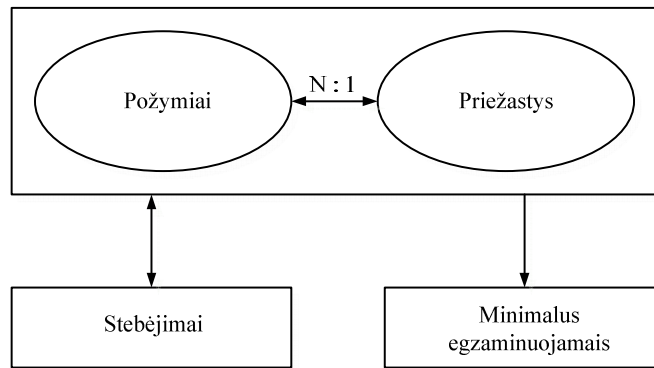
Dedukcija apibrėžiama kaip nežinomo principo pagrindimas žinomui, t.y. dedukcija yra logiškai teisinga išvada. Dedukcija yra visuotinai priimtina, suprantama ir pažįstama iš pagrindinių išvedimo metodų. Jos esmę sudaro teiginių ir predikatų logika.

Abdukcija yra pagrindimo metodas, dažnai naudojamas paaiškinimams generuoti. Skirtingai nuo dedukcijos, jos rezultatas nėra teisingos išvados, taigi tai nėra loginis išvedimas. Todėl ji vadinama išvedimo nutylėjimo taisykle. Taigi, abdukcija įgalina nustatyti tokius atvejus, kai nėra tikslumų. Abdukcijoje pradedama nuo pasekmių ir pereinama prie sąlygų, kurios galėtų sukelti šias pasekmes, nustatymo.

Indukcija yra trečiasis pagrindimo metodas. Indukcija apibrėžiama kaip pagrindimas pereinant nuo atskirų faktų arba individualių atvejų prie bendrų reiškinių. Kaip ir abdukcija, indukcija nėra pripažintas loginis išvedimas. Daugeliu atvejų abdukcija laikoma tik indukcijos forma (Tamulynas 2008).

Žinių bazė, prie labiau sudėtingų atvaizdavimo struktūrų palaikymo, taip pat turi grupę tradicinių duomenų bazių privalumų – gali aprūpinti didelius rinkinius aplikacijų, leidžia dalintis ir efektyviai kaupti ir išrinkti didelius informacijos kiekius. Žinių bazių sistema gali būti suprata kaip sistema, kuri palaiko konceptualų struktūros lygį žinių vaizdavimui, kai tuo tarpu aprūpina visais darbais, kurių mes galime tikėtis iš duomenų bazių sistemos.

Naudojamose žinių bazių lentelėse kaupiamos žinios apie ryšius tarp visų įmanomų požymių, susijusių priežasčių ir siūlomų išpėjimų. Žinias daugiausia sudaro išvadas darančios taisyklės. Žinant, kad ryšys tarp požymių ir priežasčių yra N:1 (8 pav.) dar nesumažina problemos iki vienos klaidos diagnozės. Taip yra todėl, kad egzistuoja bent vienas požymis atitinkantis dvi klaidas ir sistemos dinamiškumas, diagnozavimo metu, leidžia sudaryti įvairias kombinacijas su klaidomis bei požymiais, kurie nepriklauso vienai klaidai.

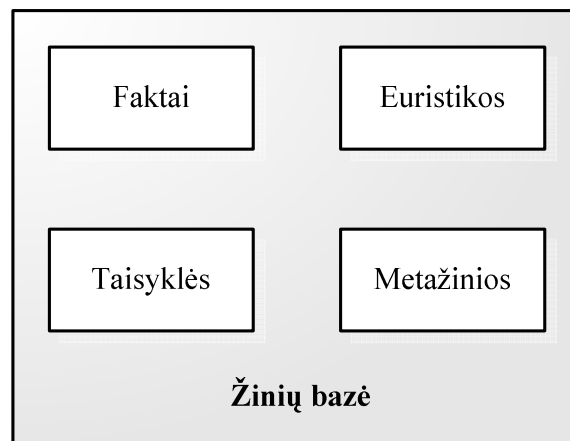


8 pav. Žinių bazė (Akerkar *et al.* 2009)

Galimos taisyklės atėjo iš įgytos patirties naudojant / palaikant sistemą ir neteikia pakankamos informacijos ryšiams tarp sąlygų iš išvadų. Jei gilios žinios būtų įmanomos, t. y. sistemos diagnozės aprašymas, galimas pilnas funkcinis argumentuotas / aiškinantis aprašymas diagnostinėms išvadoms. Netgi sistemos simuliacija gali būti pasiekta (Krishna 1992).

1.7. ŽINIŲ BAZIŲ SANDARA

Žinių bazė yra pagrindinė žinių bazėmis paremtos sistemos dalis. Sistemos kokybė ir naudingumas tiesiogiai priklauso nuo joje sukauptų žinių. Žinių bazė susideda iš žinių, pateiktų įvairiomis formomis. Akivaizdu, kad žinių bazė turi būti sudaryta iš srities žinių, kuriomis remiantis sistema sprendžia problemas. Žinių bazėje taip pat turėtų būti laikomos ir metažinios. Bendrąja prasme, jiems priklauso ontologijos – struktūra, kurioje kaupiamos srities žinios kartu su jų naudojimo metodais. 9 paveikslėlyje pateikiami žinių bazės sudedamieji komponentai:



9 pav. Žinių bazė (Tzafestas 1997)

Faktai aprašo rinkinį stebėjimų, alfabeto, simbolių ar teiginių, kuriais remiantis bus generuojamas uždavinio sprendimas. Faktų pavyzdžiai:

Žemė sukasi apie Saulę.

Visi automobiliai turi akumuliatorius.

Taisyklės aprašo sąlygas ir veiksmus, dar žinomus kaip priežastys ir pasekmės. Kad būtų išvengta neapibrėžtumo, į taisykles įtraukiamas netikrumo faktorius. Taisyklių pavyzdžiai:

Jeigu lauke šviesu, tuomet saulė yra danguje.

Jeigu neužsiveda automobilis, reikia patikrinti akumuliatorių ir kuro kiekį.

Be faktų ir taisyklių, į žinių bazes įtraukiamos ir euristikos, kurios pagerina žinių bazių efektyvumą. Euristika – tai būdas atvaizduoti problemos sprendimo patirtį žinių bazės viduje. Čia kaupiami sprendimai, kuriuos priimtų ekspertas analogiškoje situacijoje. Kadangi euristikos nėra sukurtos standartiniu būdu ir forma, jos negali suteikti problemos išsprendimo sėkmės garantijos. Taisyklės ir faktus lengva išgauti ir užrašyti. Kita vertus, euristikos yra kaupiamos ekspertų galvose neišreikštų žinių pavidalu. Taigi, euristikas sunku suformuluoti, aprašyti ir užrašyti. Euristikų pavyzdžiai:

Jeigu įvyko visiškas saulės užtemimas, negali būti dienos šviesos (nors saulė ir yra danguje).

Jeigu dabar yra lietingas oras ir automobilis pervažiuos per balą, į automobilio duslintuvą gali patekti vanduo ir automobilis gali užgesti.

Metažinios yra žinios apie žinias. Metažinios yra naudingos ieškant reikalingų žinių ir taikant jas problemos sprendimui. Jos naudojamos ontologijose ir naudinga sprendžiant problemas, lyginant problemų charakteristikas ir situacijas, kuriose metažinios gali būti pritaikytos, taigi palengvina operacijų darbą. Metažinių pavyzdžiai: planavimas, modeliavimas, žymėjimas, mokymasis ir srities žinių redagavimas.

1.8. ŽINIŲ BAZIŲ TIPAI

Taisyklėmis paremta žinių bazė pradėta formuluoti apie 1980-uosius. Tokia žinių bazė dažniausiai susideda iš taisyklių rinkinio, kur taisyklės yra aprašomos kaip

IF <sąlyga>

THEN <išvada>

teiginiai. *IF* dalis aprašo sąlygą, kuriai esant įvyks *THEN* nurodyta išvada. Kai tik programa pamato kokį nors pokytį, ji ieško tinkamos taisyklės, kurią galėtų pritaikyti.

Taisyklių sistemos automatizuoja samprotavimus, išveda naujus faktus, leidžiančius priartėti prie problemos sprendimo naudodamos paieškos metodus ir atitikmenį pagal šabloną. Problemos sprendimo procesas ieško kelio nuo žinomo problemos apibrėžimo iki jos sprendinio. Tokia išvedimų serija yra vadinama išvedimų grandine.

Taisyklėmis pagrįstos sistemos samprotavimo procesas vyksta nuo turimų duomenų iki išvados, tačiau būdai išgauti rezultatui yra skirtingi. Vienose situacijose yra ieškoma turimų duomenų išvadų, kai duomenų turime nedaug ir galime patikrinti visus galimus jų rezultatus. Pavyzdžiui, pamestas asmens dokumentas; į kokias institucijas ir kokia eilės tvarka kreiptis. Kitose jau žinome išvadas bet nežinome priežasčių. Pavyzdžiui, žinome, kad neužsiveda automobilis ir ieškoma visų tai galėjusių įtakoti priežasčių (išsikrovęs akumulatorius, nėra kuro ir t. t.)

Šios dvi taisyklių paieškos rūšys vadinamos duomenimis varomas arba tiesioginis išvedimas ir tikslo varomas arba atgalinis išvedimas. Duomenimis varoma paieška dar vadinama grandinės į priekį (*forward chaning*) metodu. Programa turi suprasti pateiktus faktus ir iš jų padaryti išvadą. Grandinė prasideda sulyginus turimus faktus su taisyklėmis iš taisyklėmis pagrįstos žinių bazės. Kai tik taisyklė bazėje randama, tos taisyklės išvada pridėjama prie žinomų faktų rinkinio. Šis naujas faktas gali būti sulyginamas su kita taisykle. Antrasis paieškos tipas yra vadinamas atgaline grandine (*backward chaning*), kai programa jau žino išvadą, bet nori ją patikrinti.

Taisyklėmis pagrįstos žinių bazės leidžia panaudoti eksperimentines žinias, išgautas iš žmogaus – eksperto. Tai leidžia žinių bazei gerai veikti nedidelėje apibrėžtoje srityje. Jei sritis yra apibrėžta, pakankamai maža ir gerai aprašyta, tuomet įmanoma aprašyti visas joje galinčias nutikti tikimybes ir numatyti situacijas. Be to, taisyklėmis paremta žinių bazei programai lengva rasti uždavinio išvadą.

Deja, tokia sistema taip pat turi ir trūkumų. Dažniausiai žmogaus – eksperto pateiktos žinios yra euristinės ir yra sunku surinkti gilesnes srities žinias. Jei srities žinios nėra labai aiškos, visuomet yra tikimybė, kad iškils nenumatyti atvejai, kurių aprašytų taisyklių programa tiesiog neras savo žinių bazėje. Čia euristinės taisyklės negali susitvarkyti su trūkstama informacija ar netikėtomis duomenų reikšmėmis. Taigi, jei reikšmė yra netikėta ar jos iš vis trūksta, programa negalės parinkti taisyklės ir sugeneruoti išvados. Dažnai, taisyklių rinkinys yra adaptuotas konkrečiam uždaviniui ir jo neįmanoma adaptuoti kitam, todėl žinios yra labai priklausomos nuo užduoties.

Priešingai nei taisyklėmis paremta žinių bazė, atvejais paremta žinių bazė įsimena prieš tai buvusį atvejį į atvejų bazę. Atvejis rodo situaciją, kuri nutiko kažkada praeityje. Bendrai paėmus, tai atvaizduojama kaip „problema – sprendimas“ pora. Problema rodo nutikusios situacijos aprašymą ir sprendimą, kuris yra traktuojamas kaip sprendimas priimtas tuo metu (išspręsti iškilusiai problemai).

Atvejų bazė susideda iš atvejų rinkinio. Programos veikimas paremtas turimos situacijos ir jai analogišku atveju, turėto praeityje, sugretinimu. Šiai žinių bazei nereikia plačios srities žinių analizės ir leidžiamos nuorodos į priežastis ir veiksmus. Taigi, jei susiduriama su aprašytu atveju, sprendimas gali būti rastas labai greitai (greičiau nei generuojant sprendimą nuo uždavinio pradžios). Atvejais paremta žinių bazė leidžia programai išvengti jau anksčiau pasitaikiusių klaidų ir pasinaudoti ankstesne sėkme, nes kiekvienos iškilusios situacijos įrašas yra saugomas ir naudojamas naujų problemų sprendimui.

Kita vertus, atvejais paremtoje žinių bazėje sprendimo paieškos laikas (laiko sudėtingumas), priklausomai nuo varganos organizacijos bibliotekos, paieškos procese gali išaugti. Taip pat gali labai išaugti ir pačios atvejų bazės dydis (užimama vieta), kai programa mokosi naujų atvejų be sistemos veikimo gerinimo. Tuomet programa turi kartoti savo paieškas kol nustato atvejų dalį pagal panašumo į kitus atvejus reikšmę. Ir nors atvejais paremtos žinių bazės kūrimas yra sudėtingas ir ilgas, intensyvus jos naudojimas ir sukaupia patirtis sukuria didesnę efektyvumą.

1.9. ŽINIŲ BAZĖS PROJEKTAVIMAS

Žinių bazės projektavimas yra nelengvas uždavinys žinių bazių kūrėjams, kuriems tenka rasti geriausią sprendimą kaip susieti žinių bazę su sistemoje jau egzistuojančia žinių baze. Pagrindinė žinių bazės projektavimo problema – programos modelio klasifikavimas į duomenų informacijos ir žinių kryptis. Dažnu atveju, duomenys yra atvaizduojami kaip populiacijų pavadinimai, informacija – kaip ryšių tinklai, o žinios – kaip sąlygos.

Kai kurie faktai yra individualūs faktai ir jie sukuria specifinius teiginius apie specifinius objektus. Pavyzdžiui, Jonas priskirtas Ilfm-12 grupei. Kai kurie faktai yra bendro pobūdžio faktai, t. y. jis aprašo ryšius tarp populiacijų, pvz., dėstytojai veda jiems paskirtas paskaitas. Savitumas yra tarp esybių ir ryšių tipų (bendro pobūdžio faktai) ir atskirų populiacijų (individualūs faktai).

Programos modelyje teiginiai turi tenkinti žinių unikalumą, tipo identifikaciją ir atominį reikalavimą. Tai reiškia, kad kiekvienas atominis objektas yra identifikuojamas vardu ar populiacija, o kiekvienas vardas yra identifikuojamas kaip populiacijos (tipo) narys ir susideda bent iš vieno funkcinio ryšio. Žodžio unikalumas nusako, kad kiekvienas to paties žodžio panaudojimas nurodo į tą patį objektą sistemos aplinkoje ir kiekvienas duoto objekto atvejis sistemos aplinkoje yra surištas su tuo pačiu žodžiu.

Žinių bazės projektavimo etapai:

1. Analizavimo etapas. Šio etapo metu identifikuojami tikslai, įvardijamos galimos problemos ir apibrėžiamas laukiamas rezultatas, iškeliami klausimai:

- 1.1. Kokios žinios bus kaupiamos?
- 1.2. Kokia žinių struktūra bus naudojama?
- 1.3. Koks žinių bazės tipas bus naudojamas?
- 1.4. Kokios uždavinių klasės ir duomenų šaltiniai bus panaudoti?
- 1.5. Kas bus galutinis sistemos naudotojas, vartotojo sąsaja?
- 1.6. Kokie ištekliai, kompiuterinė technika bus pasitelkti?
- 1.7. Kokiais kriterijais matuojama žinių bazėje sukauptų žinių vertė ir nauda?
- 1.8. Naudotojų skatinimas naudotis žinių baze?

Analizavimo etapą galima skirstyti į smulkesnius etapus:

- 1.1. Užduoties analizavimas – ekspertas, sistemų inžinierius analizuoja vartotojo (asmens, dirbančio probleminėje srityje) poreikius ir nustato, analizuoja ir įvertina problemą, kuriai bus kuriama žinių bazė. Išanalizavus vartotojo lūkesčius, suformuluojami laukiami rezultatai bei jų nauda organizacijai. Organizacijos naudai nustatyti gali būti atlikti įvairūs tyrimai, pavyzdžiui, produkto paklausos analizė. Tuomet nustatomi reikalingi ištekliai (kompiuterinė, programinė įranga, darbuotojai, žinių ištekliai, kt.) ir išlaidos;
- 1.2. Egzistuojančios sistemos analizė skirta nustatyti tiriamos srities objektams, jų savybėms ir ryšiams tarp jų. Objektas priskiriamas daiktui, kuris egzistuoja ir atskiriamas sistemoje. Objektas ar įvykis, ar jų požymis turi būti apibrėžtas žinių bazėje. Duomenų ar objektų ryšiai išreiškiami duomenų ryšių modeliu. Objektų ir ryšių tarp jų schema parodo ryšius tarp įvairių objektų sistemoje ir suformuluoja tokių žinių bazę;
- 1.3. Pradinės žinių išgavimo stadijos metu žinių inžinierius naudodamas parankiausias žinių išgavimo metodus iš eksperto išgauna pirmines žinias. Žinių išgavimas prieš sistemos projektavimą ir reikalavimų specifikacijos sudarymą padeda lengviau suvokti galutinį sistemos rezultatą, palengvina apsispręsti kokią žinių vaizdavimo kalbą pasirinkti. Žinių išgavimo metodai:
 - 1.3.1. Pasyvūs metodai naudojami, kai ekspertas moka savo mintis išreikšti žodžiais ar veiksmais, o žinių inžinierius atlieka stebėtojo – klausytojo funkciją:
 - 1.3.1.1. Stebėjimai – susipažįstama su sritimi, srityje vykstančiais procesais bei stebimas ekspertas ir jo veiksmas darbo vietoje.

Čia gali būti stebimas realus eksperto darbo procesas arba jo imitacija (demonstracija);

1.3.1.2. „Minčių balsu“ protokolavimo tikslas – išsiaiškinti visą atliekamų veiksmų grandį. Protokolavimo metu ekspertas komentuoja savo atliekamus sprendimus ir veiksmus, t. y. ekspertas žinias išreiškia ne tik veiksmais, bet ir žodžiais. Tai – vienas efektyviausių pasyvių žinių išgavimo metodų;

1.3.1.3. Paskaita. Čia ekspertas pasakoja apie savo darbą, jo specifiką, naudojamas žinias ir atliekamus veiksmus, tačiau neatliekamas joks realaus eksperto darbo proceso ar jo imitavimo vaizdavimas.

1.3.2. Aktyvūs – individualūs. Žinių ekspertas aktyviai bendrauja su ekspertu ir išgauna žinias:

1.3.2.1. Anketavimo procedūros metu, kuri gali būti atlikta dvejopai: žinių inžinierius klausinėja eksperto iš anksto paruoštus klausimus arba ekspertas pats raštu užpildo anketą. Anketa turi būti įvairi, su kontroliniais klausimais. Čia taip pat svarbi klausimų seka ir loginis atsakymų pagrindimas.

