VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Julija Škatulaitė

APLIKACIJŲ REIKALAVIMŲ VALDYMO TYRIMAS
ANALYSIS OF APPLICATIONS REQUIREMENTS MANAGEMENT

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 62407T104
Duomenų gavybos technologijų specializacija
Informatikos inžinerijos mokslo kryptis

Vilnius, 2009

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU

Katedros vedėjas

_____
(Parašas)

_____
(Vardas, pavardė)

_____
(Data)

Julija Škatulaitė

APLIKACIJŲ REIKALAVIMŲ VALDYMO TYRIMAS

ANALYSIS OF APPLICATIONS REQUIREMENTS MANAGEMENT

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 62407T104

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos mokslo kryptis

Vadovas prof.habil.dr.Genadijus Kulvietis _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas doc.dr. Angelė Kaulakienė _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas _____ _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Vilnius, 2009

Vilniaus Gedimino technikos universitetas

Fundamentinių mokslų fakultetas

Informacinių technologijų katedra

ISBN ISSN

Egz. sk. ……..

Data ….-….-….

Informacinių technologijų studijų programos baigiamasis magistro darbas

Pavadinimas **Aplikacijų reikalavimų valdymo tyrimas**

Autorius **Julija Škatulaitė** Vadovas prof. habil. dr. **Genadijus Kulvietis**

Kalba

[X] lietuvių

[ ] užsienio

Anotacija

Baigiamajame magistro darbe nagrinėjamas programinių sprendimų kūrimas, aplikacijų reikalavimų apibrėžimo, valdymo problemos IT (informacinių technologijų) projektuose. Aptariama reikalavimų svarba sėkmingam aplikacijų, atitinkančių galutinio naudotojo poreikius, kūrimui. Apžvelgiamas verslo analitiko vaidmuo ir verslo analizės, dokumentacijos procesas bei jo iššūkiai. Pateikiamas P&C (Property and Casualty) draudimo srities reikalavimų valdymo pavyzdys naudojant universalias modeliavimo metodologijas ir Use Case bei veiklos procesų valdymo metodologiją. Išnagrinėjus reikalavimų valdymo proceso poreikius ir galimybes taikant skirtingus reikalavimų valdymo metodus ir įrankius, pateikiamas palyginimas ir išvados.

Darbą sudaro 7 dalys: įvadas, programinės įrangos kūrimo modelių analizė, IT projektų valdymo metodai, reikalavimų valdymo problemos, reikalavimų valdymas naudojant programinius sprendimus, išvados, literatūros sąrašas.

Darbo apimtis – 80 p. teksto be priedų, 55 pav., 2 lent., 45 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

Prasminiai žodžiai: programinės įrangos kūrimas, projektų valdymas, ITIL, CobIT, reikalavimų valdymas, UML, Use Case, RUP, veiklos procesų valdymas, veiklos procesų modeliavimas, BPMN, BPEL.

Vilnius Gediminas Technical University
Fundamental Sciences faculty
Department of **Information Technologies**

ISBN ISSN
Copies No. ……
Date ….-….-….

Information technologies study programme master thesis

Title **Analysis of Applications Requirements Management**

Author **Julija Škatulaitė** Academic supervisor Prof. Dr.Habil **Genadijus Kulvietis**

Thesis language

[X] Lithuanian

[ ] English

Annotation

In the thesis a development of IT solutions, processes of requirements elicitation, enterprise analysis and management in IT projects are analysed. The focus is on a role of applications requirements that ensure projects success and meet end users' expectations on time and within budget. Also, the focus is on business analyst's role, tasks and main challenges upon requirements analysis and documentation. There is provided P&C (Property and Casualty) insurance related requirement example. The requirement is defined using both unified modelling methods with Use Cases and business process management methodology. In the thesis it is analysed the demand for requirements management with different methods and tools.

Thesis consists of 7 parts: Introduction, Analysis of Software Development Models, Analysis of IT Project Management Methods, Requirements Management Problems, Requirements Management with Portfolio, Conclusions, References.

Work coverage: 80 p. of text without appendixes, 55 figures, 2 tables, 45 bibliographical entries.

Appendixes included.

Keywords: Software Development, Project Management, ITIL, CobIT, Requirements Management, UML, Use Case, RUP, Business Process Management, Business Process Modelling, BPMN, BPEL.

Turinys

Santrumpų žodynėlis....4
Paveikslų ir lentelių sąrašas....5
Įvadas....7
1. Programinės įrangos kūrimo modelių analizė....8
1.1. Įvadas....8
1.2. Programinės įrangos kūrimo metodų raida....8
1.3. Šiuolaikiniai programinės įrangos kūrimo modeliai....14
1.4. Išvados....18
2. IT projektų valdymo metodai....19
2.1. Įvadas....19
2.2. ITIL reikalavimai....21
2.3. COBIT atitikimai ITIL....29
2.4. Korporaciniai projektų valdymo modeliai....35
2.5. Išvados....43
3. Reikalavimų valdymo problemos....44
3.1. Įvadas....44
3.2. Reikalavimų valdymo procesų uždaviniai....45
3.3. Universalios modeliavimo metodologijos....47
3.4. BPM metodologija....55
3.5. Išvados....62
4. Reikalavimų valdymas naudojant programinius sprendimus....63
4.1. Įvadas....63
4.2. Reikalavimų apibrėžimo su Use Case praktikų taikymas....63
4.3. Reikalavimų valdymas naudojant BPEL....69
4.4. Išvados....75
Išvados....76
Literatūros sąrašas....77
Priedai....81

Santrumpų žodynėlis

BPEL (angl. *Business Process Execution Language*) – veiklos procesų vykdymo kalba

BPEL4WS *(angl. Business Process Execution Language for Web Services*) – veiklos procesų vykdymo kalba žiniatinklio paslaugoms

BPM (angl. *Business Process Management*) – veiklos procesų valdymas

BPMS (angl. *Business Process Management Suite*) – veiklos procesų valdymo komplektai

BRD (angl. *Business Requirements Document*) – veiklos (verslo) reikalavimų dokumentas

COBIT (angl. *Control Objectives for Information and Related Technologies*) – informacijos ir susijusių technologijų kontrolės tikslai

CRM (angl. *Customer Relationship Management*) – klientų valdymas

ERD diagrama (angl. *Entity-Relationship diagram*) – esybių-ryšių diagrama

ERP (angl. *Enterprise Resource Planning*) – verslo išteklių planavimas

FSD (angl. *Functional Specification Document*) – funkcinės specifikacijos dokumentas