1.3.2.2. Interviu metu yra lankstus žinių surinkimas, nes sekantys klausimai išplaukia iš anksčiau gautų atsakymų, tačiau svarbiausi išlieka centriniai klausimai. Interviu metu užduodami kontroliniai, zonduojantys, atviri ir uždari, verbaliniai ir neverbaliniai klausimai ir t. t.

1.3.2.3. Laisvas dialogas – tai laisvas eksperto ir žinių inžinieriaus bendravimas, kai nėra jokio išankstinio plano ar klausimų. Tačiau jam reikia didžiausio ir profesionaliausio pasiruošimo (specialus, konkretus, psichologinis). Metodo trūkumas – laisvas dialogas reikalauja didelės įtampos, todėl reikalingas didelis psichologinis pasirengimas.

1.3.3. Pagrindinis aktyvaus–grupinio metodo privalumas – žinios išgaunamos iš kelių ekspertų vienu metu:

1.3.3.1. „Apvalusis stalas“. Siūlomas 3 – 7 dalyvių skaičius. Iš pradžių kiekvienas dalyvis pasisako konkrečiu klausimu,

paskui vyksta laisva diskusija. Metodo trūkumai pasireiškia, kai iškyla organizaciniai klausimai, konfliktinės situacijos ir t. t.

1.3.3.2. „Smegenų šturmas“ pirmą kartą panaudotas JAV kariniame povandeniniame laive. „Smegenų šturmas“ naudojamas kai reikia staigiai priimti sprendimus. Čia idėjos generavimas atskiriamas nuo idėjos analizės. Rekomenduojama, kad „smegenų šturmas“ truktų 40-60 min. ir dalyvautų 6-10 žmonių. Tai naujų idėjų generavimo metodas. Analizė parodo, kad 10% idėjų „smegenų šturmo“ metu sugeneruotų, yra geros arba gali būti panaudotos.

1.3.3.3. Ekspertiniai žaidimai – tai žaidimas (realios situacijos imitacija), kurios atspinti realias veiklas. Tai daugiausia diagnostiniai, loginiai, kompiuteriniai ir kt. žaidimai, kurie „ištraukia“ žinias. Dalykiniuose žaidimuose pateikiama dalykinės srities situacija. Tada analizuojami sprendimai, įvertinami dėsningumai ir t. t.

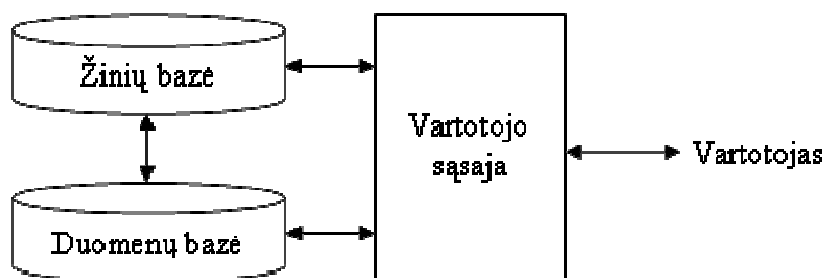
- 1.4. Reikalavimų specifikacijos sudarymas yra vienas iš pagrindinių žinių bazės sistemos kūrimo etapų. Jo metu žinių inžinierius sistemina informaciją, gautą analizavimo ir pradinių žinių išgavimo etapuose, ir paruošia sistemos specifikaciją. Galiausiai specifikacijoje apibrėžiami iškelti tikslai, naudosimos priemonės ir laukiami rezultatai.
- 1.5. Preliminaraus projektavimo metu nustatomos pagrindinės sąvokos, atributai, ryšiai tarp sąvokų, hierarchija, klasės, modulumas, įeinantys ir išeinantys duomenys ar jų struktūros, t. y. sukuriamas neformalus žinių aprašymas.
- 1.6. Žinių formalizavime žinių inžinierius, bendradarbiaudamas su programuotoju, pasirenka konkretų žinių vaizdavimo metodą.
- 1.7. Vidinių prototipų sudarymas ir įvertinimas. Programuotojas, remdamasis žinių inžinieriaus sudaryta reikalavimų specifikacija, pasirenka atitinkamas programavimo priemones ir sukuria prototipus, kurie leidžia patikrinti pagrindines žinių bazės savybes. Sukurtus prototipus vertina vartotojas ir dar ankstyvajame žinių bazės kūrimo etape – esant būtinybei – atlikti esminius pakeitimus.

2. Vystymo etape sukurtas prototipas vystomas iki galutinio, vartotoją tenkinančio lygio:
 - 2.1. Galutinis projektavimas – tai konkrečių priemonių ir išteklių pasirinkimas. Į projektavimą įtraukiamas žinių vaizdavimas ir atliekama detali architektūros analizė. Į projektuojamą žinių bazę žiūrima kaip į „juodąją dėžę“, kuri pateikia atsakymus į korektiškai suformuluotus klausimus. Projektuojant žinių bazę patartina ją izoliuoti nuo kitų komponentų: tuomet ji tampa moduli, todėl su žinių baze galima elgtis kaip su atskiru programiniu moduliu, kuris gali būti panaudotas kelis kartus ar integruotas į kitus programinius modulius.
 - 2.2. Daugiausia laiko užimantis etapas – realizavimas, kurio metu integruojama žinių bazė, sukuriami vartotojo sąsaja, išvadų generatorius ir sistemos samprotavimas bei atliekamų veiksmų aiškinimas. Taip pat ypač daug dėmesio skiriama problemos sprendimo paieškos greičio optimizavimui.
 - 2.3. Testavimo etape programuotojas, žinių inžinierius, ekspertas ir galutinis vartotojas vertina sistemą. Nagrinėjamas projektų valdymo strategijos efektyvumas, išsiaiškinamos klaidos ir reikalingi pakeitimai, prototipo privedimui iki galutinės versijos.
3. Tobulinimo etapas. Žinių inžinierius perduoda tolimesnį darbą su sistema ekspertui. Pastarasis įvertina sistemą iš išorės, ar gerai suvestos taisyklės, uždavinių kokybė, sistemos kūrėjų kompetenciją. Programuotojas papildomai gali atlikti jautrumo analizę, kai yra keičiamos struktūros ir aplinka, vartotojo sąsaja. Dėl galimų poreikių pasikeitimų ir iškeltų pastabų prototipo kūrimo metu, sistema redaguojama tol, kol išgaunamas norimas, vartotoją tenkinantis sistemos pavidalas.
4. Sistema suderinama su kitomis programinėmis priemonėmis, informacinėmis ir ekspertinėmis sistemomis, duomenų baze.
5. Palaikymo etapas panašus į vystymo, kai papildomai grįžtama į žinių išgavimo žingsnį, kur iš eksperto išgaunamos gilesnės žinios ir taip papildoma žinių bazė. Atsiradus naujiems vartotojo poreikiams, sistema modifikuojama (Akerkar *et al.* 2009) (Laučius 2009).

2. ŽINIŲ BAZĖS INTEGRAVIMAS

Kompiuterinės sistemos per pastaruosius tris dešimtmečius labai patobulėjo. Šiandien programinės įrangos technologija yra padalinta į dvi pagrindines sritis: sisteminę programinę įrangą ir taikomąją sisteminę įrangą. Sisteminė programinė įranga skirta kompiuterinėms sistemoms kurti ir valdyti, o taikomoji programinė įranga projektuojama siekiant padėti automatizuotai spręsti problemas, išskylančias įvairiose žmogaus veiklos srityse. Ji ypač plačiai pradėta naudoti naujų gaminių ir jų gamybos technologijų projektavimo ir gamybos procesų valdymo srityse.

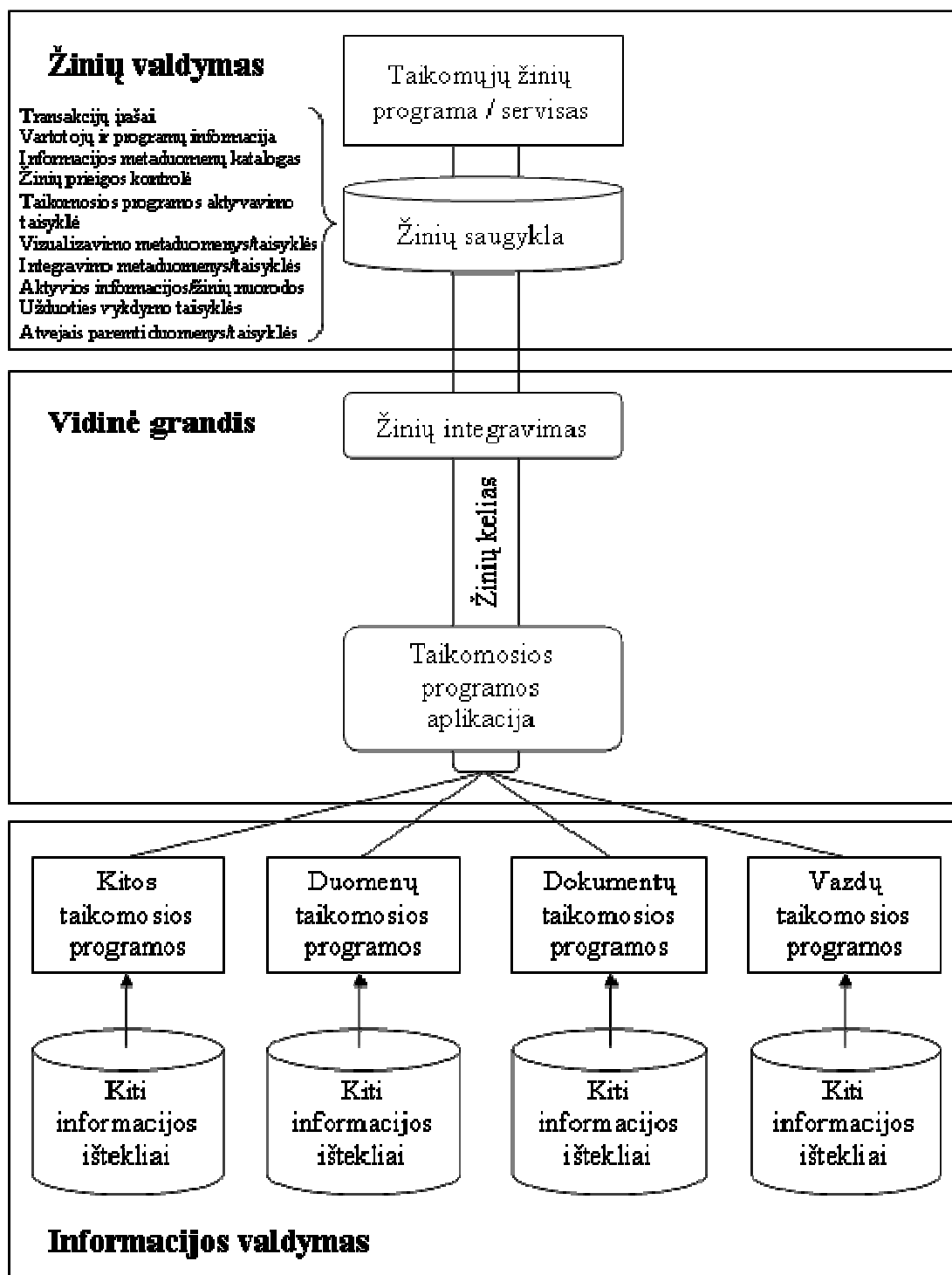
Žinių integravimas – tai asmens žinių perkėlimas į specifinę situaciją. Žinių bazės integravimo tikslas nėra sumažinti žinių skirtumo tarp atskirų asmenų ar jų grupių, o sujungti specifines žinias su atitinkamais veiksmais. Pats žinių bazės integravimas į taikomąją programinę įrangą yra konceptinis ir techninis iššūkis. Integravimui keliamas uždavinys sujungti intelektualizavimo komponentus su jau veikiančia sistema, o pačiai integruojamai žinių bazei panaudojami duomenys, esantys jau egzistuojančiose duomenų bazės lentelėse (10 pav).



10 pav. Žinių bazės integravimas

R. M. Swartzas aprašė ir užpatentavo informacijos ir žinių integravimo modelį. 11 pav. pavaizduotas aukštesnio lygio programinės įrangos architektūra, skirta žinių integravimo sistemoms. Čia sistema padalinta į tris atskirus programinės įrangos lygius – informacijos valdymą, vidinę grandį ir žinių valdymą. Informacijos valdymo lygyje dirba daugybė informacijos valdymo taikomųjų programų.

Vidinėje grandyje esantys funkcionalumai užtikrina įvairių informacijos valdymo taikomųjų programų darbą ir teikia išteklius, kurie leidžia stebėti įeinančią ir išeinančią informaciją tarp įvairių informacijos valdymo taikomųjų programų (žinių integravimas). Žinių integracijos blokas tikrina įeinančią informaciją ir per žinių saugyklą gauna instrukcijas iš žinių valdymo lygio.



11 pav. Žinių integravimas į duomenų bazes

Žinių integracijos blokas pateikia žinių saugyklai įeinančius duomenis, t. y. teikia transakcijų įrašus, gautus iš vartotojų ar taikomųjų programų, bei kitą gautą informaciją. Taikomoji žinių programa / servisas savyje turi savybių, kurios įgalina efektyvų darbą ir integruotų žinių priskyrimą sistemoje. Pavyzdžiui, aktyvių žinių nuoroda gali būti panaudota norint vartotojui pasirinkti ir įvykdyti keletą

užduočių, kurias vykdytų viena ar daugiau informacijos valdymo taikomųjų programų. Tiksliau sakant, žinių nuoroda gali būti apibrėžta susiejant dokumentą su susijusiu objektu duomenų analizės sistemoje. Bet kokia dokumento nuoroda būtų sukurama kai tik dokumentas pateikiamas vartotojui reikalingu formatu (pvz., *.pdf). Tuomet vartotojas gali pasinaudoti žinių nuoroda į žinių saugyklą ir išgauti apibrėžtą kiekį užduočių, kurios gali inicijuoti taikomosios programos transakcijų rinkinį.

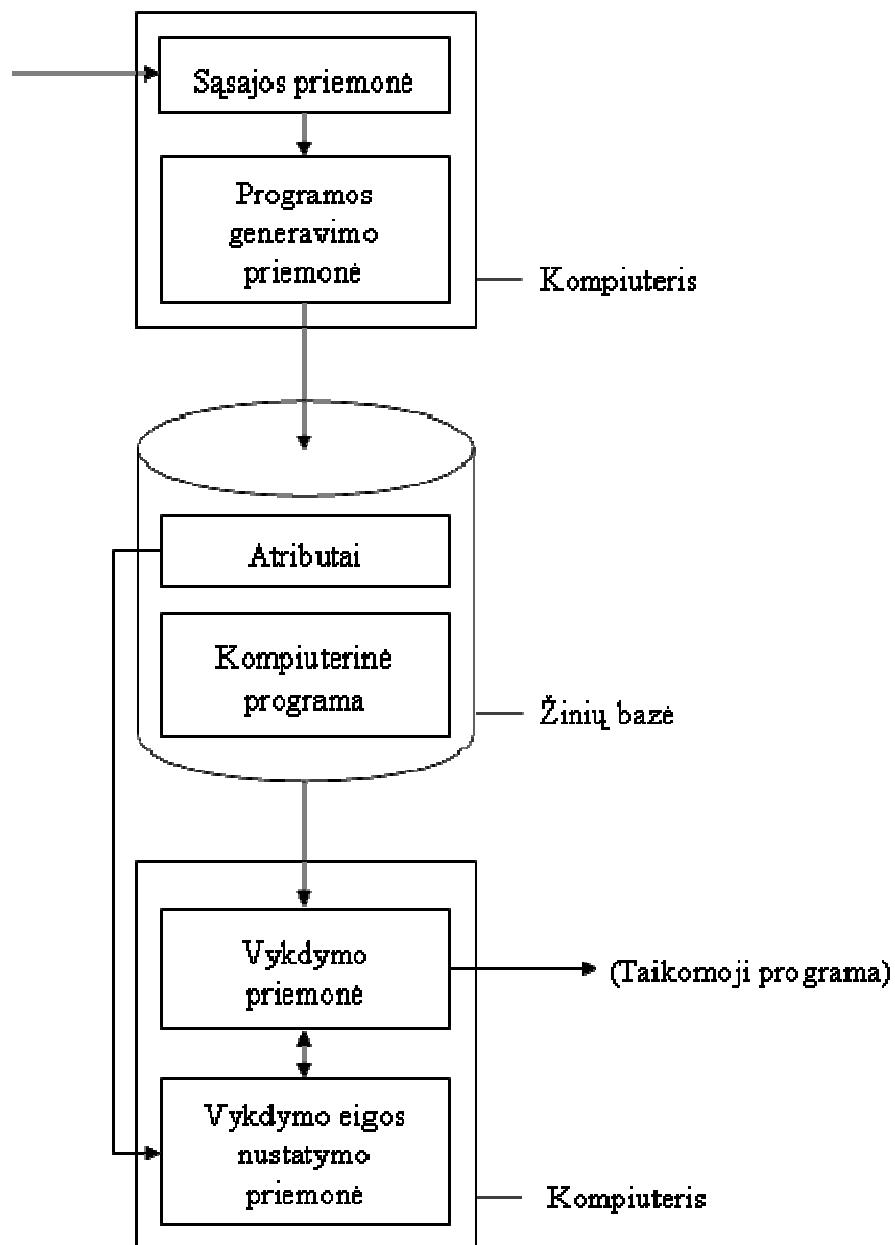
Šioje architektūroje aprašytos savybės ir funkcionalumai yra skirtos žinių darbuotojo darbui, siekiant suteikti galimybę dirbti su įrankiais, nedarant įtakos integruotos analizės efektyvumui ir sprendimo priėmimo procesui. Esminė šios architektūros savybė – visų reikalingų darbų įtraukimas į žinių kelią. Toks funkcionalumas reikalauja tekstinio vartotojo bendravimo, pavyzdžiui, nustatyti kodėl vartotojas atlieka pasirinktą veiksmą (Swartz *et al.* 2001).

12 pav. pavaizduota patentuota S. Matsushita, M. Itoh ir M. Kawanobe blokinė žinių bazės sistemos išsidėstymo diagrama. Čia žinių bazė yra sukurama viename kompiuteryje ir vykdoma kitame. Žinių bazių sistemą sudaro žinių bazės vykdančysis įrenginys, kuris naudojasi vykdymo eigos nustatymo priemonėmis, kad išgautų ryšius tarp žinių kintamųjų.

Žinių bazių sistema sudaryta iš žinių bazę kuriančio įrenginio, žinių bazės vykdančiojo įrenginio bei žinių bazę naudojančio įrenginio. Žinių bazę kuriantis įrenginys naudojasi žinių bazės sąsajos priemone, kuri turi vartotojo žinių bazę, ir programos generavimo priemone, kuri generuoja kompiuterines programas, atsižvelgdama į žinių bazės sąsajos priemonėje esančias žinių bazes. Pati žinių bazės sąsajos priemonė yra sudaryta iš žinių lentelių, kurios laukuose saugomi įeinantys ir išeinantys kintamieji.

Žinių bazę kuriantis įrenginys gali paleisti žinių bazę kuriančią programą, instaliuotą visiems prieinamame kompiuteryje. Analogiškai, žinių bazės vykdančysis įrenginys gali paleisti programą, instaliuotą visiems prieinamame kompiuteryje, žinių bazės kūrimui. Nors šie įrenginiai yra tarpusavyje nepriklausomi, tačiau programos gali būti instaliuotos tame pačiame kompiuteryje. Žmogus, turintis sukauptų žinių, naudojasi žinių bazės kuriančiuoju įrenginiu, o žmogus, kuris neturi sukauptų žinių ir nori jų įgyti, naudojasi žinių bazės naudojančiuoju įrenginiu.

Programos generavimo priemonė generuoja atributus, susijusius su žinių kintamaisiais žinių lentelėje ir sudaro lentelę vertimo funkcijų, skirtų operacijos rezultatų transformavimui į kompiuterinę programą. Ši programa gali būti iškart įvykdyta vykdymo priemonių arba programa gali būti sukompiliuota kompiliatoriumi.



12 pav. Blokinė žinių bazės išsidėstymo schema

Žinių bazę naudojantis įrenginys sudaro vykdymo eigos nustatymo priemonę, priklausomai nuo sugeneruotų vertimo funkcijos kintamųjų, ir vykdymo priemonę programų vykdymui. Vykdyto sekos nustatymo priemonė sudaro kontrolines funkcijas pagal įeinančias užklausas, kai metodas apskaičiuoja vertę (Matsushita *et al.* 1997).

3. PROJEKTŲ VALDYMAS

3.1. IT PROJEKTŲ VALDYMAS

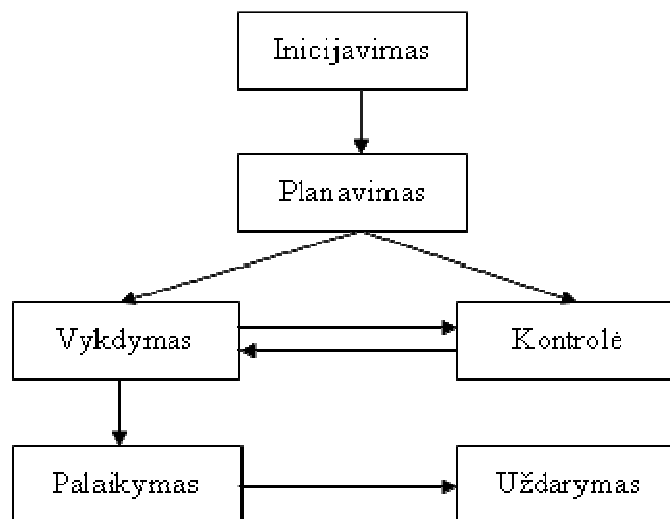
Projektų valdymo institutas (*Project Management Institute*) pateikia tokį projekto apibrėžimą: projektas – tai laikina veikla, kuria siekiama sukurti unikalų produktą ar paslaugą, turinti pradžią, pabaigą, valdoma žmonių ir įtakojama laiko, kainos, išteklių ir kokybės. Projektas yra laikina veikla, nes turi būti įgyvendintas per apibrėžtą laikotarpį su tiksliais pradžios ir pabaigos datomis. Kadangi projekto metu sukurto produkto ar paslaugos unikalumą rodo jų naujumas, ne kiekviena veikla gali būti vadinama projektu.

Projektų valdymas – tai veikla, apimanti žinias, patirtį, metodus, priemones ir technologijas, skirta projekto tikslui pasiekti (Laučius *et al.* 2007). IT projektų valdymas yra sudėtingesnis nei, tarkim, inžinerinių projektų (tilto, namo statyba) valdymas. Viena priežasčių – IT projekto metu kuriamas produktas gali būti ne visuomet aiškiai apibrėžtas arba žinomas iki vėlyvųjų produkto kūrimo etapų. Be to, dažnai programinės įrangos projektai yra kuriami pagal individualų užsakymą, todėl nesinaudojama iš anksto paruoštais programinio kodo moduliais, o yra kuriamas visiškai naujas programinis kodas (Walker 2001).

Vienas svarbiausių projekto uždavinių yra aiškiai identifikuoti tikslai. Juos projekto vadovas turi parinkti tokius, kad jie būtų priimtini visiems (ar daugumai) projekto dalyvių, nes tik tokiu atveju projekto dalyviai galės susikoncentruoti į tai, kas projekte svarbiausia. Siekiant sėkmingai įgyvendinti projektą, turi būti rastas sprendimo būdas ir sudarytas planas kaip šis sprendimas bus įgyvendintas.

IT projekto vykdyme dalyvauja programinės įrangos specialistų komanda, kuriems dažniausiai tenka kelios rolės arba atsakomybė už keletą sričių. Esant didesnei komandai ar komandai, kurioje vienas žmogus yra atsakingas už kelias sritis yra sunku suvaldyti projekto darbus ir jų atlikimo terminus. Be to, pasitaiko atvejų, kai tampa nebeaišku, kuris iš projekto dalyvių turi atlikti vieną ar kitą projektinį darbą. Kita vertus, viena geriausių IT projektų savybių yra projekto lankstumas. Kuriamą programinę įrangą gali būti lanksčiai modifikuojama prie besikeičiančios aplinkos ar vartotojo (užsakovo) poreikių, todėl net įpusėjus projektą dažniausiai nereikia visko „nugriauti ir statyti iš naujo“.

Visi projektai pereina keletą skirtingų, tarpusavyje susijusių procesų, vadinamų projekto valdymo procesais (13 pav.).



13 pav. Projekto procesų ryšys (Neverauskas *et al.* 2007)

Projektų valdymo procesai skirstomi į 6 pagrindinius procesus:

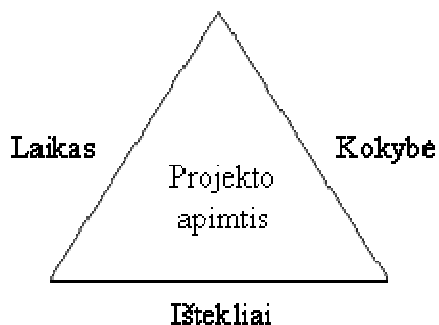
1. Inicijavimo procesas. Tai projekto gyvavimo pradžios momentas. Šio proceso metu nustatomas projekto tikslas, generuojamos ir atrenkamos idėjos kaip bus pasiektas projekto tikslas ir priimamas sprendimas pradėti projektą.
2. Planavimo procesas. Nustatomi tikslai ir vertinimo kriterijai projekto metų sukurto produktui. Paruošiama reali schema projektų valdymui, atrenkamas ir projektui priskiriamas personalas. Identifikuojamos ir įvertinamos galimos rizikos.
3. Vykdomo proceso metu projekto komandos darbuotojams priskiriamos užduotys, nustatoma jų atlikimo seka ir terminai.
4. Sekimo ir kontroliavimo procesas. Čia stebima ir vertinama projekto eiga. Esant būtinybei, priimami sprendimai, keičiantys projekto eigą.
5. Palaikymo proceso metu sukurta produktas yra pateiktas vartotojui ir naudojamas. Rasti produkto defektai taisomi, o pats produktas pritaikomas naujoje aplinkoje ir gerinama jo teikiama kokybė. Šis procesas laikomas mažiau įgūdžių reikalaujančiu procesu, todėl jį atlieka mažiau patyrę projekto komandos darbuotojai.
6. Uždarymo procesas – formalus projekto užbaigimo patvirtinimas ir gautų rezultatų įvertinimas.

Projekto valdymui, be abejo, nemažai įtakos turi ir pats projekto dydis – kuo didesnis, tuo sudėtingesnis. Projektai pagal dydžius dažniausiai klasifikuojami į trivialius, mažus, vidutinius, didelius ir labai didelius projektus. Trivialus projektas – tai maža programinė įranga, naudojama iki keleto mėnesių ir turintį 10–20 komponentų ar procedūrų, pvz., baigtinių elementų metodu

skaičiuojamos konstrukcijos deformacijos, veikiant įvairioms apkrovoms ir išorinėms jėgoms. Atlikusi darbą programa paprasčiausiai pašalinama. Mažas projektas gali būti mokslinius skaičiavimus, mažus komercinius uždavinius sprendžianti programinė įranga bei studentų projektai, turintys 25—50 komponentų ar procedūrų. Vidutiniai projektai trunka nuo keleto mėnesių iki pusės metų, kurių metu sukuriama įvairūs kompiliatoriai, verslo valdymo sistemos su 250—1000 komponentų ar procedūrų. Dideli projektai vyksta 1—2 metus ir sukuriamos sudėtingos programos, pavyzdžiui, laiko paskirstymo sistemos, duomenų bazių valdymo paketai, realaus laiko valdymo sistemos. Labai didelių projektų pavyzdžiai – telekomunikacijų sistemos, operacinės sistemos, karinės pramonės sistemos ir pan. Pavyzdžiui, operacinę sistemą OS/360 kūrė 5000 programuotojų. Tokio projekto trukmė – 5 metai, dydis apie 1 mln. kodo eilučių ir apie 100 mln. JAV dolerių kaina.

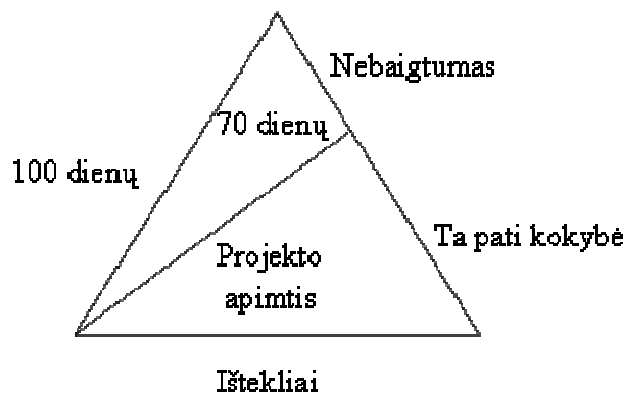
Taigi matome, kad didėjant projektų apimčiai, žymiai pasunkėja situacijos, išteklių ir laiko – svarbiausių sėkmingo IT projektų valdymo komponentų – valdymas. Maža to, tenka susidurti ir su kitais sunkumais: nepatenkintais klientais, nesuprastais ar blogais suprastais tikslais, netiksliais biudžetais ir grafikais, neužimtais arba labai užimtais ištekliais.

Tam, kad projektas būtų sėkmingai įgyvendintas, patariama remtis trimis sėkmės pagrindais – kaina, apimtimi (kokybe) ir laiku – kurie sudaro geležinį trikampį (14 pav.).



14 pav. Geležinis trikampis (Vasiliausienė 2011)

Bet kurio iš šių pagrindų kitimas lemia visą projektą. Pavyzdžiui, jei projekto įgyvendinimui buvo nuspręsta skirti 100 dienų, bet projekto vykdymo metu buvo sumažinta iki 70-ties, tuomet tam, kad projektas būtų įvykdytas laiku teks atsisakyti kai kurių projekte numatytų užduočių arba viršyti projekto vykdymo laiką, tačiau susidurti su kita – finansų ir išteklių – problema (15 pav.).



15 pav. Veiksnių įtaka projektui (Vasiliauskienė 2011)

Planuojant projektą, dažniausiai dar pačioje jo pradžioje yra sukuriama apytikslis projekto planas ir detalus planas pirmam projekto etapui. Vėliau tas pats apytikslis projekto planas yra detalizuojamas kiekvieno kito etapo pradžioje. Kiekvienas didesnis projektas turėtų turėti savo grafiką, kuriame būtų nurodyta kiekvienos veiklos (darbo, užduoties) pradžios ir pabaigos datos. Toks projekto grafikas leidžia užtikrinti, kad reikalingi ištekliai bus skiriami tada, kai jų reikės ir bus išvengta veiklų konfliktų, susijusių su tų pačių išteklių panaudojimu vienu metu, todėl nebus trukdoma suplanuotų darbų vientisumui. Be to, pagal sudarytą detalų planą galima ruošti patikimas finansines prognozes ir vertinti faktinius pasiekimus.

Projekto planas turi parodyti realią ir baigtinę veiksmų seką, leidžiančią laiku gauti rezultatą. Jei planas bus nerealias ar jo realumu netikės projektą finansuojantys ir realizuojantys žmonės, tai kris darbingumas bei projekto finansavimas. Taip pat labai svarbu, kad projektas būtų lengvai suprantamas visiems: ir tiems kurie daro esminius darbus (turi būti suvokiama, kas ir kodėl yra daroma), ir neesminius (svarbu, kad būtų matoma pati projekto esmė).

Sėkmingu projekto užbaigimu suinteresuoti visi jo pagrindiniai dalyviai. Užsakovas (savininkas, klientas) iš užbaigto projekto naudojimo gauna pajamų, projekto vadovas ir komanda – atlyginimą pagal kontraktą bei augantį profesinį reitingą. Projekto metu sukurto produkto vartotojai gauna laukiamą paslaugą, prekę ar produktą, o kitos suinteresuotos šalys patenkina savus ekonominius ir neekonominius interesus.

Galiausiai, IT projekto rezultatų vertinimas yra objektyvus dalykas, nes rezultatas retai kada būna tiksliai apibrėžtas ir kiekvienas žmogus gautą rezultatą interpretuoja savaip. Projektas laikomas sėkmingu, jei jis užbaigiamas nustatytu laiku, neperžengiant biudžeto rėmų ir užtikrina rezultatų kokybę. Kiti sėkmingo projekto kriterijai – minimalūs (suderinti) projekto objektinės srities pokyčiai, funkcionuojančios (bazinės) organizacijos veiklos nenutrūkstamumas bei organizacijos gamybinės

kultūros ir vertybių saugojimas ir ugdytas. Dažniausiai projektų įgyvendinimo rezultatus įvertina atskira konsultacinė įmonė kartu su projektą iniciavusia institucija.

Deja, valdant IT projektus dažnai susiduriama su problemiškomis jo savybėmis. Viena dažniausiai pasitaikančių projektų valdymo klaidų – tai naujų projektų vykdymas dar neužbaigus senųjų. Dažnu atveju, vėluojama užbaigti senuosius projektinius darbus, todėl sulaukiama finansinių nuostolių arba prarandama atliekamų darbų kokybė. Įvertinti kiek kainuos programinės įrangos kūrimas gali tik patyrę ekspertai, jau turintys panašių programinės įrangos kūrimo projektų patirties. Blogai įvertinus projekto kainą gali tecti atsisakyti kai kurių numatytų darbų įgyvendinimo, siekiant nevirsyti projekto biudžeto ir įgyvendinti projektą laiku.

Kita IT projektų valdymo klaidų grupė yra susijusi su projekto metu kuriamu produktu. Paprastai kuriama programinė įranga yra sudėtinga, todėl projekto komandoje dirbantys darbuotojai ne visuomet gali pilnai ją suvokti, kontroliuoti ir nustatyti koks yra projekto progresas. Be to, kūrimo metu iškyla įvairios problemos ir nauji uždaviniai, kurie gali smarkiai itakoti IT projekto įgyvendinimo sėkmingumą. Kadangi sudėtingos programinės įrangos testavimas yra sunkus ir brangus, ir pačiam testavimui numatoma per mažai laiko, programinė įranga išleidžiama su klaidomis. Taip rizikuojama geru organizacijos vardu, o klaidų taisymas kainuoja nepalyginamai brangiau.

Kitos neretai pasitaikančios klaidos: netinkamas projektų prioritetų paskirstymas, dažni užsakovo reikalavimų pasikeitimai, nėra reikiamu metu suplanuotų ir tolimesniam darbui reikalingų išteklių. Tokios klaidos lemia investuotų išteklių bei dalies planuoto pelno praradimą.

3.2. PROJEKTŲ VALDYMO SISTEMOS

Projektų valdymą galima laikyti kaip įvairių išteklių panaudojimo koordinavimo per visus projekto gyvavimo ciklus metodologiją, siekiant efektyviai pasiekti tikslą. Šiuolaikiniais valdymo metodais pasiekiami projekte numatyti darbų struktūros ir apimties, laiko, materialinių ir finansinių sąnaudų, laiko bei kokybės rezultatai, tenkinantys visus projekto dalyvius.

Projektų valdymo sistemos padeda sudaryti projekto darbų planą, materialių bei žmogiškųjų išteklių poreikio biudžetus ir tapo nepakeičiamu pagalbininku projektų vadovams. Jos taip pat laiku suteikia informaciją apie šių išteklių naudojimą ir projekto darbų įvykdymo lygį. Visa tai sumažina projektų rengimo, jų organizavimo ir įgyvendinimo trukmę ir su tuo susijusias sąnaudas.

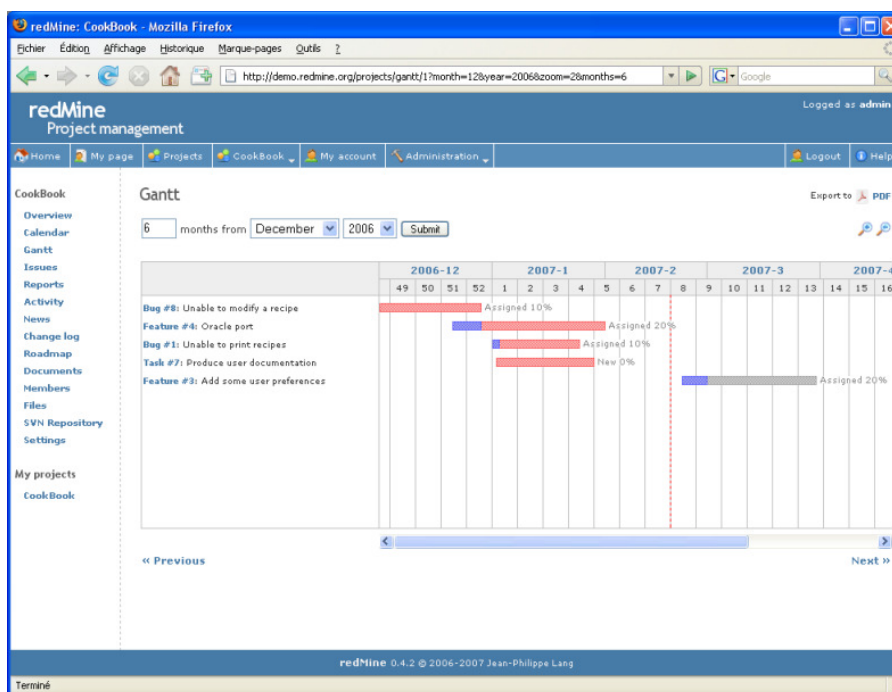
Projekto valdymo sistema nukreipta į visų jo gyvavimo ciklo fazių tikslų bei galutinio rezultato pasiekimą. Projektų valdymo sistemos posistemai:

- 1) integracinis valdymas;
- 2) projekto turinio (darbų) valdymas – administruojami projekto darbai, juos pridedant, modifikuojant ir – esant poreikiui – šalinami. Projekto darbai priskiriami atsakingiems asmenims, numatomos darbams atlikti reikalingos priemonės ir metodai;
- 3) laiko ir trukmių valdymas – nustatomas atliekamų darbų nuoseklumas ir trukmė, sudaromi grafikai, optimizuojamas projekto darbams skirtas laikas, numatomi sprendimai esant laiko nukrypimams nuo suplanuotų darbų grafiko;
- 4) sąnaudų valdymas – planuojami ištekliai, sudaromas projekto biudžetas, įvertinamas projekto pelnas (galimas nuostolis), kontroliuojamos išlaidos ir numatomos bei šalinamos kylančios rizikos;
- 5) žmogiškųjų išteklių valdymas – tai projekto komandos darbuotojų paieška, atsakomybės apibrėžimas, kvalifikacijos kėlimas, darbo apmokėjimas ir darbo krūvio paskirstymas;
- 6) informacijos ir komunikacijų valdymas – šis posistemis leidžia projekto komandai bendrauti virtualioje aplinkoje, kur jie gali būti informuojami apie vykdomo projekto aktualijas, sprendžiami su projektu susiję klausimai ir t. t.;
- 7) rizikos valdymas – tai projekto metu kylančių rizikų prognozavimas, analizė, įvertinimas ir išvengimas (pašalinimas). Projekte rizika nusakoma nelauktais, neigiamą įtaką projektui darančiais įvykiais;
- 8) kokybės valdymas – nustatomi standartai ir reikalavimai projekte sukurtam produktui / paslaugai;
- 9) kt. (Tamulynas 2008).