GUI (angl. *Graphical User Interface*) – grafinė naudotojo sąsaja

HTTP (angl. *Hyper Text Transfer Protocol*) – hipertekstų perdavimo protokolas

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

ITIL (angl. *Information Technology Infrastructure Library*) – IT infrastruktūros biblioteka

ITSM (angl. *Information Technology Service Management*) – IT paslaugų valdymas

JAD (angl. *Joint Application Development*) – jungtinis sprendimų kūrimas

KPI (angl. *Key Performance Indicators*) – vykdymo indikatorius

P&C draudimas (angl. *Property & Casualty Insurance*) – nuosavybės draudimas

ROI (angl. *Return of Investments*) – investicijų grąža

RFC (angl. *Request for Change arba Change Request*) – pokyčio reikalavimas

RUP (angl. *Rational Unified Process*) – Rational unifikuotas procesas

SOA (angl. *Service-oriented Architecture*) – į paslaugas orientuota architektūra

SOAP (angl. *Simple Object Access Protocol*) – paprastas objektų pasiekimo protokolas

QA (angl. *Quality Assurance*) – kokybės užtikrinimas

UC (angl. *Use Case*) – naudojimo atvejis

UDDI (angl. *Universal Description, Discovery and Integration*) – universalus aprašymas, atradimas ir integracija

UML (angl. *Unified Modeling Language*) – unifikuota modeliavimo kalba

WSDL (angl. *Web Services Description Language*) – žiniatinklio paslaugų aprašomoji kalba

XML (angl. *Extended Markup Language*) – praplečiamoji dokumentų aprašų kalba

Paveikslų ir lentelių sąrašas

1.1 pav. Programinės įrangos kūrimas (TechnoSolutions 2009)

1.2 pav. Krioklio modelis

1.3 pav. Klaidų vertės augimas

1.4 pav. V formos (su įtrauktu testavimu) modelis

1.5 pav. Iteracinis modelis

1.6 pav. Rizikos palyginimas krioklio ir iteracinio modelių atvejais (IBM 2003)

1.7 pav. Originalaus Agile manifesto vaizdas

1.8 pav. Įrankiai tradiciniams ir Agile procesams

1.9 pav. Pagrindiniai žiniatinklio paslaugų (angl. Web Services) standartai

1.10 pav. Aplikacijos jungiamumas žiniatinklio paslaugomis (angl. Web Services)

1.11 pav. Websphere BPMS įrankiai, leidžiantys dirbti su modeliu pateikta informacija naudojant UML, BPEL standartus

1.12 pav. Veiksmų ir įrankių aplikacijoms kurti apibendrintas modelis

2.1 pav. Projekto valdymo procesas

2.2 pav. Projektų valdymo sferos ir procesai

2.3 pav. Ryšiai: ISO 20000, ITIL ir procedūros

2.4 pav. ITIL suvienija žmones, procesus ir technologijas pristatant IT paslaugas (angl. IT Services)

2.5 pav. Tiltas tarp verslo (veiklos) ir IT technologijų valdant paslaugas

2.6 pav. ITIL procesų gyvavimo ciklo schema

2.7 pav. Paslaugų komplekto sandara

2.8 pav. Paslaugų komplekto kontrolė – priežiūra ir sprendimų priėmimas

2.9 pav. Demingo kokybės ciklas – tęstinė kokybės kontrolė ir stiprinimas

2.10 pav. IT priežūrai sutelktos sritys

2.11 pav. COBIT IT procesai

2.12 pav. COBIT kubas

2.13 pav. RUP 2003 iteracinio ciklo grafikas

2.14 pav. RUP fazės ir jų etapai (angl. Milestones)

2.15 pav. Į paslaugas orientuoto sprendimo įgyvendinimo užduotys

2.16 pav. MOF modelis

2.17 pav. AIM procesai arba disciplinos ir jų ryšys su programinės įrangos gyvavimo ciklu

3.1 pav. Analitiko vaidmens ryšys su kitais projekto dalyviais

3.2 pav. Reikalavimų analizės proceso akcentai

3.3 pav. Reikalavimų valdymo uždaviniai
3.4 pav. UML praktikos skirtingiems tikslams pasiekti
3.5 pav. Apibendrinta reikalavimų apibrėžimo struktūra, naudojant reikalavimų kaupimo įrankius
3.6 pav. Use Case dokumentavimo šablonas (Wiegers 2006)
3.7 pav. Volere šablono modelis
3.8 pav. Volere principo modelis (angl. Volere Requirements knowledge model)
3.9 pav. „Reikalavimo lentynos“ originalo pavyzdys
3.10 pav. Kompanijos ir verslo aplikacijos, organizuojamos pagal atskiras veiklos funkcijas, kai dėmesys sutelktas į tikslą, procesai naikina ribas (Silver 2006)
3.11 pav. Seno požiūrio BPM įgyvendinimas (Silver 2006)
3.12 pav. Naujo požiūrio BPM įgyvendinimas – Clean Handoff stilius (Silver 2006)
3.13 pav. BPM architektūra
3.14 pav. IBM WebSphere Business Suite komponentai ir paskirtis
4.1 pav. Paieškos UC, naudojamas inicijuojant Claim dublikatams išvengti
4.2 pav. Claim (galiojančiam polisui) registravimo UC
4.3 pav. Jira kortelės (angl. Jira Ticket) pavyzdys
4.4 pav. Reikalavimų procesas, palaikomas Rational Requirements Composer
4.5 pav. Išteklio kūrimas IBM Business Modeleriu
4.6 pav. Tvarkaraščio kūrimas IBM Business Modeleriu
4.7 pav. Veiklos objekto Kreiptis1 atributai
4.8 pav. Veiksmo atributų nustatymas Business Modeleryje
4.9 pav. Draudžiamojo įvykio registracijos modelis
4.10 pav. Draudžiamojo įvykio registracijos modelis pagal pasirinktą kriterijų
4.11 pav. Žmonių ir priskirtų vaidmenų ryšių atvaizdavimas naudojant statinę analizę
4.12 pav. Ataskaitos „Process Procedure“ fragmentas parodo veiksmų seką

2.1 lentelė COBIT procesų ir verslo poreikių sąsajos lentelė (ISACA 2009)
3.1 lentelė Produktyvumo gerinimas įvairiose projekto fazėse (Lombardi 2008)

Įvadas

Per pastaruosius 10 metų dėl informacinių technologijų atsirado fundamentalių pokyčių pasaulyje. Padidėjo ne tik vykdomų operacijų ir komunikacijos sparta, bet ir pakito patys verslo modeliai bei veiksnių skaičius. Šiandien korporacijoms, norinčioms išsilaikyti iššūkių rinkoje, neatsiejama dalis yra tarptautinės operacijos, galimas darbų perkėlimas ar padalijimas (angl. *Outsourcing*), visuotinis bendravimas bei bendradarbiavimas, prisitaikantis darbas, patikimumas, verslo tęstinumas, matomumas pasauliniame tinkle ir klientų poreikių tenkinimas. Verslai, taip pat ir atitinkami IT sprendimai, prisitaikydami prie rinkos poreikių bei pokyčių, turi turėti galimybę sureaguoti laiku ir efektyviai pagal situaciją (angl. *On-Demand*).