Toliau pateikiama trumpa kelių laisvai pasirinktų projektų valdymo sistemų apžvalga.

3.2.1. Redmine

Redmine yra atvirojo kodo internetinė projektų valdymo sistema, sukurta naudojant Ruby on Rails struktūrą (16 pav.). Redmine turi kalendorių, Ganto diagramą, kur galima vizualiai sekti projekto vykdymą iki nustatytų terminų.

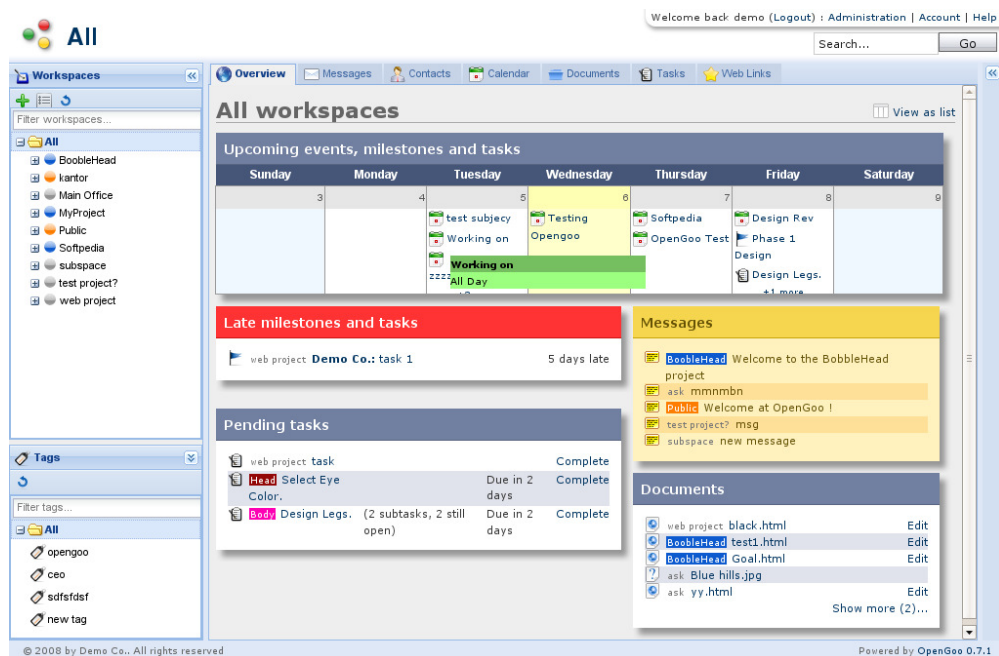


16 pav. Redmine projektų valdymo sistema (Redmine 2011)

Dar viena svarbi sistemos savybė – įmonės vikio (*wiki*) ir forumo palaikymas bei daugelio rolių turėjimas. Kitos Redmine projektų valdymo sistemos savybės: neribotas projektų ir subprojektų skaičius, dirbto laiko registravimas, automatinių pranešimų el. paštu galimybė, esant kokiems nors projekto pokyčiams, failų saugykla, vartotojo sukurti laukai, standartinės ir vartotojo sukurtos užklausos.

3.2.2. Opengoo

Feng Office Community Edition (anksčiau vadintas OpenGoo) yra Fengo Office ir OpenGoo bendruomenės sukurta atvirojo kodo programa. OpenGoo dar žinoma kaip privačios informacijos valdymo programa, leidžianti tvarkyti dokumentus, laiškus ir kontaktus, valdyti projektą ir jo vykdymo laiką (17 pav.).



17 pav. OpenGoo projektų valdymo sistema (Feng Office 2011)

Tekstiniai dokumentai gali būti kuriami ir redaguojami realiu laiku, o failai (nepriklausomai nuo jų tipo) gali būti įkelti, tvarkomi ir dalinami tarpusavyje. Darbuotojai savo puslapyje mato jam aktualią informaciją, susijusią su vykdomais projektais bei užduotimis, darbų grafikus, žinutes ir kt.

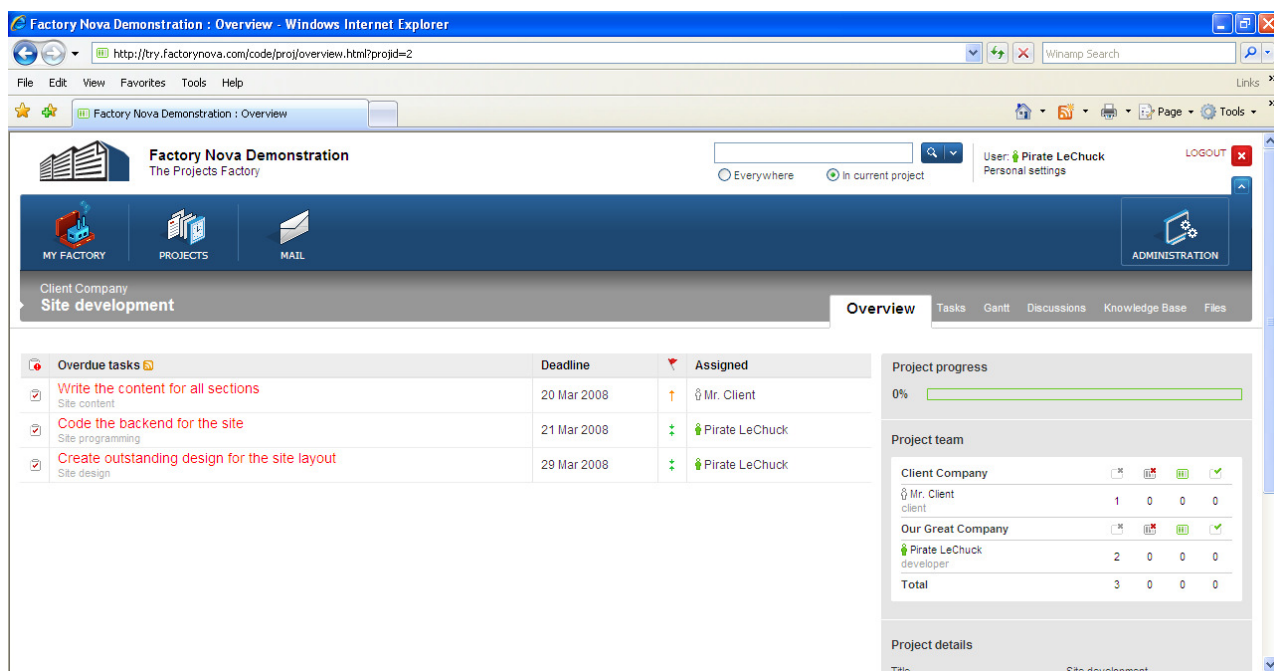
3.2.3. Factory Nova

Factory Nova projektų valdymo sistema yra pritaikyta tiek organizacijos darbuotojams, tiek organizacijos klientams, kurie gali matyti projektą ir jo vykdymo etapų bei numatytų darbų įgyvendinimą bei sukurti ir priskirti naujas užduotis projekto komandai (18 pav.).

Dirbant šia projektų valdymo sistema, lengva rasti reikalingą informaciją, patalpintą dokumentuose, užduotyse, diskusijose, žinių bazių įrašuose bei el. laiškuose, nes leidžiama kaupti visus reikalingus duomenis projekte. Taip pat realizuota galimybė apibrėžti komandos narių roles projekte ir pateikiama daug dėmesio nereikalaujanti projekto kontrolės peržiūra, kurioje pateikiama komandos narių statistika. Išmanus rikiavimas (*smart sorting*) svarbiausias užduotis bei projektus su problemomis (vėluojančiomis užduotimis) rodo sąrašo viršuje, vėluojančios užduotys pažymimos pažymimo spalva.

Projektų sistemoje daug dėmesio skirta užduočių valdymui. Todėl vartotojas turi galimybę greitai ir patogiai konvertuoti duomenis projekte: el. laiško konvertavimas į užduotį, užduoties – į diskusiją, diskusijos – į žinių bazės įrašą. Po užduoties sukūrimo ir redagavimo, klientai ir projekto

komandos nariai informuojami el. laišku apie pasikeitimą užduotyje, projekte. Pačioms užduotims ir jų aprašymui naudojamos žymos, kurioms galima nustatyti prioritetiškumą.

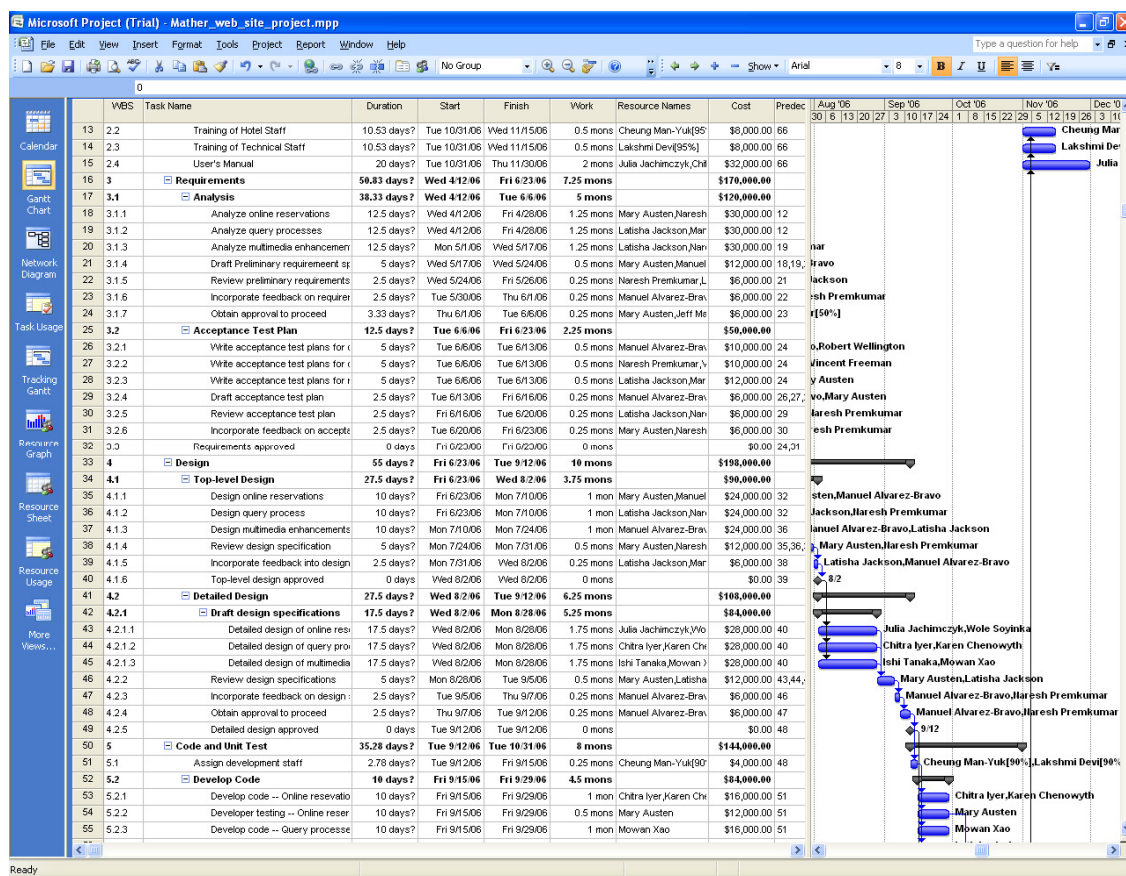


18 pav. Factory Nova (StuffedGuys 2011)

Dar viena svarbi Factory Nova savybė – laiko apskaita. Sistemoje kiekvienas komandos narys gali suvesti laiką, praleistą sprendžiant užduotį, rankiniu arba automatiniu būdu (pažymėdamas, kad pradeda spręsti užduotį ir pažymėdamas, kad baigė). Vėliau galima įvairiais pjūviais analizuoti komandos darbuotojų sugaištą laiką skirtingiems projektams ar jų užduotims. Sistema taip pat palaiko multikalbiškumą. Ir nors produktas instaliuojamas su anglų ir rusų kalbų paketais, tačiau galima prijungti ir kitus kalbų paketus.

3.2.4. MS Project

Microsoft Project (19 pav.) yra galingas Microsoft korporacijos projektų valdymo įrenginys, skirtas projektų planų sudarymui, išteklių priskyrimui, progreso stebėjimui, projekto biudžeto valdymui bei darbo krūvių paskirstymo analizei.



19 pav. Microsoft Project (Microsoft Project 2011)

Programa geba stebėti ir vertinti kritinius suplanuotų darbų kelius kritinės grandinės Ganto diagramos pavidalu, skirstyti darbuotojus į atskiras grupes, kurios turi skirtingas teises priėjimui prie projekto duomenų peržiūros.

Microsoft Project realizuota galimybė sukurti projekto biudžetą, paremtą numatytais projekto darbais ir išteklių įkainiais. Kiekvienas išteklius turi savo kalendorių, kuriame nurodyta, kada išteklius bus laisvas. Kiekvienas išteklius gali būti priskirtas kelioms užduotims ir kiekviena užduotis gali turėti kelis išteklius – atsižvelgdama į tai, programa sudaro užduočių tvarkaraštį.

4. ŽINIŲ BAZIŲ PRITAIKYMAS ĮMONĖSE

4.1. „KUONI“ KELIONIŲ AGENTŪRA

„Kuoni“ Šveicarijos kelionių agentūra nuo pat įkūrimo rėmėsi pagrindiniu šūkiu – „Rinkitės „Kuoni“ ir jūsų atostogos bus puikios“. Kai paskutinįjį praėjusio amžiaus dešimtmetį sparčiai išaugo verslo kelionių paklausa, agentūra nusprendė užkariauti ir šią rinkos dalį. Tačiau daugelis verslo kelionių buvo tarptautinės, todėl daug bilietų užsakymo bei kelionės organizavimo būdų. Tai padidino krūvį tų darbuotojų, kurie buvo atsakingi už įmonės verslo kelionių organizavimą: jie dažnai jautėsi priblokšti, jog viską turi padaryti vieni.

Kai įmonės kreipiasi į kelionių agentūras, jos tikisi aukšto lygio organizacinių sugebėjimų ir techninės kompetencijos. Be to, sąnaudų skaidrumas gali būti esminis veiksnys dėl augančių įmonės išlaidų verslo kelionėms: po atlyginimų ir informacinių technologijų (toliau – IT), kelionės dažnai priskiriamos trečiajai pagal dydį išlaidų kategorijai.

Atsižvelgdama į spartų verslo kelionių sektoriaus augimą, „Kuoni“ nusprendė iš tradicinės kelionių agentūros reorganizuotis į „Verslo kelionių informacijos valdymo įmonę“. Ji planavo savo klientams siūlyti tokią informaciją, kuri reikalinga jų verslo kelionių valdymui tobulinti. Žiniomis pagrįstos paslaugos teikimas tapo strategine sėkme.

Siekdama tapti „kelionių biudžeto patikėtine“, ji klientams siūlė įvairias paslaugas, įskaitant specifinius pasiūlymus ir specialias keliones į prekybos muges bei įprastas verslo keliones. „Kuoni“ turimose kompiuterizuotose klientų rinkmenose yra visi duomenys, susiję su įmonių, kurias jie aptarnauja, verslo kelionėmis. „Kuoni“ kaupia duomenis apie tai, kokia klase keliauja kiekvienas klientas, kokia buvo automobilių nuomos kategorija ir kokią vietą lėktuve ar maistą klientas pasirinko.

Esminė „Kuoni“ į žinias orientuoto klientų aptarnavimo dalis yra programinės įrangos sistema, skirta kelionės išlaidoms analizuoti. Ši unikali sistema, vadinama „Knows“ („Kuoni“ Nationally Offered Worldwide Statistics – „Kuoni visai šaliai siūloma pasaulinė statistika“), suteikia kompanijai galimybę surinkti ir apdoroti visus duomenis apie kliento, kuris naudojosi „Kuoni“ paslaugomis, kelionę. Klientui pageidaujant gali būti įvertintas ir pateiktas duomenų paketas, o tai užtikrina didžiausią kelionės išlaidų skaidrumą. Išlaidos skrydžiams, viešbučiams ir automobilių nuomai gali būti suskirstytos pagal kelionės tikslą, kelionės klasę, tiekiją ir laikotarpį. Greta metinės ataskaitos palyginimui pateikiami ir praėjusių metų duomenys. Vadybos informacija apie klientą pasiekia tiesiai iš visų svarbiausių kelionių tikslų per „Kuoni“ partnerių, priklausančių Tarptautinės verslo kelionių asociacijai, tinklą.

Informacija pateikiama patogią formą. Tai leidžia lengvai atsakyti į klausimus, pavyzdžiui, kuriomis oro linijomis tenka didžiausia užsakytų bilietų dalis arba kuris kelionės tikslas yra brangiausias. „Knows“ padeda klientui efektyviau stebėti verslo keliones ir nustatyti grupinių užsakymų, kuriems taikomos nuolaidos, galimybės. Ši papildoma paslauga padeda „Kuoni“ ilgiau išsaugoti visus kliento pageidavimus: kitą kartą keliaujančiam klientui įmonė, atsižvelgdama į praėjusių kelionių statistiką, gali pasiūlyti skristi kliento mėgstamiausia oro linijų bendrove, apsistoti jo mėgstamiausiame viešbutyje, kur jo jau lauks mėgstamiausias šampanas ir t. t.

„Kuoni“ sėkmė, naudojant „Knows“, rodo, kaip žiniomis pagrįsti produktai gali atverti naujų galimybių rinkoje ir sustiprinti įmonės ilgalaikį konkurencingumą. Tačiau tokie produktai meta ir naujų iššūkių. Naudojant „Knows“, „Kuoni“ pasiekė tik todėl, kad ji sukūrė tinkamas sąlygas sistemingam turimo žinių turto valdymui. Bet kuri įmonė, kuri priima žinių valdymo mestą iššūkį, pirmiausia turi išsiaiškinti, ką ji žino ir ko ji nežino. Tik tuomet ji galės sukurti strategijas, pagrįstas savo kompetencija (Probst *et al.* 2006).

4.2. „SIEMENS AG“

Siemens AG, 73 mlrd. dolerių vertės elektronikos ir elektrotechnikos konglomeratas, gamina viską nuo elektros lempučių iki rentgeno nuotraukų mašinų, nuo energijos generatorių įrangos iki greitųjų traukinių. Per 164 metų raidą, Siemens išsivystė į vieną iš didžiausių ir sėkmingiausių pasaulinių korporacijų. Žinoma, kad Siemens dirba daug aukštą techninį išsilavinimą turinčių inžinierių, bet didžioji dalis jų žinių buvo uždara ir neprieinama kitiems darbuotojams. Norėdamas maksimizuoti kiekvieno korporacijos verslo vieneto nešamą pelną, Siemens AG turėjo išmokti sutelkti žinias ir jos 460 000 tarnautojų visame pasaulyje ekspertizę.

Žinių valdymo ištakos Siemens kompanijoje prasideda 1996 m., kai įmonės vadovybė susidomėjo žinių bazėmis ir jų galimu pritaikymu Siemens. 1999 m. centrinė valdyba patvirtino žinių bazių poreikį ir galimą naudą visai kompanijai. Eidama žinių vadybos link, Siemens pristatė kelis svarbius techninius ir kultūrinius pokyčius. Taip buvo sukurta Siemens internetinė žinių vadybos sistema ShareNet, kuri sujungė duomenų bazių, pokalbių kambarių ir paieškos variklius. Taip Siemens darbuotojams buvo suteikta galimybė ieškoti reikiamos informacijos sandėliuose ir tuomet susisiekti su informacijos autoriumi, pasinaudojus kitais komunikacijos kanalais.

Visgi kompanijos vadovai suprato, kad IT yra tik įrankis realizuoti žinių kaupimo, apdorojimo ir paskleidimo kitiems darbuotojams. Amerikos regiono žinių vadybos padalinio vadovas Rendelas

Sellersas (*Randall Sellers*) teigia, kad „technologijos ar IT yra tik maža dalis visos sistemos. Manau, kad reikia 20% IT ir 80% vadybos, kad būtų galima daryti įtaką kultūriniais pokyčiams ir žmonių bendradarbiavimui“.

Siemens panaudojo įvairius būdus įtikinti darbuotojus, kad yra svarbu dalytis idėjomis, patirtimi ir žiniomis. Taigi Siemens pasamdė 100 „evangelistų“ darbuotojų visame pasaulyje, atsakingų už darbuotojų apmokymą, atsakymų į klausimus pateikimą ir sistemų priežiūrą. Buvo dėtos visos vadybos pastangos apeiti darbuotojų nenorą permainoms. Pavyzdžiui, darbuotojui įkėlus failą į sistemą ar pasinaudojus sistemoje sukauptomis žiniomis, Siemens jį apdovanoja „dalintukais“ (dažnumo žymekliais), kuriuos vėliau darbuotojas galėjo iškeisti į kompanijos gaminamos elektronikos prietaisus arba nuolaidas atostogų kelialapiams į svečias šalis.

Tačiau tikroji sistemos paskata žymiai paprastesnė. Prekybininkai suprato, kad žinios ir kitų kolegų patirtis, pasiekiamos per ShareNet, yra nepakeičiami norint laimėti pelningas sutartis. Rinkodaros, aptarnavimo, tyrimo, plėtros ir kitų departamentų darbuotojai taip pat rado naudingos informacijos.

ShareNet tapo internetiniu pagrindu sukurtų žinių valdymo sistemų pavyzdžiu, kuri saugo bei apdoroja daug vertingų žinių ir jas padaro prieinamas visiems darbuotojams. Tokios kompanijos kaip Intel ar Volkswagen studijavo ShareNet sistemą prieš kurdamos savo nuosavas žinių valdymo sistemas. Teleos, nepriklausoma žinių vadybos kompanija, penkerius metus iš eilės pripažino Siemens geriausia žinių verslumo kompanija.

Taip 1999 m. balandį pradėta ir 7,8 mln. dolerių kainavusi ShareNet sistema per pirmuosius dvejus gyvavimo metus įmonei padėjo uždirbti 122 mln. dolerių pelno (Zakiri 2011).

5. PRAKTINĖ DALIS

5.1. TEAMWORK

2001 m. įkurta Italijos įmonė Open Lab Java5 pagrindu sukūrė patikimą internetinę projektų valdymo sistemą Teamwork. Teamwork yra paprastas naudoti, todėl lengva valdyti daug sudėtingų projektų vienu metu. Open Lab siekė, kad jų kuriamas produktas atitiktų keliamus reikalavimus:

- 1) turėtų tiesioginę prieigą prie duomenų bazės;
- 2) integruotas LDAP;
- 3) palaikytų HTTPS;
- 4) integravimas su Linux failų serveriu;
- 5) būtų protingai įkainotas.

Teamwork 4.5 versijoje palaikomas sistemos multikalbiškumas, leidžiantis darbuotojams pasirinkti jiems patogiausią kalbą iš 26 kalbų sąrašo, asmeninis darbuotojų kalendorius, kuriame atvaizduojama vartotojo darbo dienos informacija (susitikimai, atostogos, ...), bei individualus vartotojo pirmojo puslapio administravimas, t. y. vartotojas pagal savo poreikius šablone gali sudėlioti jam reikalingus VSB.

Teamwork yra patogus projektų ir užduočių valdymui, jų kainų skaičiavimui, prioritetų ir kritiškumų, išteklių užduotims priskyrimui. Taip pat įdiegta galimybė įkelti Microsoft Project failus ir juos atvaizduoti Ganto diagramoje. Numatytas funkcionalumas HD užduočių įvedimui ir darbuotojų darbo laiko apskaitai (*worklog*). Svarbu paminėti, kad sistema palaiko automatinį žinučių siuntimą vartotojams ir užduočių kūrimą elektroniniais laiškais.

Kadangi Teamwork yra sukurtas Java pagrindu ir naudoja reliacinę duomenų bazę, programa gali veikti visur Teamwork yra serverio aplikacija, kuri turi būti instaliuota serveryje. Kreipimasis į aplikaciją vyksta per interneto naršykles Firefox, Chrome, Microsoft Explorer ar Safari. Teamwork veikimui reikalinga reliacinė duomenų bazė, kurioje būtų laikomi duomenys. Beveik visos reliacinės duomenų bazės yra palaikomos: MySQL, SQL Server, Oracle, HsqlDB, Ingres (OpenLab 2010) (Teamwork 2010).

5.2. ŽINIŲ BAZĖS INTEGRAVIMAS Į TEAMWORK

Žinių bazės integravimui į Teamwork sistemą buvo pasirinkta atvejais pagrįsta žinių bazė, kuri susideda iš atvejų rinkinio. Buvo pasirinkta atvejais pagrįsta žinių bazė, nes ji gerai veikia

apibrėžtoje srityje, leidžia pasinaudoti ankstesne sėkme, sprendžiant analogiškas arba panašias situacijas bei panaudoti eksperimentines žinias, išgautas iš žmogaus – eksperto.

Integruota žinių bazė buvo orientuojama į du projekto valdymo procesus:

- 1) planavimą – nes tai sudėtingas daugiatapis procesas, nuo kurio priklauso sėkmingas projekto tikslų įgyvendinimas;
- 2) palaikymą – jo metu paaiškėja projekto metu sukurto produkto defektai ar nuolatinį nekorektišką vartotojų darbą sukurtu produktu.

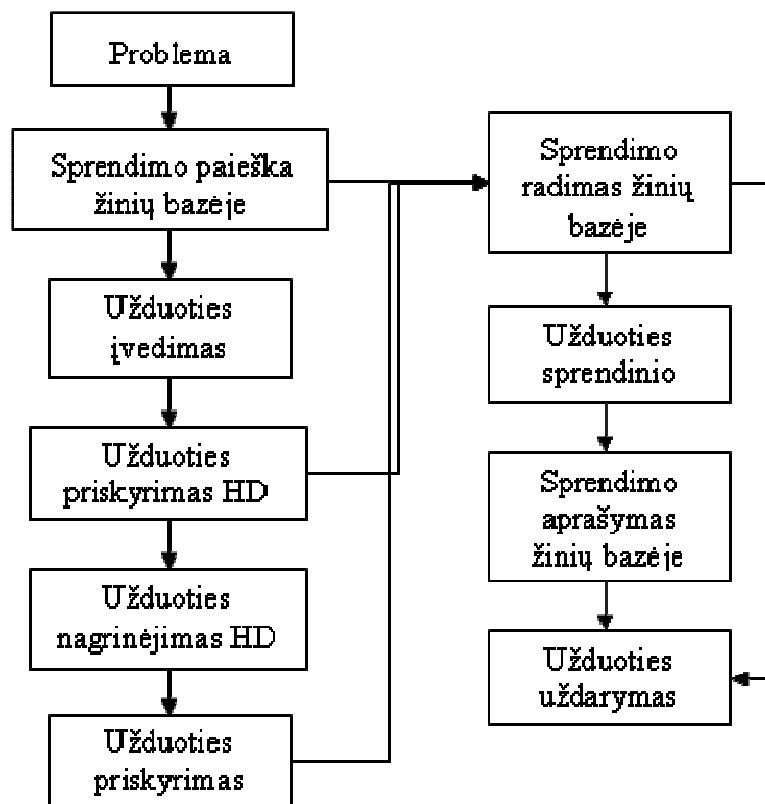
Nuspręsta, kad integruojamoje žinių bazėje bus kaupiama organizacijos vykdytų projektų informacija, kuri padės pigiau ir greičiau spręsti užduotis, teiks pagalbą renkant projekto komandą, nustatant projekto biudžetą ir vykdymo terminus bei bus prieinama darbuotojams, norintiems praplėsti savo žinias arba jomis pasidalinti su kitais.

5.3. NAUJAS UŽDUOČIŲ SPRENDIMO PROCESAS

Teamwork 4.5 esančiame HD lange įmonės darbuotojai registruoja įvairias problemas, kurios vėliau yra priskiriamos atitinkamiems darbuotojams. Dažnu atveju, tokios klaidos yra nuolat pasikartojančios ir išsprendžiamos pačių besikreipiančių darbuotojų lygyje.

Be to, nors tokių užduočių sprendimai ir žinomi, tačiau pasikeitus darbuotojui ar jam nebūnant darbo vietoje (liga, atostogos, komandiruotė), užduočiai priskirtą darbuotoją pavaduojantys asmenys ne visuomet žino kaip išspręsti kilusią problemą. Taigi, net pasitaikius elementariausioms ir paprasčiausioms problemoms ne visuomet pavyksta greitai ir pigiai jas išspręsti bei išvengti kitų susijusių nepatogumų.

HD VSB realizavus galimybę darbuotojui dar prieš įvedant užduotį patikrinti ar tokia problema jau yra pasitaikiusi anksčiau ir – jei tokia problema buvo užregistruota ir išspręsta – pasinaudoti jos sprendimu. Taip išvengiama pasikartojančių užduočių ir tausojamas kitų darbuotojų laikas. Naujasis užduočių įvedimo / sprendimo procese užduotys yra suskaidomos į tris – pirmojo, antrojo ir trečiojo – lygio palaikymus (20 pav.).



20 pav. Užduoties sprendimo procesas

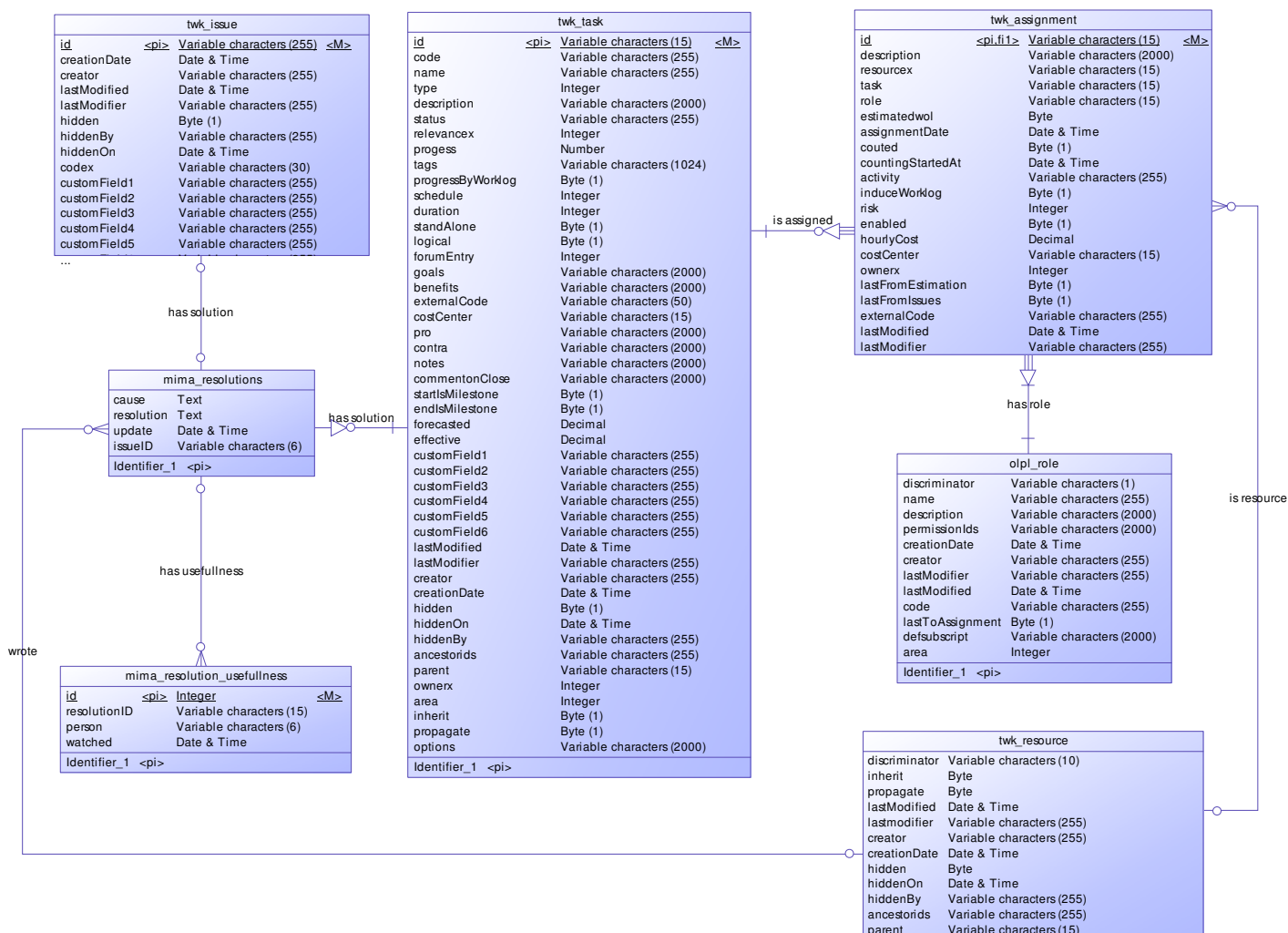
Taigi, kilus problemai darbuotojas žinių bazėje patikrina ar panaši užduotis jau buvo pasitaikiusi praeityje ir koks buvo jos sprendimo būdas. Jei darbuotojas problemos sprendimą rado ir gali išspręsti pats, užduoties neregistruoja ir pagal rasto sprendimo aprašymą sprendžia problemą savarankiškai.

Jei darbuotojas problemos sprendimo nerado arba pagal rastą sprendimo aprašymą negali problemos išspręsti savarankiškai, tuomet HD lange užregistruoja užduotį, nurodydamas užduoties pavadinimą / problematiką, jos aprašymą, atlikimo terminus ir kritiškumą.

Užregistruotą užduotį perima HD darbuotojas. Jis patikrina žinių bazėje ar panaši užduotis buvo pasitaikiusi anksčiau ir koks buvo jos sprendimas. Jei HD darbuotojas sprendimą rado ir / arba gali ją išspręsti pats, tuomet užduotis yra išsprendžiama ir uždaroma. Priešingu atveju, jei užduoties išspręsti negali ir HD darbuotojas, tuomet jis užduotį patikslina ir priskiria kitam darbuotojui. Gavęs naują užduotį, darbuotojas ją išsprendžia ir, įvedęs į sistemą problemos priežastį ir sprendimą, uždaro. Sekantį kartą pasitaikius tokiai problemai jau bus žinomas jos sprendimo būdas ir ji bus išspręsta greičiau ir pigiau.

5.4. PAKEITIMAI TEAMWORK DUOMENŲ BAZĖJE

Tam, kad Teamwork sistemoje būtų įgyvendintas žinių bazės funkcionalumas, duomenų bazės schema (21 pav.) buvo papildyta lentelėmis „Resolutions“, kurioje kaupiamos užduoties priežastys ir sprendimai, bei „Usefulness“, kurioje žymimas užduoties sprendimo populiarumas.



21 pav. Papildyta Teamwork sistemos duomenų bazė

Pagrindinė užduočių informacija yra kaupiama *twk_task* lentelėje. Atkreiptinas dėmesys, kad tėvinis stulpelis yra lentelės išorinis raktas ir rodo į tą pačią užduoties lentelę. Tai leidžia nustatyti užduoties poziciją užduočių medyje.

Twk_issue lentelėje kaupiamos sistemos vartotojų užregistruotos problemos, kurios gali būti priskirtos projektams ar užduotims, arba egzistuoti kaip savarankiški darbai (*todo*). Svarbiausias ryšys, jungiantis užduotį su žmogumi yra užduoties priskyrimas, kuris yra saugomas *twk_assignment* lentelėje. Ši reliacinė lentelė sujungia užduotis su resursais (žmonėmis) ir rolėmis. Prie Teamwork

prisijungę vartotojai saugomi dvejose esybėse: jie yra „resursai“, kuriems priskiriami darbai, ir sistemos vartotojai, kurie galėtų prisijungti. Sistemos vartotojai, kurie gali prisijungti prie sistemos, yra *olpl_operator* lentelės įrašuose, kuris yra ryšys vienas su vienu su įrašu *twk_resources* lentelė. Resursai turi ryšius su beveik visomis Teamwork lentelėmis. Pagrindiniai resursų – sistemos vartotojų – duomenys kaip telefonas, el. paštas ar valandinis darbo įkainis yra kaupiama *olpl_anagraphicaldata* lentelėje ir su *twk_resource* lentelė surišama per *twk_res_ad*.

Užduoties ir problemos priežastys bei sprendimai saugomi *mima_resolutions* lentelėje, kuri ryši turi vienas-su-daug su *mima_resolution_usefulness* lentelė. Šioje lentelėje kaupiama informacija apie problemos sprendimo naudingumą. Kiekvieną kartą įmonės darbuotojui ieškant kilusios problemos sprendimo ir jį radus žinių bazėje, darbuotojas pažymi, kad jis padėjo. Sekantį kartą, kai darbuotojas ieškos užduoties sprendimo žinių bazėje, rezultate pirmiau bus pateikti sprendimai, turintys didesnę populiarumo statusą (Teamwork database Structure 2010).