Aktualumas. Sėkmingam IT projektų įgyvendinimui didelį poveikį turi projektų valdymas. Tai ypač aktualu ilgalaikiams dideliems projektams. Projektų nesėkmės priežastimi dažnai įvardijami būtent prasti reikalavimai. Su projektų valdymo disciplina (apimančia laiko, išteklių, biudžeto, apimties ir t.t. valdymą) glaudžiai susijęs reikalavimų valdymas. Verslo analitikas analizuoja situaciją, identifikuoja verslo tikslus, kliento poreikį, planuoja funkcionalumą, funkcinius ir nefunkcinius aplikacijų reikalavimus, jų svarbą, bendrauja su komanda, taigi kartu su projekto vadovu dalyvauja nustatydamas projekto apimtį, kontroliuodamas eigą. Projektui ir reikalavimų valdymui svarbi dokumentavo metodika, įgalinanti vienareikšmį susitarimą su klientu ir aiškų užduočių pateikimą vykdančiajai komandai.

Mokslinis naujumas. Efektyviam projektų ir reikalavimų proceso suvaldymui žinomas ne vienas metodas. Visgi kiekvienas konkretus atvejis reikalauja individualaus metodų taikymo, savito proceso organizavimo, akcentų išskyrimo. Darbe analizuojamas reikalavimų valdymas, konkrečiam draudimo srities reikalavimo pavyzdžiui naudojama ir universali, ir veiklos procesų valdymo metodologijos.

Praktinis pritaikymas. Darbe atlikta projektų valdymo ir reikalavimų valdymo metodų, jų kaitos ir tendencijų analizė. Pateikta informacija – analizė ir reikalavimo pavyzdys – naudinga valdant reikalavimus, organizuojant procesus, renkant, derinant ir dokumentuojant reikalavimus. Analizė ir reikalavimų valdymo galimybių žinios naudojamos įmonės veikloje.

Šio darbo **tikslas** – ištirti aplikacijų reikalavimų valdymą. Apibrėžus reikalavimų svarbą projekte, apžvelgti reikalavimų inžinerijos proceso (reikalavimų suvokimo, dokumentavimo, pateikimo ir valdymo) užduotis ir problemas, su kuriomis susiduria verslo arba veiklos analitikas. Palyginti verslo analitiko darbo organizavimą naudojant universalią modeliavimo metodologiją ir *Use Case* šablonus ir modeliuojant veiklos procesų valdymo metodologija. Darbui analizuoti pasirinkta nuosavybės draudimo sritis, t.y. viena iš sričių, kur IT labai plačiai taikomos.

Uždaviniai tikslui pasiekti:

- ištirti programinės įrangos kūrimo modelius: eigą, tendencijas, svarbiausius aspektus;
- išanalizuoti projektų valdymo metodus, uždavinius, reikalavimų problemas;
- išanalizuoti reikalavimų valdymo procesą naudojant universalias modeliavimo metodologijas ir BPM metodiką;
- naudojant skirtingus metodus apibrėžti reikalavimų pavyzdį, palyginti valdymo galimybes pasitelkus programinius sprendimus (angl. *Portfolio*).

Darbo rezultatai buvo pristatyti:

- 11-oje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“, Vilnius, 2008.
- 12-oje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“, Vilnius, 2009.

Publikacijos:

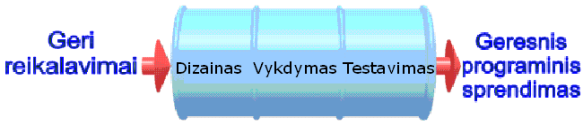
- J.Škatulaitė, T.Rasulovas, J.Mikučionis. Veiklos procesų valdymas. 11-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ straipsnių rinkinys. Vilnius: Technika, 2008. p.82-87.
- J.Škatulaitė. Aplikacijų reikalavimų valdymas. 12-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ straipsnių rinkinys (spaudoje).

1. Programinės įrangos kūrimo modelių analizė

1.1. Įvadas

Sėkmingo programinio sprendimo pateikimas rinkai ar klientui reikalauja suvokimo ir požiūrio įvairiais programinės įrangos inžinerijos aspektais, t.y. programinės įrangos gyvavimo ciklo supratimo ir programinės įrangos kūrimo procesų – projektų valdymo, reikalavimų inžinerijos, programinės įrangos specifikavimo, objektinio projektavimo, sistemų architektūros, naudotojo sąsajos kūrimo, testavimo, sukurtos programinės įrangos palaikymo ir pan. – įvaldymo.

Apibendrintai žiūrint į programinės įrangos kūrimo ciklą, galima išskirti keturis pagrindinius etapus: reikalavimų atskleidimas, analizė ir pateikimas, modeliavimas ir realizacija, produkto testavimas ir kokybės užtikrinimas, rezultatų įvertinimas ir tobulinimas, kurių pagrindinis tikslas - sukurti galutinio naudotojo poreikius atitinkantį sprendimą, jį pristatyti laiku, reikiamos kokybės ir nepažeisti biudžeto (1.1 pav.). Kiekvienu etapu vykdoma daug procesų, jiems atlikti efektyviai žinoma ne viena technika, kurią galima pritaikyti. Šiandien jau žinant daug technikų, turint įrankių ir patirties dėmesys krypsta į efektyvų procesų valdymą, žinomų praktikų taikymą ir gebėjimą greitai reaguoti į aktualius poreikius tam, kad konkretūs projektai būtų sėkmingi.



1.1 pav. Programinės įrangos kūrimas (TechnoSolutions 2009)

1.2. Programinės įrangos kūrimo metodų raida

Kai 1941 m. pasirodė pirmasis skaitmeninis kompiuteris, veikimo instrukcijos buvo susietos su mašina. Tokia tų laikų architektūra dabar vadinama saugomoji programos architektūra arba von Neumanno architektūra. Greitai buvo suprasta, kad techninių įrenginių (techninės įrangos) ir vykdymo galimybių bei funkcionalumo (programinės įrangos) susiejimas yra nelankstus sprendimas, todėl šias dalis pradėta skirti. 1950-aisiais pasirodė programavimo kalbos, kurios dar labiau naikino pastovumą, įnešė lankstumo (Sureau 2009).