5.5. PROJEKTO BIUDŽETO VERTINIMAS

Projekto kaina yra vienas iš trijų pagrindinių sėkmingai įvykdyto projekto komponentų. Blogai įvertinta mažo projekto kaina didelės žalos organizacijai nepadarys. Bet vertinant didelius ir labai didelius projektus iškyla didelis pavojus, kad projekto kaina bus netinkamai įvertinta ir gali tekti ieškoti papildomo biudžeto, įgyvendinti projektą mažesniais ištekliais, įvykdyti ne visus tikslus arba išvis atsisakyti jau prasidėjusio projekto įgyvendinimo.

Projekto kaina – tai projekto išlaidų ir laukiamo pelno suma. Tačiau skaičiuojant projekto kainą reikia įvertinti gerokai daugiau veiksnių.

Egzistuoja įvairūs projekto kainos vertinimo metodai:

- 1) algoritminis modelis projekto kainą įvertina remdamasis įvykdytais tyrimais ar istorine informacija;
- 2) projekto kainos vertinimas, remiantis ekspertų rekomendacijomis;
- 3) įvertinimas pagal analogą – lyginama su anksčiau vykdytais projektais, kurių informacija yra žinoma;
- 4) parkinsono principo modelis projekto kainą nustato atsižvelgdamas į projektui skiriamą laiką ir finansinius išteklius;
- 5) laimėjimo kainos principas – kainą lemia suma, pakankama laimėti paskelbtą projekto konkursą. Šis principas yra vienas nepatikimiausių, nes yra orientuojamasi į projekto

konkurso laimėjimą, bet neatsižvelgiama kokia yra tikroji projekto apimtis ir jam reikalingi resursai;

- 6) iš viršaus žemyn metodas projekto komponentų kainas apskaičiuoja atitinkamomis proporcijomis nuo visos projektui skirtos sumos;
- 7) iš apačios į viršų metodas iš pradžių įvertina komponentų kainas ir tuomet jas susumavus gaunama galutinė projekto kaina;
- 8) kiti (Laučius *et al.* 2007).

Tik labai patyrę ekspertai gali įvertinti preliminarą projekto, kurio metu, pavyzdžiui, bus kuriama programinė įranga, kainą. Tačiau ekspertų darbo valandos įkainojamos brangiai, todėl projekto kainos lange buvo realizuotas įvertinimo pagal analogą funkcionalumas, leidžiantis peržiūrėti panašius projektus ir jų biudžetus praeityje. Taip pat pateikiamas šių projektų biudžetų vidurkis (22 pav.).

project	cost	start date	end date
☒ Antras projektas	123.25	2011-05-02 00:00:00.0	2011-05-10 23:59:59.0
☒ Trecias projektas	1760	2011-05-02 00:00:00.0	2011-05-02 23:59:59.0
☐ Ketvirtas projektas	0	2011-05-31 00:00:00.0	2011-06-24 23:59:59.0
☒ Penktas projektas	750	2011-05-11 00:00:00.0	2011-06-17 23:59:59.0

average cost 877.75 average day cost 73.15 **count**

22 pav. Panašių projektų biudžetų langas

Projekto kainos („cost“) lange paspaudžiamas mygtukas „recommend“. Pagal įvestas projekto žymes atrenkami ir atvaizduojami panašūs projektai. Čia sistemos vartotojas gali pažvelgti šių projektų aprašymus ir, pažymėjęs panašiausius arba jį labiausiai dominančius projektus, paspaudžia mygtuką „count“. Sistema apskaičiuoja ir atvaizduoja pasirinktų projektų kainų vidurkį. Taip pagal analogus įvertinama naujo projekto vykdymo kaina.

5.6. PROJEKTO LAIKO VERTINIMAS

Siekiant, kad projektas būtų sėkmingai įgyvendintas, svarbu tinkamai įvertinti projekto pradžios ir pabaigos terminus. Jei projekto įgyvendinimui numatyta per mažai laiko arba įvertinta pernelyg optimistiškai (t. y. nenumatant tokių atvejų, kad atsiras kitų skubių darbų, darbuotojai gali susirgti, pakeisti darbą), tuomet tenka atsisakyti dalies projekto apimties arba ieškoti papildomų išteklių. Tačiau jei projekto vykdymo laikotarpis bus įvertintas pernelyg pesimistiškai, projekto užsakovai gali išvis atsisakyti projekto įgyvendinimo.

Analogiškai projekto kainos vertinimui, projekto vykdymo laiko vertinimui pasirinkau projekto laiko įvertinimą, remiantis anksčiau vykdytais projektais. Tuo tikslu pagrindiniame projekto lange buvo realizuota galimybė peržiūrėti panašius projektus praeityje ir jų įgyvendinimo laikotarpius (23 pav.).

project	start date	end date	duration
☒ Antras projektas	2011-05-02 00:00:00.0	2011-05-10 23:59:59.0	7
☒ Trecias projektas	2011-05-02 00:00:00.0	2011-05-02 23:59:59.0	1
☐ Ketvirtas projektas	2011-05-31 00:00:00.0	2011-06-24 23:59:59.0	19
☒ Penktas projektas	2011-05-11 00:00:00.0	2011-06-17 23:59:59.0	28

average duration: 12 count

23 pav. Panašių projektų laiko įvertinimo langas

Pagrindiniame projekto lange, paspaudus mygtuką „*recommend*“, iššoka langas su projektais, turinčiais panašias žymes į vertinamo projekto žymes. Vartotojas, susipažinęs su projektais ir pasirinkęs jam labiausiai tinkančius, paspaudžia mygtuką „*count*“, kuris apskaičiuoja pasirinktų projektų vykdymo laikotarpių vidurkį.

5.7. DARBUOTOJŲ REKOMENDAVIMAS

Vienas iš pagrindinių sėkmingo projekto veiksmų yra tinkamą kompetenciją turinčių darbuotojų atrinkimas ir priskyrimas bei tinkamos socialinės aplinkos sukūrimas. Rekomenduojama į projektą įtraukti asmenis, kurie numatytam darbui gali skirti ne mažiau kaip 50 proc. darbo laiko. Tai leidžia išvengti laiko nuostolių kai darbuotojas bando persiorientuoti nuo vieno projekto prie kito, įsigilinti į naujas užduotis ir t. t.

Surinkus projekto komandą, kuri neturi darbo patirties panašiuose projektuose, kyla didelė rizika, kad komanda nesugebės laiku ir kokybiškai atlikti jiems priskirtų projektinių užduočių – bus viršytas projekto laikas ir (arba) netilps į projekto biudžetą. Kita vertus, į projekto komandą atrinkus geriausius specialistus, projekto kaina smarkiai išauga.

Rekomenduojama atrinkti tokią projekto komandą, kurios daugiau nei 40 % narių sudaro darbuotojai, turintys didelę patirtį panašiuose projektuose. Projekto komandos dažniausiai nėra sudaromos tik iš aukščiausią kvalifikaciją ir didžiausią patirtį turinčių darbuotojų, nes dažnai, tai brangiausiai apmokami įmonės darbuotojai, todėl projekto kaina smarkiai išauga. Tačiau jų dalyvavimas projekte būtinas, nes taip perteikia kitiems komandos nariams sukauptas žinias ir kompetenciją.

Apie 50 % projekto komandos gali sudaryti darbuotojai, kurie dalyvavo tik keliuose arba išvis nedalyvavo panašiuose projektuose, tačiau turi reikalingą kvalifikaciją ir gebėjimus. Tai gali būti darbuotojai, kurie įmonėje dirba trumpą laiką ir nespėjo sukaupti darbo patirties įmonės projektuose ar turi daug patirties kituose projektuose, tačiau nedalyvavo vykdant analogiškus projektus.

Galiausiai iki 10 % komandos darbuotojų – panašiuose projektuose yra nedalyvavę ir (arba) atitinkantys ne visus reikalingus kvalifikacijos komponentus darbuotojai. Jei į projekto komandą bus įtraukta daugiau nepatyrusių darbuotojų, kyla grėsmė, kad projektas gali būti neįgyvendintas laiku ir (arba) bus susidurta su projekto užduočių vykdymo problemomis, kilusiomis dėl darbuotojų kvalifikacijos stokos. Kita vertus, svarbu į komandą įtraukti ir nelabai patyrusius darbuotojus, kad jie galėtų įgauti naujos patirties ir asmeninių gebėjimų.

Sėkmingam projekto komandos subūrimui projekto darbuotojų priskyrimo lange („assignment“ → „add assignment“) realizuota galimybė rasti darbuotojus, turinčius darbo patirtį panašiuose projektuose, jų roles bei panašius projektus (24 pav.).

employee	role	project	hourly cost
System Manager	Project manager	Antras projektas	135
System Manager	Project manager	Trecias projektas	135
System Manager	Stakeholder/Customer	Ketvirtas projektas	135
Vardenis Pavardenis	Stakeholder/Customer	Antras projektas	115
Vardenis Pavardenis	Worker	Ketvirtas projektas	115
Vardenis Pavardenis	Worker	Penktas projektas	115
Vardenis Pavardenis	Worker	Trecias projektas	115

24 pav. Panašiuose projektuose dirbusių įmonės darbuotojų langas

Projekto darbų priskyrimo lange, paspaudus mygtuką „*recommend*“, atsidaro langas, kuriame atvaizduojami visi darbuotojai, dalyvavę panašiuose projektuose. Rekomenduojamų darbuotojų sąrašas rikiuojamas pagal jų patirtį projektuose, pirmiausia pateikiant didžiausią patirtį turinčius darbuotojus. Sistemos vartotojui suteikiama galimybė susipažinti tiek su rekomenduojamo darbuotojo duomenimis, tiek su to darbuotojo vykdytų projektų informacija.

5.8. ŽINIŲ BAZĖS PANAUDOJIMAS UŽDUOČIŲ SPRENDIMUI

Atlikus reikalingus pakeitimus pagal aukščiau nurodytą naują užduočių įvedimo / sprendimo procesą, gautas lūkesčius pateisinęs rezultatas. Vartotojas, Teamwork sistemoje pasirinkęs HD langą, mato 25 paveiksle pateiktą užduoties įvedimo langą.

“*Add service request*” VSB darbuotojas, prieš įvesdamas užduotį HD, paieškos laukelyje įvedęs raktinius žodžius, patikrina nėra sistemoje panašių užduočių su galimais sprendimais. Sistema ieško aktyvių ir jau uždarytų užduočių, kurių sprendimo („*resolution*“) laukas duomenų bazėje yra ne tuščias ir turinčių panašumų užduoties pavadinime (*description*) arba priežasties aprašyme (*cause*) su čia įvesto pavadinimo tekstu.

The screenshot shows the Teamwork software interface. On the left is the 'add issue / todo' form with fields for 'kil' (search), 'task*', 'status' (set to 'open'), 'description*' (a large text area), 'tags', 'severity' (set to 'block'), and 'requester'. A 'save' button is at the bottom right of the form. On the right is the 'my inserted issues' table, which lists four issues with columns for severity, status, description, task, requester, assignee, and done by. Below the table is the 'top resolutions' section, showing a list of user requests and solutions with vote counts.

sev.	s.	description	task	requester	assignee	done by
Yellow		Ketvirta kilusi problema		System Manager	System Manager	
Red		Trecia kilusi problema		System Manager	System Manager	
Orange		Antra kilusi problema		System Manager	System Manager	
Red		Pirma kilusi problema		System Manager	System Manager	01/04/2011

votes	request
10	Pirma kilusi problema
8	Pirmas projektas
1	Trecias projektas
1	Antra kilusi problema
0	Trecia kilusi problema

25 pav. HD langas

Surastus rezultatus sistema pateikia žemiau sąrašo pavidalu („*Relevant tasks and resolutions*“). Paspaudęs ant bet kurios užduoties, vartotojui atvaizduojamas langas su tos užduoties priežastimi ir sprendimu. Teamwork prie kiekvienos užduoties yra papildomai sukurta galimybė išsaugoti užduoties sprendimo informaciją (27 pav.), tad užduoties priežasties ir išsprendimo lange vartotojui atvaizduojama užduoties aprašymo (*description*) ir jos sprendimo (*resolution*) informacija kartu su kitais parametrais (26 pav.).

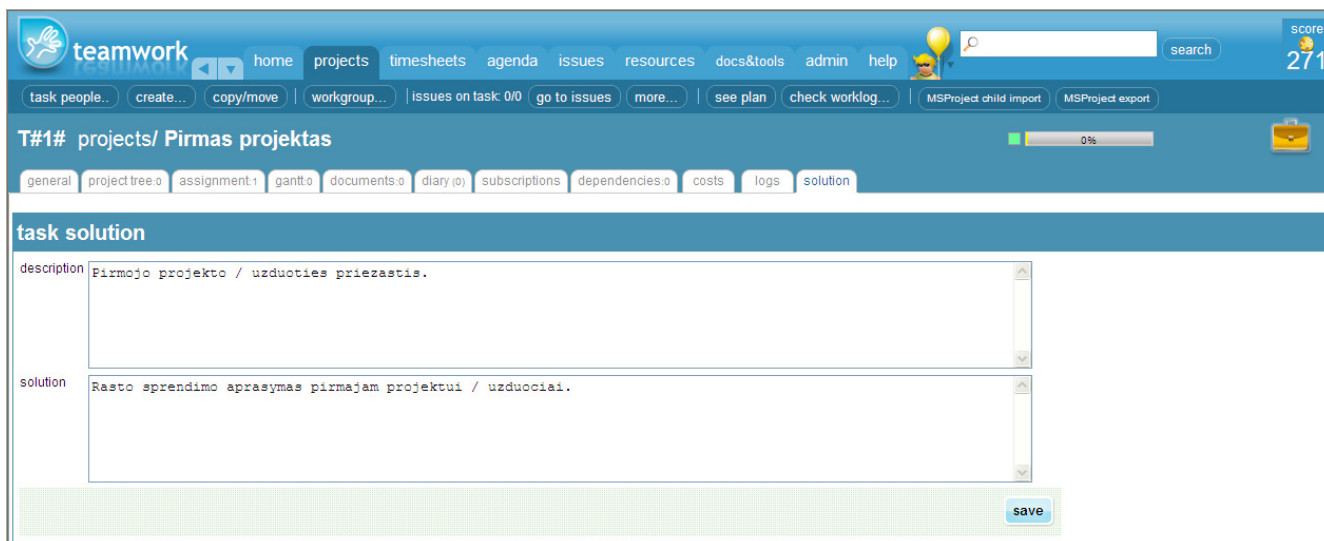
The screenshot shows the 'task: Pirmas projektas' form. It has two main text areas: 'Cause' and 'Resolution'. The 'Cause' field contains the text 'Pirmojo projekto / uzduoties priezastis.'. The 'Resolution' field contains the text 'Rasto sprendimo aprasymas pirmam projektui / uzduociai.'. Below these fields are fields for 'assignee' (System Manager), 'requester' (System Manager), 'date closed by' (2011-04-25), and 'opened' (2011-04-25). A 'full task information' link is at the bottom, and a 'Yes, it helped!' button is at the bottom right.

26 pav. Užduoties priežasties ir išsprendimo langas

Čia taip pat yra numatyta galimybė vartotojams “balsuoti” (paspaudžiant mygtuką „Yes, it helped!“) už konkrečius sprendimus ir tokiu būdu juos reitinguoti pagal populiarumą, kurie atitinkamai turės būti atvaizduojami „Top resolutions“ VSB.

HD darbuotojas mato visas jam priskirtas užduotis ir išnagrinėja naujai atsiradusios užduoties turinį. Antrojo lygio palaikyme HD darbuotojas priskiria užduoties tipą ir patikslina prie užduoties nurodytą informaciją. Tuomet bando išspręsti užduotį pats arba naudodamasis žinių bazėje esančia informacija. Teigiamu atveju jis „resolution“ lauke užfiksuoja sprendimą. Sulaukęs užsakovo patvirtinimo, kad užduotis išspręsta, HD darbuotojas užduotį uždaro. Neigiamu atveju, t. y. negalėdamas pats išspręsti užduoties, vykdomas trečiojo lygio palaikymas. Jo metu, HD darbuotojas perduoda užduoties vykdymą už sritį atsakingam darbuotojui. Atsakingo žmogaus priskyrimas atliekamas rankiniu (prie užduoties „Assignee“ sąraše priskirdamas organizacijos darbuotoją). Tokiu atveju HD darbuotojas pašalina save iš šios užduočiai priskirtųjų sąrašo.

Atsakingas IT darbuotojas perėmęs užduoties vykdymą analizuoja užduoties turinį, patikslina užduoties tipą ir, bendraudamas su užsakovu, bando ją išspręsti. Išsprendęs užduotį, į sistemą įveda užduoties priežastį ir sprendimą (27 pav.), užduotis uždaroma.

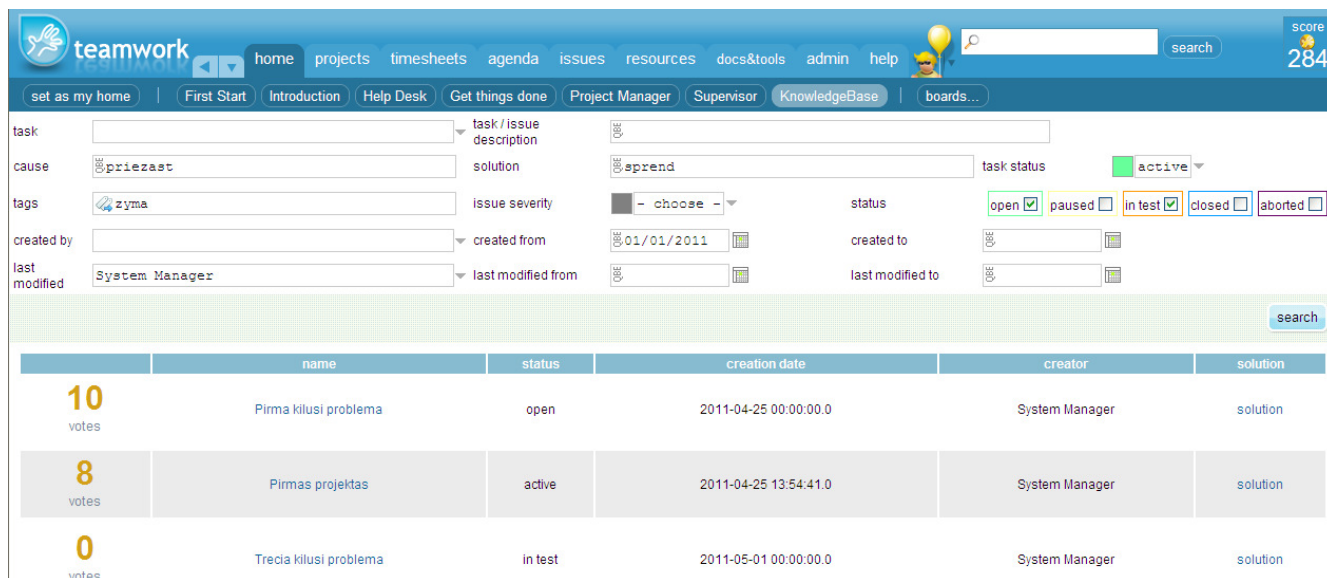
The screenshot shows the Teamwork web application interface. At the top is a navigation bar with tabs for 'home', 'projects', 'timesheets', 'agenda', 'issues', 'resources', 'docs&tools', 'admin', and 'help'. Below this is a sub-navigation bar with buttons like 'task people...', 'create...', 'copy/move', 'workgroup...', 'issues on task: 0/0', 'go to issues', 'more...', 'see plan', 'check worklog...', 'MSProject child import', and 'MSProject export'. The main content area is titled 'T#1# projects/ Pirmas projektas' and includes a progress bar at 0%. Below the title are tabs for 'general', 'project tree', 'assignment', 'gantt', 'documents', 'diary', 'subscriptions', 'dependencies', 'costs', 'logs', and 'solution'. The 'solution' tab is active, showing a 'task solution' form. The form has two text input fields: 'description' with the text 'Pirmojo projekto / uzduoties priezastis.' and 'solution' with the text 'Rasto sprendimo aprasymas pirmajam projektui / uzduociai.'. A 'save' button is located at the bottom right of the form.