Vis giliau suprantant technologijų taikymo naudą, taikymo ir įgyvendinimo galimybes, taip pat srities platumą, 1968 m. NATO programinės įrangos inžinerijos konferencijoje „įvestas“ **programinės įrangos inžinerijos** (angl. *Software Engineering*) terminas. Ši sritis apėmė darbą ir tyrimus, skirtus kokybiškesnės, mažiau lėšų reikalaujančios, lengviau palaikomos, greičiau kuriamos programinės įrangos kūrimui bei tobulinimui. Kartu su terminu, programos kūrimas pradėtas suvokti plačiau nei tik programavimas. 1968 m. gruodžio mėn. dr. Douglas C. Engelbartas ir tyrimų centras (Augmentation Research Center) San Franciske pristatė 90 minučių

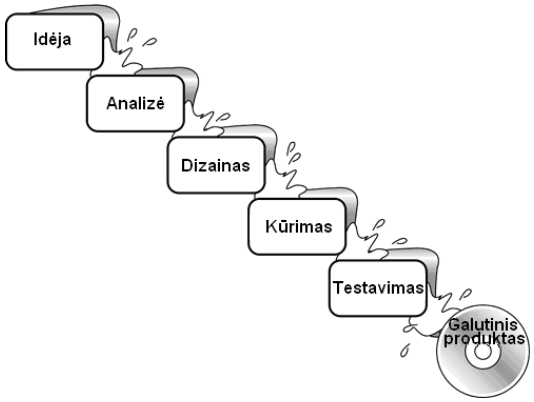
viešą multimedia demonstraciją. Tai buvo personalinės ir interaktyvios kompiuterijos pasaulinis debiutas: pirmą kartą parodyta pelė, kuri valdo tinklo kompiuterių sistemą ir buvo demonstruojamas redagavimas realiu laiku, keli kontroliuojamo vaizdo langai, katodinis monitorius, padalinto vaizdo (angl. *Shared-screen*) telekonferencija (The Programming Languagel Weblog 2009).

Programinės įrangos inžinerijos disciplinų gausa bei žinomos praktikos programinės įrangos inžinerijos koordinavimo komiteto (Software Engineering Coordinating Committee) ir IEEE kompiuterių sąjungos (IEEE Computer Society) iniciatyva nuo 1998 m. buvo surinktos į SWEBOK (angl. *Software Engineering Body of Knowledge*). Idėja buvo išskirti žinomą informaciją ir surinkti žinių paketą (angl. *Body of Knowledge*), kuris atspindėtų konkrečiai sričiai reikalingą kvalifikaciją, ir tai būtų atraminis taškas sertifikacijoms nustatyti. SWEBOK žinių sritys (Knowledge Area, KA) skirstomos į: programinės įrangos reikalavimus (angl. *Software requirements*); dizainą (angl. *Software design*); konstravimą (angl. *Software construction*); testavimą (angl. *Software testing*); palaikymą (angl. *Software maintenance*); konfigūracijos valdymą (angl. *Software configuration management*); inžinerijos valdymą (angl. *Software engineering management*); inžinerijos procesus (angl. *Software engineering process*); įrankius ir metodus (angl. *Software engineering tools and methods*); kokybę (angl. *Software quality*). Sritys, jų gyvavimas susiję su matematika, kompiuterija, vadyba, ergonomika, įvairiomis valdymo (pvz., projektų, kokybės) disciplinomis. Kiekvienai sričiai būdingi sudėtingi procesai, daug metodų užduotims spręsti ir sprendimams priimti, todėl galima teigti, kad SWEBOK susistemina sudėtingą visumą, tačiau kiekviena sritis tiriama, tobulinama, palaikoma atskirai (IEEE Computer Society 2009) Pvz., nuo 1969 m. egzistuojantis JAV projektų valdymo institutas (Project Mangement Institute, PMI) 1996 m. išleido pirmą projektų valdymo praktikų rinkinį (Project Management Body of Knowledge, PMBOK) kuris vis pildomas ir šiandien yra pripažintas JAV standartas ANSI/PMI 99-001-2004; nuo 2003-ųjų įsteigtas tarptautinis veiklos (verslo) analizės institutas (International Institute of Business Analysis, IIBA), kuris pristato verslo analizės žinių rinkinį, ir t.t.

Laikui bėgant mašininis kodas pakeistas asemblerinėmis kalbomis. Aukšto lygio programavimo kalbų naudojimas priartino programavimo kalbų struktūrą prie žmogui suvokiamų sakinių. Tapo naudojamas funkcinis programavimas, vėliau objektiškai orientuoto projektavimo ir programavimo (OOP) samprata. OOP grįstas duomenų abstrakcijos (angl. *Abstraction*), uždaro duomenų tipo (angl. *Encapsulation*), moduliarumo (angl. *Modularity*) ir hierarchijos (angl. *Hierarchy*) pagrindiniais principais, jau seniai tapo įprasta paradigma. Projektuojant naudojami projektavimo šablonai (angl. *Design Patterns*). Be to, kūrimo procese palaikomos ir suprantamos modeliavimo kalbos (pvz., dažnai vizualizacija atliekama UML diagramomis) (Larman 2004). Šiandien kuriant sudėtingus programinius sprendimus, ašimi tampa veiklos ar verslo procesas, teikiamos paslaugos, kuriamos žiniatinklio paslaugos (angl. *Web services*), įgyvendinama į

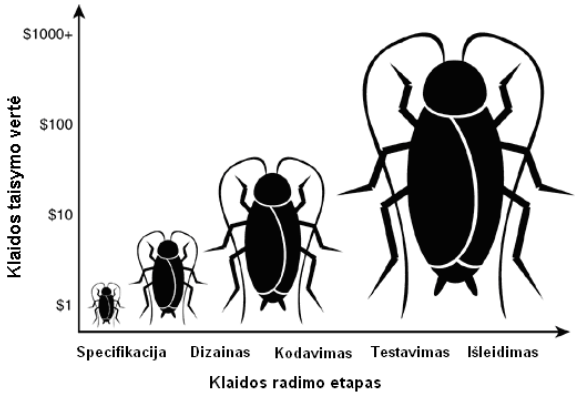
paslaugas orientuota architektūra SOA ir sudėtingas projektų, procesų, uždavinių **valdymas bei tinkamų sprendimų priėmimas.**

Dėl uždavinių gausos kuriant programinę įrangą žinomas ne vienas programinės įrangos kūrimo gyvavimo ciklo modelis (angl. *Software Lifecycle Models*). Klasikiniu programinės įrangos kūrimo modeliu laikomas krioklio (angl. *Waterfall*) modelis, kuriame nuosekliai išskiriamos pagrindinės veiklos grupės, skirtingų komandų atliekami etapai (1.2 pav.). Pirmą kartą jis buvo paminėtas 1970 m. kaip alternatyva paprastam kodavimo ir taisymo (angl. *Code & Fix*) modeliui (Futrell, Shafer, Safer 2002).