27 pav. Užduoties priežasties ir sprendimo įvedimo langas

Kitą kartą kilus panašiai problemai bus galima pasinaudoti sukauptomis panašių problemų sprendimų žiniomis ir ją išspręsti žemesniame lygyje arba sutrumpinti problemos sprendimo laiką.

5.9. PAIEŠKA ŽINIŲ BAZĖJE

Vartotojų patogumui, buvo sukurtas paieškos žinių bazėje puslapis, kuriame pagal nurodytas filtro reikšmes ieškomi ir atvaizduojami žinių bazėje sukaupiti sprendimai (28 pav.).



	name	status	creation date	creator	solution
10 votes	Pirma kilusi problema	open	2011-04-25 00:00:00.0	System Manager	solution
8 votes	Pirmas projektas	active	2011-04-25 13:54:41.0	System Manager	solution
0 votes	Trecia kilusi problema	in test	2011-05-01 00:00:00.0	System Manager	solution

28 pav. Paieška žinių bazėje

Sistemos vartotojui suteikiama galimybė filtruoti praeityje pasitaikiusių projektų, užduočių ir problemų sprendimus pagal jų pavadinimus, užduočių ar sprendimų priežastis, sprendimus ir jų aprašymus, reikšminius žodžius, užduoties / problemos svarbumą, statusą, kūrėją, paskutinį sprendimą modifikavusį asmenį ar sprendimo sukūrimo ir paskutinio redagavimo laikotarpius.

Paieška vykdoma tik tarp tų projektų, užduočių ir problemų, kurios turi užpildytus priežasties ir sprendimo laukus. Taigi, vartotojui pateikiami geriausiai tinkantys sprendimai, o ne projektai ar užduotys, kurie neturi praktinės vertės sprendžiant kilusią situaciją. Vartotojo paieškos darbo palengvinimui, filtravimo laukuose naudojami protingi iškrentantys sąrašai (angl. *smart combo box*), iš kurių vartotojas gali pasirinkti ieškomą parametro reikšmę arba ją įvesti. Prie užduoties ir problemos sprendimo statusų laukų pateikiamos ir jų spalvinės vertės, kurios padeda vartotojui greičiau pasirinkti norimą paieškos vertę.

Rezultatai pateikiami užduočių sprendimų populiarumo mažėjimo tvarka. Vartotojui suteikiama galimybė tiek susipažinti su užduotimis, tiek ir su užduočių priežastimis bei sprendimais.

5.10. POPULIARIAUSI PROBLEMŲ SPRENDIMAI

Kad ieškomo problemos sprendimo būtų paprasčiau ieškoti, buvo sukurtas „Top resolutions“ VSB (29 pav.), kuriame atvaizduojami populiariausi problemų sprendimai.



29 pav. „Top resolutions“ VSB

VSB pateikiami 10 populiariausių sprendimų, išrikiuotų jų populiarumo mažėjimo tvarka. Darbuotojai gali greitai surasti reikalingą sprendimą arba turėdami šiek tiek laisvo laiko gali su jais susipažinti ir pasitaikius panašiai problemai ateityje jau žinoti galimą jos sprendimą. Toks paprastas darbuotojų žinių įgijimo būdas leidžia išvengti užduočių registravimo proceso, aprašyto 4.2 skyriuje, ir sutaupyti išteklius, nes sprendimas darbuotojui jau bus žinomas.

Paspaudus ant sprendimo nuorodos, atidaromas problemos priežasties ir sprendimo langas (27 pav.). Taip pat pateikiama greitoji nuoroda į paieškos žinių bazėje langą (28 pav.), tad reikiamo sprendimo neradęs populiariausių sprendimų lange vartotojas gali greitai patekti daugiau žinių galinčią suteikti sistemos dalį.

6. IŠVADOS IR PASIŪLYMAI

1. Atlikta projektų valdymo ir susijusių procesų organizacijoje analizė. Nustatyta, kad didžiausios projektų valdymo rizikos yra susijusios su planuotų terminų nesilaikymu, biudžeto viršijimu ir prasta rezultatų kokybe. Pasiūlyta į projekto komandą įtraukti ne mažiau 40 % panašios patirties praeityje turinčius ir ne mažiau kaip 50 % savo darbo laiko galinčius skirti darbuotojus.
2. Pastebėta, kad tinkamai pasirinkta projekto valdymo sistema padeda užtikrinti sėkmingą projekto įgyvendinimą, gali palengvinti tiek projekto vadovo, tiek ir visos projekto komandos darbą.
3. Atkreiptas dėmesys, kad efektyvus organizacijos žinių kaupimas bei platinimas kelia darbuotojų ir kartu visos organizacijos kompetencijos lygį.
4. Atlikta žinių bazių projektavimo ir integravimo analizė parodė, kad žinių išgavimas pradinėje žinių bazės integravimo stadijoje padeda lengviau suvokti laukiamą galutinį sistemos rezultatą, pasirinkti tinkamiausias žinių vaizdavimo metodus bei priemones. Pastebėta, kad prototipų kūrimas yra greitas ir pigus būdas leisti vartotojui įvertinti būsimos sistemos patogumą ir naudingumą.
5. Pastebėta, kad žiniomis pagrįstos sistemos naudingumas priklauso nuo jos žinių bazėje sukauptų žinių svarumo ir kiekio. Panaudojus žinių bazėje sukauptas žinias apie anksčiau (sėkmingai) įvykdytus projektus, galima sulaukti didesnės sėkmės vykdant naujus projektus bei efektyviau valdyti projektų valdymo procesus.
6. Pasirinkta atvejais paremta žinių bazė žinių bazės integravimui, kuri leidžia pasinaudoti ankstesne sėkme ir įsisavinti organizacijos žinias, sukauptas praktiškai sprendžiant panašias situacijas. Žinių bazės integravimui pasirinkta Teamwork projektų valdymo sistema. Atlikta žinių bazės integracija į Teamwork projektų valdymo sistemą, išlaikant esamos sistemos logiką. Sukurtas Teamwork programinis modulis, kuris naudoja projektų valdymo sistemos žinių bazėje sukauptas žinias ir palengvina sistemos vartotojų darbą vertinant projekto laiką ir kainą, sudarant projekto komandą bei sprendžiant kilusias problemas. Pasiūlytas naujas užduočių sprendimo procesas supaprastina užduočių administravimą ir leidžia jas išspręsti žemesniuose lygiuose.

LITERATŪROS SĄRAŠAS

- Kirvaitis, A. 2003. Žinių valdymo programos: tikrovė ir mitai greta. NK verslas, 03(8) [interaktyvus] [žiūrėta 2011.05.01]. Prieiga per internetą:
<<http://www.verslobanga.lt/lt/leidinys.full/3e490d9cd72cf>>
- Laučius, J. Žinių inžinerija ir žinių valdymas. Vilniaus Gedimino technikos universitetas, magistrantūros studijos. 2009
- Laučius, J.; Vasilecas, O. 2007. Informacinių technologijų projektų ir kokybės valdymas. Vilnius: Technika. 223 p.
- Neverauskas, B.; Stankevičius V.; Viliūnas V.; Černiūtė, I. 2007. Projektų valdymas. Kaunas: Technologija
- Tamulynas, B. 2008. Intelektualiųjų sistemų modeliavimas. Kaunas: Technologija. 275 p.
- Vasiliauskienė L. Programų sistemų inžinerija. Vilniaus Gedimino technikos universitetas, bakalauro studijos [interaktyvus] [žiūrėta 2011.05.01]. Prieiga per internetą:
<http://www.techmat.vgtu.lt/konspektai/PSI/psi_05_pi_projektai.pdf>
- Akerkar, R.; Sajja P. 2009. Knowledge-Based Systems. United States of America: Jones and Bartlett Publishers. 351 p.
- Feng Office [interaktyvus] [žiūrėta 2011.01.13]. Prieiga per internetą:
<<http://fengoffice.com/web/index.php>>
- Knowledge Base Systems [interaktyvus] [žiūrėta 2010.12.17]. Prieiga per internetą:
<<http://linas.org/linux/pm.html>>
- Krishna, S. 1992. Introduction to database and knowledge-base systems. Singapore: World Scientific Publishing Co. 328 p.
- Leondes, C. T. 2005. Knowledge-based systems: Implementation. United States of America: Kluwer Academic Publishers. 351 p.
- Levesque, H. J.; Lakemeyer, G. 2000. The logic of knowledge bases. Massachusetts Institute of Technology. 282 p.
- Matsushita, S.; Itoh, M.; Kawanobe, M. 1997.07.23. Knowledge base system. European Patent Office No EP 0 858 028 A1
- Microsoft Project [interaktyvus] [žiūrėta 2010.12.17]. Prieiga per internetą:
<www.microsoft.com/project>

- Nikolopoulos, C. 1997. Expert systems: introduction to first and second generation and hybrid knowledge based systems. Marcel Dekker. 331 p.
- OpenLab [interaktyvus] [žiūrėta 2010.12.07]. Prieiga per internetą: <http://www.openlab.com/index_en.html>
- Probst, G.; Raub, S.; Romhardt, K. 2006. Žinių vadyba. Sėkmės komponentai. Vilnius: Knygiai. 350 p.
- Redmine [interaktyvus] [žiūrėta 2010.12.17]. Prieiga per internetą: <www.redmine.com>
- Setzer, V. W. Data, Information, Knowledge and Competency [interaktyvus] [žiūrėta 2011.04.01]. Prieiga per internetą: <<http://www.ime.usp.br/~vwsetzer/data-info.html>>
- StuffedGuys Factory Nova [interaktyvus] [žiūrėta 2011.04.18]. Prieiga per internetą: <<http://www.stuffedguys.com/products/factory/>>
- Swartz, M., R.; Winkler, L. J., Janos, A. E.; Markidan, I.; Qun, D. 2001.05.22. Methods and apparatus for the integration of information and knowledge. United States Patent No US 6,236,994 B1
- Teamwork [interaktyvus] [žiūrėta 2010.12.07]. Prieiga per internetą: <<http://www.twproject.com/>>
- Teamwork database Structure [žiūrėta 2010.12.08]. Prieiga per internetą: <<http://twproject.files.wordpress.com/2010/07/teamworkrelationalstructure1.pdf>>
- Tzafestas, S. G. 1997. Knowledge based systems. Singapore: World Scientific Publishing Co. 618 p.
- Walker, G. 2001. IT problem management. United States of America: Prentice-Hall Inc. 234 p.
- Zakiri, A. Azportal Siemens knows what it knows through knowledge management [interaktyvus] [žiūrėta 2011.04.27]. Prieiga per internetą: <<http://azportal.uum.edu.my/portal/courses/tid3143/Assignment/siemens.pdf>>

PRIEDAI

1. Straipsnis „Žinių bazės integravimas į projektų valdymo sistemą“.

ŽINIŲ BAZĖS INTEGRAVIMAS Į PROJEKTŲ VALDYMO SISTEMĄ

Asta Aleknaitė

Vilniaus Gedimino technikos universitetas

El. paštas: asta.aleknaite@gmail.com

Santrauka. Nagrinėjama žinių svarba organizacijos veikloje. Apžvelgiamos žinių bazių ir žiniomis pagrįstų sistemų sandaros ir pagrindinės savybės, naudojami žinių bazių tipai. Aptariama kokiomis duomenų struktūromis aprašomos žinios. Pateikiamas žinių bazių integravimo metodas.

Reikšminiai žodžiai: žinios, žinių bazė, žiniomis pagrįsta sistema, taisyklėmis paremta sistema, atvejais paremta sistema, semantinis tinklas, freimas.

Įvadas

Įmonių, užsiimančių intelektinių produktų kūrimu, vadovai daugiau nei prieš dešimtmetį atkreipė dėmesį, kad brangiausias įmonės turtas yra darbuotojas ir jo sukauptos žinios. Iškylant naujiems iššūkiams, vadovai vis daugiau pastangų skiria darbui su žiniomis kaip su valdomu ištekliumi.

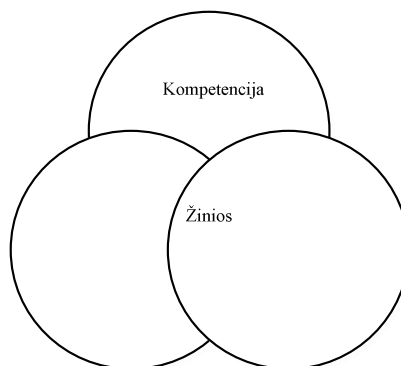
Žinios apibrėžiamos kaip trijų sričių – duomenų, informacijos ir kompetencijos – sankirta (1 pav.).

Duomenys – tai eilė išmatuojamų arba apibrėžiamų simbolių, tokie kaip skaičiai, raidės, paveikslėliai, signalai ir kt.

Tuo tarpu informacija yra neformali abstrakcija, kuri yra žmogaus galvoje minčių pavidalu ir turi tam žmogui reikšmingą prasmę.

Kompetencija yra žmogaus gebėjimas atlikti užduotį realioje aplinkoje, t.y. žmogus turi įgūdžių, kurie buvo įgauti ir panaudoti anksčiau atliktuose darbuose (Setzer 2011).

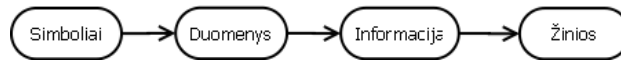
Žinios – tai informacija, paremta išvalgomis ir patirtimi, kuria remiantis priimami sprendimai ir veiksmai (Levesque 2000). Žinios yra pažinimo ir įgūdžių, kuriuos asmenys vartoja sprenddami problemas, visuma, ir vienintelis išteklius, kuris kapiasi jį naudojant. Jos apima tiek teoriją, tiek praktines, kasdienes taisykles bei instrukcijas. Žinios yra pagrįstos duomenimis ir informacija, tačiau skirtingai nuo pastarųjų, jos visuomet yra susijusios su asmenimis ir jų kompetencija.



1 pav. Žinių sąryšis su duomenimis, informacija ir patirtimi / kompetencija

Fig. 1. Knowledge relationship between data, information and experience / competence

Dažniausiai apibrėžiamas žinių išgavimo modelis atvaizduotas 2 pav.



2 pav. Žinių išgavimo modelis

Fig. 2. Knowledge extraction model

Duomenis sudarantys simboliai yra apdorojami juos filtruojant, formatuojant, atliekant aritmetines operacijas ir pan. Gautą informaciją žmogus vertina, interpretuoja ir priima sprendimą.

Organizacijos žinių bazę sudaro individualus ir kolektyvinis žinių turtas, kurį organizacija gali naudoti vykdydama užduotis. Žinių bazė apima ir duomenis bei informaciją, kuriomis remiantis kuriamos individualios ir organizacinės žinios. Taigi, žinios yra įmonės kapitalas, kurį privalu saugoti, plėsti ir dalinti įmonės darbuotojams.

Inovacijas kuriančios įmonės suvokia, kad produktų, kurių palyginti paprastos bazinės funkcijos, vertė gali išaugti, jei jie bus susiję su intensyviu žinių naudojimu. Pavyzdžiui, produktui gali būti suteikta galimybė pačiam prisitaikyti prie kintančių sąlygų arba rinkti ir saugoti informaciją bei naudoti ją vartotojo labui.

Straipsnyje bus apžvelgtos žiniomis pagrįstos sistemos ir žinių bazės, jų savybės bei pasiūlytas žinių integravimo į projekto valdymo sistemą modelis.

Žiniomis pagrįsta sistema

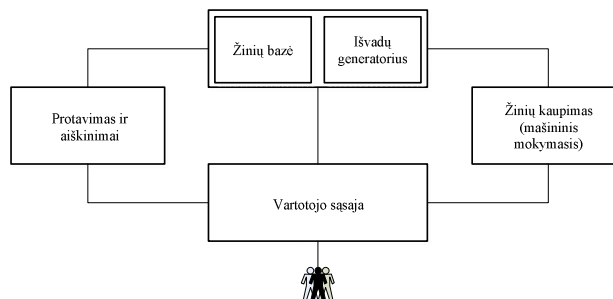
Visuomenė ir industrija vis labiau orientuojasi į žinias ir remiasi skirtingais ekspertų sprendimais, kad išspręstų kilusias problemas.

Kad kompiuterinė programa veiktų kaip ekspertas tam tikroje srityje, ji turi ne taikyti žinomus algoritmus, o naudotis sukauptais faktais ir taisyklėmis. Žiniomis pagrįsta sistema simuliuoja sprendimo priėmimą ir problemos sprendimo galimybes, kaip žmogus – ekspertas savo kompetencijos ribose; ir yra tikimasi, kad ji veiks tokia pačia lygyje kaip ir žmogus – ekspertas.

Tokia žiniomis pagrįsta sistema yra veiksmingas įrankis, kuris gali veikti kaip ekspertas bet kur ir bet kada, taupo pinigus, leidžia vartotojams aukšto lygio funkcionalumus ir užtikrina neprieštarumą.

Žiniomis pagrįstos sistemos sandara

Žiniomis pagrįstą sistemą sudaro keturios pagrindinės dalys: žinių kaupimo modelis, žinių bazė, vartotojo sąsaja ir išvadų generatorius. 3 pav. pateikiama žiniomis pagrįstos sistemos visuotina bendra architektūra:



3 pav. Žiniomis pagrįstos sistemos architektūra

Fig. 3. Knowledge-based system architecture

Žinių bazė gali būti naudojama kaip įvairių formų žinių talpykla. Tai gali apimti tuščią vietą, kurioje bus saugomi laikini rezultatai ir informacijos ar žinių dalys. Išvadų generatoriuje esanti programinė įranga, daro išvadas iš žinių, esančių žinių bazėje ir ieško optimaliausio problemos sprendimo.

Žiniomis pagrįstos sistemos patikimumas priklauso nuo gebėjimo paaiškinti priimtą sprendimą ir veiksmų, lėmusių priimti vienokius ar kitokius sprendimus.