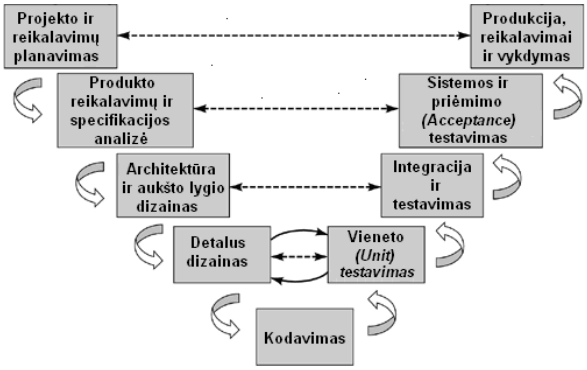
1.2 pav. Krioklio modelis

Tai paprastas, elegantiškas, nuoseklus ir formalizuotas modelis, kūrimas vyksta aiškiai, disciplinuotai. Didelis dėmesys skiriamas reikalavimų rinkimui/analizei, architektūrai, projekto dokumentacijai. Kuriama krioklio principu vis „žemyn“, taigi kiekviena nauja stadija pradedama, kai baigiama ankstesnė, nėra sanklotų, aiški tvarka. Dirbant pagal tokį modelį kartais netgi per daug laiko sugaištama reikalavimų specifikacijai. Kadangi rezultatas matomas tik pabaigoje, reikalauja labai kvalifikuotos komandos ir aiškių reikalavimų. Testavimas aiškus, nes yra produktas ir aiški specifikacija, bet kadangi testuojama tik pabaigoje, tai klaidos brangiai kainuoja, nes kuo vėlesniame etape klaida atrandama, tuo jos vertė projekte didesnė – taisoma ilgiau, įtraukiama daugiau žmonių ir pan. (1.3 pav) (Futrell, Shafer, Safer 2002).



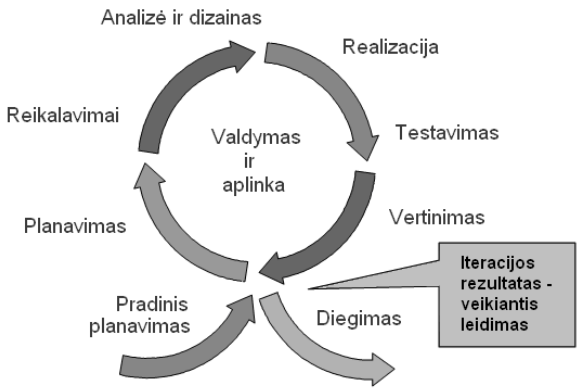
1.3 pav. Klaidų vertės augimas

Taigi, kai matome rezultatą tik pabaigoje ir testavimas vyksta paskutiniame etape - brangi klaidų kaina. Be to, modeliui įgyvendinti turi būti iš anksto puikiai žinomi nesikeičiantys reikalavimai, kas daro modelį netinkamu dideliems projektams, kur svarbu įvertinti ir valdyti besikeičiančius reikalavimus ir mastus. Norint įtraukti testavimą ankstesniuose etapuose modelis modifikuojamas į V formos (*V-shaped*) modelį, pavaizduotą 1.4 pav. (manoma, kad modelis sukurtas apie 1991-uosius metus) (Futrell, Shafer, Safer 2002).



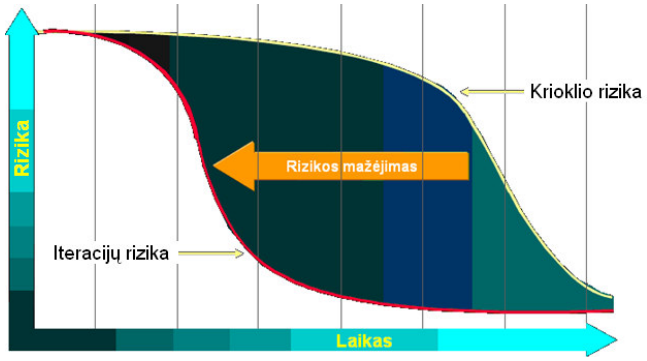
1.4 pav. V formos (su įtrauktu testavimu) modelis

Čia išlaikoma krioklio modelio tvarka, aiškumas ir kt. Modelis pabrėžia ne tik pristatomos programos, bet ir visų atskirų dalių patikrinimą, patvirtinimą bei vertinimą. Testai kuriami kiekviename programinės įrangos kūrimo metu, smulkiai testuojami absoliučiai visi etapai. Tačiau kaip ir krioklio modelyje reikalavimai turi būti žinomi iš anksto, nesikeičiantys. Toks detalus kiekvieno etapo testavimas, užimantis labai daug laiko, ypač taikytinas iš anksto aiškių reikalavimų sistemoms, reikalaujančioms ypatingų garantijų, pvz., aviacijoje, ligonio kontrolės sistemoje, oro pagalvių reguliavime. Visgi, greitai besikeičiančioje informacinėje visuomenėje moderniems veiklos sprendimams dideliems projektams labiau tinkamas iteracinis spiralinis (angl. *Spiral*) modelis, pristatytas 1986 m. Barry Boehmo (1.5 pav.) (IBM 2003).



1.5 pav. Iteracinis modelis

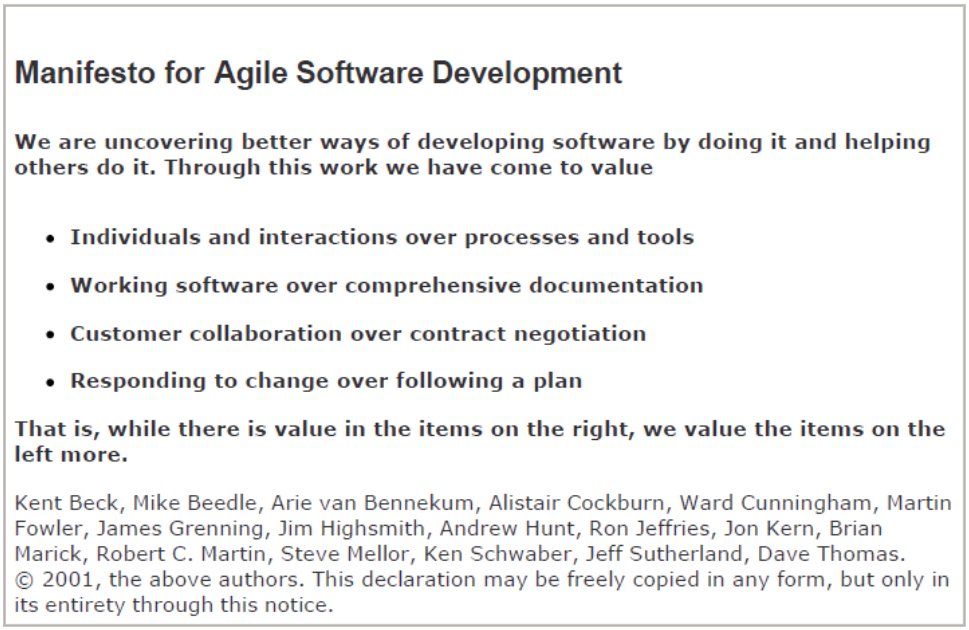
Jame yra paprastesnių modelių privalumai ir pridėti sprendimai silpnosioms modelių vietoms gerinti. Tvarkingas eiliškumas panašus į krioklio modelį, bet vyksta iteracijomis turint grįžtamąjį ryšį. Kiekviena iteracija praeina visus etapus ir baigiasi matoma, vykdoma, testuojama programa (programos dalimi). Testavimo procesas labiau įtraukiamas į galutinio produkto kūrimą, nes testuojama kiekviena fazė, rizika išskaidoma, atskleidžiama anksčiau, ji pasiskirsto per fazes (1.6 pav.). Kuriant fazėmis produktas matomas, yra galimybė keisti, valdyti reikalavimus remiantis grįžtamuoju ryšiu.



1.6 pav. Rizikos palyginimas krioklio ir iteracinio modelių atvejais (IBM 2003)

Nors formaliai krioklio modelį Winstonas Royce‘as pristatė tik apie 1970-uosius, o kiti modeliai buvo pristatyti dar vėliau, realiai dar nuo 1950-ųjų metų NASA ir JAV gynybos komitetas taikė iteracinį ir didėjantį (angl. *Incremental*) programavimą bei kūrimą. 1960 m. Tomas Gilbas įvedė evoliucinio projektų valdymo (angl. *Evolutionary Project Management*), Evo koncepcijas ir tai pristatė 1976 m. Evo rekomenduoja trumpas (savaitės ar dviejų) iteracijas, kurių rezultatas – pristatymas, pabrėžiant pristatymo dalimis naudą. 2001 m. buvo pristatytas Agile manifestas (1.7

pav.) ir **Agile** tapo formalizuotu principu programiniams sprendimams pristatyti rinkinio pavadinimu (Sliger, Broderick 2008).



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value

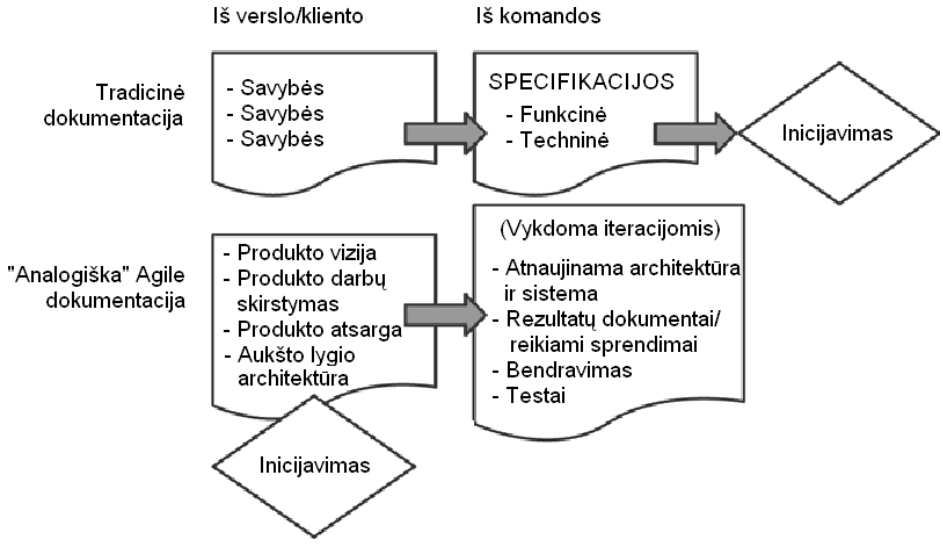
- **Individuals and interactions over processes and tools**
- **Working software over comprehensive documentation**
- **Customer collaboration over contract negotiation**
- **Responding to change over following a plan**

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.
© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.

1.7 pav. Originalaus Agile manifesto vaizdas

Agile‘ui priklauso programinio sprendimo įgyvendinimui skirtų užduočių valdymo metodas Scrum, ekstremalaus programavimo XP, sutelkto į srities (ang. *Domain*) modelio sukūrimą savybėm grįsto kūrimo (angl. *Feature Driven Development*), nuolat prisitaikančio ir besimokančio iš rezultatų adaptyvaus programinės įrangos kūrimo (angl. *Adaptive Software Development, ASD*) sampratos. Agile unifikuotas procesas AUP (angl. *Agile Unified Process*) nuo RUP skiriasi tuo, kad pirmoji disciplina yra modeliavimas (angl. *Model*), kuris RUP‘e yra veiklos modeliavimo, reikalavimų, analizės ir dizaino disciplinų kombinacija (Ambler 2008). Agile ypač daug dėmesio skiriama modeliams, jų kūrimui ir vaidmeniui darbo metu (1.8 pav.) (Sliger, Broderick 2008).