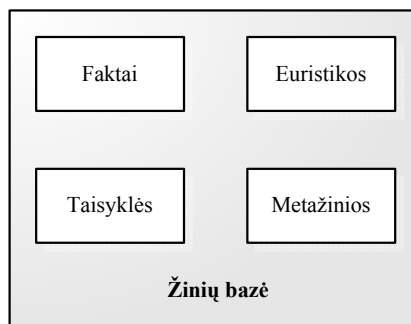
Žmonės gali išmokyti naujų dalykų, tačiau pamiršta žinias, kurias naudoja nereguliariai. Toks mokymasis yra esminis žiniomis pagrįstos sistemos komponentas. Žiniomis pagrįsta sistema gali būti nuolat papildoma arba automatiškai atnaujinama pačios mašinos (mašininis mokymasis). Idealiu atveju, žiniomis pagrįstos sistemos pagrindą retai kada reikia modifikuoti (Krishna 1992).

Taip pat žiniomis pagrįsta sistema privalo turėti priimtina vartotojo sąsają, kuri gali turėti natūralios kalbos apdorojimo galimybę ir leidžia vartotojo ir sistemos dialogą, pateikiant klausimus ir gaunant atsakymus.

Žinių bazės sandara

Žinių bazė yra pagrindinė žinių bazėmis paremtos sistemos dalis. Sistemos kokybė ir naudingumas tiesiogiai priklauso nuo joje sukauptų žinių.

Žinių bazė susideda iš žinių, pateiktų įvairiomis formomis. Akivaizdu, kad žinių bazė turi būti sudaryta iš srities žinių, kuriomis remiantis sistema sprendžia problemas. Žinių bazėje taip pat turėtų būti laikomos ir metažinios. Bendrąja prasme, jiems priklauso ontologijos – struktūra, kurioje kaupiamos srities žinios kartu su jų naudojimo metodais. Paveikslėlyje pateikiami žinių bazės sudedamieji komponentai:



4 pav. Žinių bazės sandara

Fig. 4. Knowledge Base Structure

Faktai aprašo rinkinį stebėjimų, alfabeto, simbolių ar teiginių, kuriais remiantis bus generuojamas uždavinio sprendimas.

Taisyklės aprašo sąlygas ir veiksmus, dar žinomus kaip priežastys ir pasėkmės. Žiniomis paremta sistema susidedanti tik iš taisyklių vadinama taisyklėmis paremta sistema. Kad būtų išvengta neapibrėžtumo, taisyklės įtraukia netikrumo faktorius.

Be faktų ir taisyklių, į žinių bazes įtraukiamos ir euristikos, kurios pagerina žinių bazių efektyvumą. Euristika – tai būdas atvaizduoti problemos sprendimo patirtį žinių bazės viduje. Čia kaupiami sprendimai, kuriuos priimtų ekspertas analogiškoje situacijoje. Jei taisyklės ir faktus lengva išgauti ir užrašyti, euristikos yra kaupiamos ekspertų galvose neišreikštų žinių pavidalu, todėl jas sunku suformuluoti, aprašyti ir užrašyti.

Metažinios yra naudingos ieškant reikalingų žinių ir taikant jas problemos sprendimui. Jos naudojamos ontologijose ir naudingos sprendžiant problemas, lyginant problemų charakteristikas ir situacijas, kuriose metažinios gali

būti pritaikytos, taigi palengvina operacijų darbą. Metažinių pavyzdžiai: planavimas, modeliavimas, žymėjimas, mokymasis ir srities žinių redagavimas (Tzafestas 1997).

Žinių bazė

Žinių bazė susideda iš specifinių srities ir kontrolės žinių, kurios yra skirtos padėti išspręsti kilusias problemas srityje. Šios žinios gali būti išgautos iš ekspertų arba jas gali įgyti pati sistema besimokydama. Per žinių įgijimo procesą, žinių inžinierius (sistemos kūrėjas) sukaupia problemos sprendimui reikalingas žinias iš srities ekspertų ir kitų šaltinių, kaip knygos, internetas, ir t. t. Automatizuotame žinių įgijime sistema naudoja mašininio mokymosi techniką, kad galėtų tiesiogiai įgyti žinias iš gautų duomenų eilutės.

Konceptualus įgytų žinių modelis yra konvertuojamas į kompiuteriui valdyti patogesnę formą. Toks procesas vadinamas žinių vaizdavimu. Žinios žinių bazėje gali būti vaizduojamos ir laikomos įvairiomis formomis.

Duomenų struktūra

Vienas dažniausiai naudojamų būdų atvaizduoti žinioms yra taisyklių forma. Taisyklė turi formą:

IF (sąlyga yra patenkinama)

THEN (veiksmas ar išvedimas prie naujų išvadų).

Taisyklėms keliama reikalavimai, kad sistema turi veikti net kai nėra kai kurių pradinių duomenų ir turi būti numatyta taisyklių vykdymo eiga.

Semantinis tinklas yra grafinis žinių vaizdavimo būdas mazgais ir ryšiais tarp jų. Pirmą kartą semantiniai tinklai buvo panaudoti dirbtinio intelekto sistemose, skirtose kompiuteriniam vertimui. Pagrindinis semantinių tinklų bruožas – deklaratyviai atspindimos žinios. Semantiniais tinklais vaizduojamos žinios labai gerai atvaizduoja ilgalaikę atminties struktūrą.

1962 m. K. A. Petri aprašė naują informacinių srautų sistemos modelį. Kuriant modelį, buvo laikytasi prielaidos, kad atskiros sistemos dalys funkcionuoja asinchroniškai ir konkuruoja tarpusavyje. Ryšiai tarp šių atskirų dalių buvo vaizduojami grafu arba tinklu. Dažniausiai Petri tinklai naudojami modeliuojant sistemas, kuriose vienu metu gali įvykti keletas įvykių ir yra apribojimai procesų dažnumui, sekai. Įvykiams yra suteikiami prioriteto įvertinimai. Tai labai patogus būdas informaciniams procesams aprašyti, kuomet valdymo sistemose vyksta konfliktinės situacijos, lygiagretūs atsitiktiniai ir nedeterminuoti procesai.

Freimas – tai semantinė duomenų sandara su tuščiomis prasminėmis pozicijomis, kurias užpildžius duomenimis, freimas tampa informacijos nešikliu. Freimo idėją 1975 m. pasiūlė Minsky: freimas sukoncentruoja visas žinias apie objektų klasę su vieninga duomenų struktūra. Taip susiejamos deklaratyvinės ir procedūrinės žinios. Freimas susideda iš lizdų:

<(lizdo vardas), (lizdo reikšmė)>.

Galima atlikti įvairias operacijas su freimais: santykio operacijos, jungtis, sankirta. Taip pat svarbi freimo savybė – paveldimumas, kuris leidžia aprašyti įvairias dalykines sritis.

Žinių bazių tipai

Taisyklėmis paremta žinių bazė pradėta formuluoti apie 1980-uosius. Taisyklėmis paremta žinių bazė dažniausiai susideda iš taisyklių rinkinio. Šios taisyklės yra aprašomos kaip

IF <sąlyga>
THEN <išvada>

teiginiai. IF dalis aprašo sąlygą, kuriai esant įvyks THEN nurodyta išvada. Kai tik programa pamato kokį nors pokytį, ji ieško tinkamos taisyklės, kurią galėtų pritaikyti. Yra dvi taisyklių paieškos rūšys: duomenimis varomas arba tikslo varomas. Duomenimis varoma paieška dar vadinama grandinės į priekį (forward chaining) metodu. Programa turi suprasti pateiktus faktus ir iš jų padaryti išvadą. Grandinė prasideda sulyginus turimus faktus su taisyklėmis iš taisyklėmis pagrįstos žinių bazės. Kai tik taisyklė bazėje randama, tos taisyklės išvada pridedama prie žinomų faktų rinkinio. Šis naujas faktas gali būti sulyginamas su kita taisykle. Antrasis paieškos tipas yra vadinamas atgaline grandine (backward chaining), kai programa jau žino išvadą, bet nori ją patikrinti.

Taisyklėmis pagrįsta žinių bazė naudoja eksperimentines žinias, išgautas iš žmogaus – eksperto. Tokia žinių bazė gerai veikia nedidelėje apibrėžtoje srityje: jei sritis yra apibrėžta, pakankamai maža ir gerai aprašyta, tuomet įmanoma aprašyti visas joje galinčias nutikti tikimybes ir numatyti situacijas. Taisyklėmis paremta žinių bazė programai lengva rasti uždavinio išvadą.

Jei srities žinios nėra labai aiškios, visuomet yra tikimybė, kad iškils nenumatyti atvejai, kurių aprašytų taisyklių programa tiesiog neras savo žinių bazėje. Euristinės taisyklės negali susitvarkyti su trūkstama informacija ar netikėtomis duomenų reikšmėmis. Jei reikšmė yra netikėta ar jos iš vis trūksta, programa negalės parinkti taisyklės. Taigi taisyklės negali prisitaikyti prie naujų problemų. Taip pat svarbu paminėti, jog žinios yra labai priklausančios nuo užduoties. Dažnai, taisyklių rinkinys yra adaptuotas konkrečiai problemai ir jo neįmanoma adaptuoti kitai problemai.

Priešingai nei taisyklėmis paremta žinių bazė, atvejais paremta žinių bazė įsimesina prieš tai buvusį atvejį į atvejų bazę. Atvejis rodo situaciją, kuri nutiko kažkada praeityje. Bendrai paėmus, tai atvaizduojama kaip „problema - sprendimas“ pora. Problema rodo nutikusios situacijos aprašymą ir sprendimą, kuris yra traktuojamas kaip sprendimas priimtas tuo metu (išspręsti iškilusiai problemai). Atvejų bazė susideda iš atvejų rinkinio. Programos veikimas paremtas turimos situacijos ir jai analogišku atveju, turėtos praeityje, sugretinimu.

Atvejais paremta žinių bazė nereikalauja plačios srities žinių analizės. Čia leidžiamos nuorodos į priežastis ir veiksmus: jei susiduriama su aprašytu atveju, sprendimas gali būti rastas labai greitai (greičiau nei generuojant sprendimą nuo uždavinio pradžios). Be to, leidžia programai išvengti jau anksčiau pasitaikiusių klaidų ir pasinaudoti ankstesne sėkme. Atvejais paremtose žinių bazėse, kiekvienos iškilusios situacijos įrašas yra saugomas ir naudojamas naujų problemų sprendimui.

Tokios žinių bazės trūkumai: paieškos laikas priklauso nuo organizacijos bibliotekos, todėl paieškos laikas procese gali išaugti; bazės dydis gali labai išaugti, kai programa mokosi naujų atvejų be sistemos veikimo gerinimo; sudėtingas ir ilgas atvejais paremtos žinių bazės kūrimas.

Žinių bazės integravimas į projektų valdymo sistemą

Bet kuri įmonė, kuri priima žinių valdymo mestą iššūkį, pirmiausia turi išsiaiškinti, ką ji žino ir ko ji nežino. Tik tuomet ji galės sukurti strategijas, pagrįstas savo kompetencija.

„Kuoni“ sėkmė, naudojant „Knows“ („Kuoni“ Nationally Offered Worldwide Statistics – „Kuoni visai šaliai siūloma pasaulinė statistika“), rodo kaip žiniomis pagrįsti produktai gali atverti naujų galimybių rinkoje ir sustiprinti įmonės ilgalaikį konkurencingumą. Tačiau tokie produktai meta ir naujų iššūkių. Naudojant „Knows“, „Kuoni“ sėkmę pasiekė tik todėl, kad ji sukūrė tinkamas sąlygas sistemingam turimo žinių turto valdymui (Probst et al. 2006).

Šiandien naudojamos IT projektų valdymo sistemos dar nėra pakankamai išnaudotos sėkmingam projektų valdymui, nes vykdant naują projektą ir susiduriant su problemomis, kurios jau buvo pasitaikiusios anksčiau, projektų valdymo sistemos dažniausiai nėra pajėgios parodyti ar kas iš darbuotojų jau yra susidūręs su panašia problema praeityje. Taip yra prarandamas projekto laikas, kol randamas reikiamas žmogus, galintis išspręsti susidariusią problemą arba išsprendžiama ta pati problema kitų žmonių pagalba.

Esant dideliems ir sudėtingiems IT projektams, jų planavimui ir organizavimui buvo sukurtos projektų valdymo sistemos, kurios padeda efektyviau apibrėžti projektą ir valdyti dalyvaujančių asmenų roles ir atsakomybes, projekto ar atskirų jo dalių atlikimo terminus, aiškiai matoma sąnaudų apskaita, atlikti darbai, siekiant užsibrėžto tikslo, ir galimos rizikos.

Tačiau ne visos sistemos turi integruotas žinių bases, kurios palengvintų sistemos vartotojų darbą teikdamos informaciją apie analogiškus uždavinius ir jų įgyvendinimui priimtus sprendimus, per kiek laiko užduotis galėtų būti atlikta (remiantis anksčiau pasitaikiusiais panašiais atvejais) vieno ar kito darbuotojo, kokia preliminarinė projekto kaina, rekomenduojamų žmonių sąrašas prjektiniams darbams ir užduotims.

Sėkmingam žinių bazės projektavimui ir integravimui į sistemą, reikalingi keturi pagrindiniai žmogiškieji ištekliai:

- vartotojas – asmuo, kuris naudosis arba jau naudojasi sistema ir suvokia ką ji daro;
- ekspertas – tai pagrindinis žinių šaltinis, svarbiausias objektas žinių bazių kūrime, kuris pasižymi gebėjimu perteikti žinias;
- programuotojas turi išmanyti naudojamos sistemos architektūrą, gebėti įsisavinti naujus metodus, žinoti pagrindinius žinių vaizdavimo metodus;
- žinių inžinierius – svarbiausias žmogiškasis išteklis, kurio pagrindinis ir svarbiausias darbas yra žinių įkėlimas į bazę.

Pirmajame etape vartotojas, ekspertas ir žinių inžinierius turi identifikuoti problemą, tikslus ir laukiamą rezultatą:

- 1) kokios žinios bus kaupiamos;
- 2) kokia žinių struktūra bus naudojama;
- 3) koks žinių bazės tipas bus naudojamas;
- 4) kokios uždavinių klasės ir duomenų šaltiniai bus panaudoti;
- 5) kas bus galutinis sistemos naudotojas, vartotojo sąsaja;
- 6) kokie ištekliai, kompiuterinė technika bus pasitelkti;
- 7) kokiais kriterijais matuojama žinių bazėje sukauptų žinių vertė ir nauda;
- 8) naudotojų skatinimas naudotis žinių baze.

Antrajame etape žinių inžinierius iš žmogaus – eksperto išgauna žinias, kurios vėliau struktūrizuojamos: nustatomos pagrindinės sąvokos, atributai ir ryšiai tarp jų, įeinančių ir išėinančių duomenų struktūra, t.y. sukuriamas neformalus žinių aprašymas. Žinių išgavimas prieš sistemos projektavimą padeda lengviau suvokti galutinį sistemos rezultatą, be to palengvina apsispręsti kokia bus pasirinkta žinių vaizdavimo kalba, numatyti galimas problemas ir iššūkius.

Žinių inžinieriui ir programuotojui atlikus projektų valdymo sistemos architektūros analizę, projektuojama žinių bazė. Į pačią projektuojamą žinių bazę reikia žiūrėti kaip į juodąją dėžę, kuri pateikia atsakymus į korektiškai suformuluotus klausimus. Projektuojant žinių bazę patartina ją izoliuoti nuo kitų komponentų: tuomet ji tampa moduli, todėl su žinių baze galima elgtis kaip su atskiru programiniu moduliu, kuris gali būti panaudotas kelis kartus ar integruotas į kitus programinius modulius.

Vėliau žinių inžinierius ir programuotojas paruošia vietą duomenų bazėje žinių bazei ir pasirinkta programavimo kalba sukuriamas projektų valdymo sistemos su integruota žinių baze prototipas. Šioje vietoje skiriama ypač daug dėmesio problemos sprendimo paieškos greičio optimizavimui bei vartotojo sąsajai. Galiausiai sukurtas prototipas yra testuojamas programuotojo, žinių inžinieriaus, eksperto ir galutinio vartotojo. Nagrinėjamas projektų valdymo strategijos efektyvumas, išsiaiškinamos klaidos ir reikalingi pakeitimai, prototipo privedimui iki galutinės versijos.

Prototipo išvystymo etape pagrindinis darbas – žinių bazės išvedimas. Šiame etape žinių inžinierius perduoda tolimesnį darbą su sistema ekspertui. Pastarasis įvertina sistemą iš išorės, ar gerai suvestos taisyklės, uždavinių kokybė, sistemos kūrėjų kompetenciją. Programuotojas papildomai gali atlikti jautrumo analizę, kai yra keičiamos struktūros ir aplinka, vartotojo sąsaja.

Suderinus sistemą su kitomis programinėmis priemonėmis, duomenų bazėmis, informacinėmis sistemomis, pereinama prie sistemos palaikymo. Čia gali būti papildomai grįžtama į žinių išgavimo žingsnį, kur iš eksperto išgaunamos gilesnės žinios ir taip papildoma žinių bazė.

Dažniausiai įmonės žinioms kaupti ir perteikti yra naudojamos išreikštinės žinios, tačiau vertėtų atkreipti dėmesį ir į neformalioms žinios kaupti ir platinti skirtas technologijas: įmonės forumus, DVS, el. pašta ir kt.

Išvados

Pasiūlytas papildytas žinių bazės integravimo metodas, turintis tris svarbius papildymus:

1. pirminių žinių išgavimas ankstyvoje žinių bazės integravimo stadijoje;
2. projektuojamos žinių bazės kaip „juodosios dėžės“ suvokimas;
3. bendras žinių bazės integravimo metodas papildytas pastebėjimais ir pasiūlymais papildomai testuoti sistemos jautrumo analizę, didesnę dėmesį skirti paieškos greičio optimizavimui ir neišreikštinių žinių kaupimui bei pateikimui.

Literatūra

- Krishna, S. 1992. Introduction to database and knowledge-base systems, *World Scientific Publishing Co.*
- Leondes, C. T. 2005. Knowledge-based systems: Implementation, *Kluwer Academic Publishers*
- Levesque, H. J.; Lakemeyer, G. 2000. The logic of knowledge bases, *Massachusetts Institute of Technology*
- Nikolopoulos, C. 1997. Expert systems: introduction to first and second generation and hybrid knowledge based systems, *Marcel Dekker*
- Probst, G.; Raub, S.; Romhardt, K. 2006. Žinių vadyba. Sėkmės komponentai, *Knygiai*
- Setzer, V. W. Data, Information, Knowledge and Competency . [interaktyvus] [žiūrėta 2011 m. balandžio 1 d.].
Prieiga per internetą: <http://www.ime.usp.br/~vwsetzer/data-info.html>
- Tzafestas, S. G. 1997. Knowledge based systems, *World Scientific Publishing Co.*
- Walker, G. 2001. IT problem management, *Prentice-Hall Inc.*

KNOWLEDGE BASE INTEGRATION INTO PROJECT MANAGEMENT SYSTEM

A. Aleknaitė

Abstract

The study deals with the importance of knowledge inside the organization. The short knowledge base and knowledge-based system analysis, their structure and characteristics are provided. A method of integrating the knowledge bases is presented.

Keywords: knowledge, knowledge base, knowledge-based system, rule-based system, case-based system, semantic network, frame.