1.8 pav. Įrankiai tradiciniams ir Agile procesams

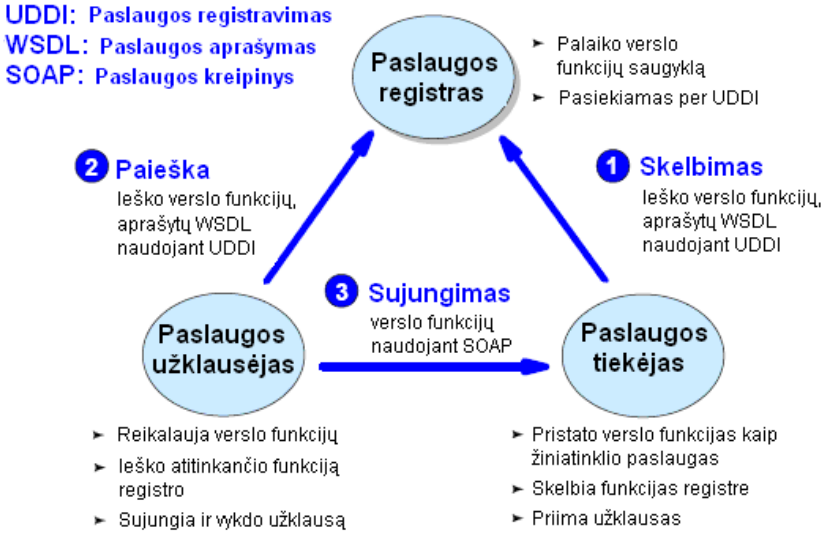
Apibendrintai galima pasakyti, kad Agile (pavadinimas į lietuvių kalbą gali būti verčiamas kaip „judrumas“, „paslankumas“) tikslas sutelktas į lankstumą, metodą, sugebantį prisitaikyti prie aplinkybių, leidžiantį kuo lengviau vykdyti pokyčius ir juos valdyti. Agile ypač daug dėmesio skiriama modeliams, jų kūrimui ir vaidmeniui darbo metu bei skirtingų projekto aspektų atstovų nuolatiniam bendravimui. Agile idealogija yra kurti produktą pagal viziją (angl. *Vision Driven*), o ne iš anksto sudarytą ilgalaikį planą.

1.3. Šiuolaikiniai programinės įrangos kūrimo modeliai

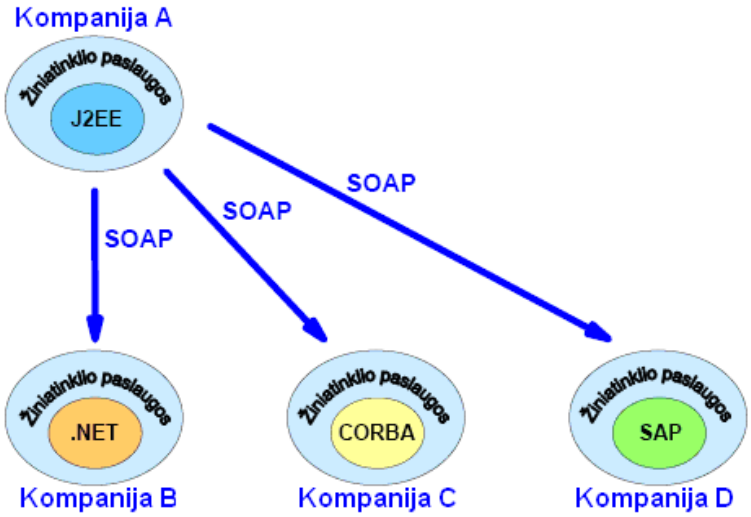
Šiandien norint išlikti konkurencingiems rinkoje, kuriant aplikacijas, daug dėmesio skiriama poreikio atskleidimui, modeliavimui, pažangių technologijų, metodų taikymui, greitos reakcijos ir daugkartinio panaudojimo galimybėms. Reikalingos veiklai funkcijos įgyvendinamos kuriant **žiniatinklio paslaugas** (angl. *Web Services*), kurias galima panaudoti skirtinguose kontekstuose (angl. *Reusable*) ir kurios yra į paslaugas orientuotos architektūros **SOA** pagrindas. SOA – architektūros metodas, kurio principai ir technikos orientuotos į enkapsuliaciją (angl. *Encapsulation*) ir panaudojamų funkcijų bei procesų modeliavimą. Pagrindiniai žiniatinklio paslaugų standartai yra WSDL, UDDI, SOAP, bet ne visi privalomi žiniatinklio paslaugos realizacijai. Naudojimas UDDI (kaip serviso registru) ir SOAP (kaip pranešimo formatu) yra neprivalomas.

BPEL4WS išplečia pagrindinę žiniatinklio paslaugų (angl. *Web Services*) architektūrą visai (angl. *end-to-end*) veiklos proceso kompozicijai pagal žiniatinklio paslaugas. Taigi, žiniatinklio paslaugos įgalina jungiamumą (angl. *Interoperability*) ir daugkartinį panaudojimą (angl. *Reuse*) aplikacijos funkcijos lygmenyje, o BPEL4WS – veiklos proceso lygmenyje. Abiem atvejais

dirbama veiklos ribose ir suderinant tarp partnerių. BPEL4WS taip pat leidžia saugoti būsenos informaciją tarp žiniatinklio paslaugų kreipimųsi veiklos proceso lygmenyje. BPEL yra egzistuojančių žiniatinklio paslaugų standartų veiklos proceso išplėtimas. Anksčiau žiniatinklio paslaugos buvo nepriklausančios sąveikos. BPEL ir kitos procesų ir choreografijos (angl. *Choreography*) kalbos leidžia kurti priklausančius, sąveikaujančius, integruojamus ir vienas kitą suprantančius veiklos procesus iš žiniatinklio paslaugų (1.9, 1.10 pav.) (Keen, Cavell, et al. 2004).



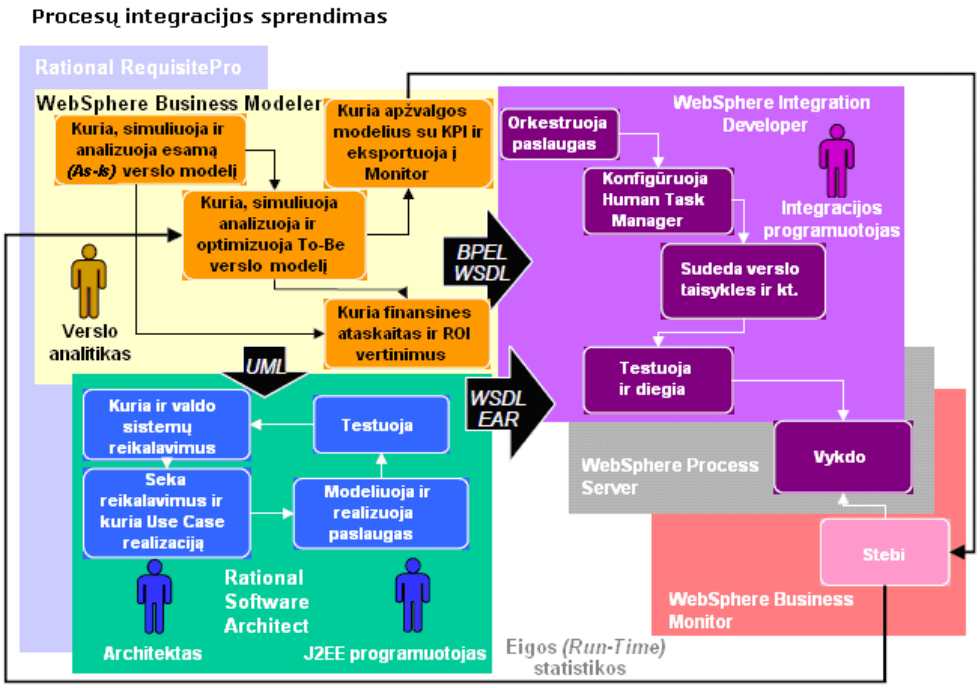
1.9 pav. Pagrindiniai žiniatinklio paslaugų (angl. *Web Services*) standartai (Keen, et al. 2004)



1.10 pav. Aplikacijos jungiamumas žiniatinklio paslaugomis (Keen, et al. 2004)

Visa SOA realizacija transformuoja požiūrį į verslą ir automatizavimą. SOA apibūdina atskirus gebėjimus kaip dalinamus paslaugų (angl. *service*) vienetus, ir veiklos sritis pristatoma kaip atskirų paslaugų kombinacija, tuo būdu turint galimybę greitai perorganizuoti, papildyti,

modifikuoti ir adapduoti reikiamiems verslo (veiklos) reikalavimams. SOA atsirado iš IT pusės. Veiklos verslo aspektui atstovauja BPM projektavimo, automatizavimo, valdymo ir tęstinio tobulinimo disciplina (Swenson 2009). Veiklos (verslo) procesai lemia, kaip vykdomos paslaugos, skirstomos užduotys, integracija, ryšiai. Kuriami BPMS sprendimai, palaikantys visą kūrimo ciklą nuo analitinio modeliavimo iki integracijos, sprendimo vertinimo. Pirmoji pristačiusi ir palaikanti visą kūrimo ciklą yra IBM, sukūrusi IBM Webshpere BPMS (1.11 pav.) (Silver 2007).

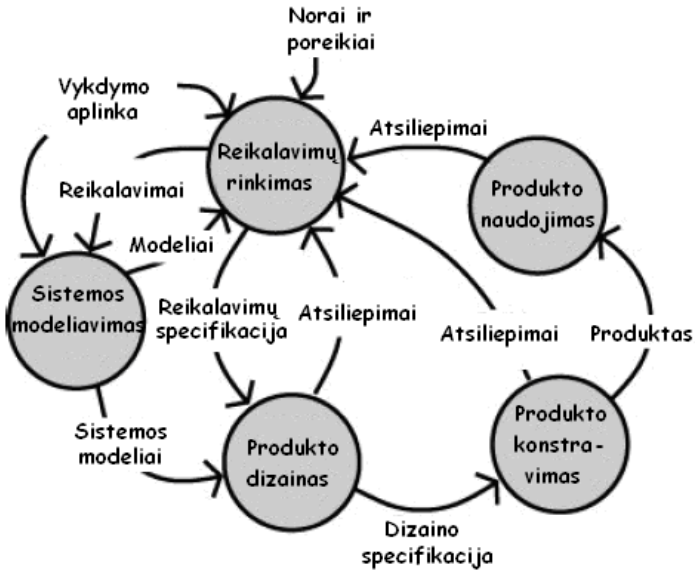


1.11 pav. Websphere BPMS įrankiai, leidžiantys dirbti su modeliu pateikta informacija naudojant UML, BPEL standartus

Tarp šiandien aktualių temų yra verslo (veiklos) taisyklių (angl. *Business Rules*), kurios leidžia valdyti taisykles neprisirišant prie realizacijos technologijų, įgyvendinimas. Verslo taisyklių mechanizmo įdiegimas į sprendimą, pvz., OpenL, leidžia lengvai koreguoti funkcionalumą naudotojams, administratoriams. Tai vienas sprendimų, kurį priima realizuojanti projektą IT komanda. Kita aktuali sąvoka yra verslo intelektika (angl. *Business Intelligence*), naudojama duomenų analizei, apdorojimui ar prognozavimui. Verslo intelektikai (anksčiau vadintai sprendimus palaikančiomis sistemomis) priklauso duomenų sandėlio (angl. *Data Warehouse*), OLAP (*Online Analytical Processing*), duomenų gavybos (angl. *Data Mining*) sritys. Taip pat tiriamos ir tobulinamos ERP sistemos. Sritys tiriamos ir analizuojamos atskirai, žiūrint į jas reikiamu aspektu bei analizuojant ryšį ir integraciją su kitomis žinomomis koncepcijomis.

Žinant ne vieną galimybę, turint įgūdžių bei žinant geriausių praktikų rinkinį, kuriant dideles sistemas, reikalaujančias įvairių sričių specialistų bendradarbiavimo, o taip pat naudojantis darbo padalinimo (angl. *Outsourcing*) privalumais, iškyla sprendimų priėmimo, koordinavimo ir **valdymo** poreikis bei svarba. Kiekvienu momentu reikia žinoti, **ką daryti, kaip, kas ir kada** tai turi atlikti nepamirštant rizikos vertinimo svarbos. Įvairiais etapais reikalingas skirtingas valdymas. Kūrimo valdymas (angl. *Development Management*) skirtas vykdymo koncepcijų ir eigos valdymui, saugumo valdymas sutelktas į saugumo užtikrinimą ir susijusius procesus, skaidymo į procesus ir valdymo dėka galima matuoti našumą, laiką, vertę ir pan. Nepriklausomai nuo strities ar konkrečių įgyvendinimo detalių **projektų valdymui priklauso projekto sėkmės atsakomybė, nuo ko priklauso visi smulkesni procesai ir uždaviniai. Sėkmingai suvaldytas (kartu ir įgyvendintas) projektas, turi atitikti tikrąjį srities ir naudotojo poreikį, pristatytas laiku ir biudžeto rėmuose.**

Pradiniai reikalavimai projekte - tai dalis, egzistuojanti nepriklausomai nuo to, kokios technologijos yra pasirinktos (jie neriboja sprendimo dizaino). Tai pradinis impulsas darbui pradėti. Dizainas kuriamas supratus tikslą ir įžvelgiant geriausią tinkamą realizacijos galimybę. Pradėjus naudoti sukurtą produktą, reikalavimai pradeda keistis, jų daugėja. Iškyla naujo funkcionalumo, naujų galimybių, savybių poreikis. Tai natūralus tobulėjimo procesas. Kuriant programinį sprendimą, įvertinamas šis aspektas ir siekiama turėti kuo paslankesnį, legviau pritaikomą produktą (1.12 pav.).



1.12 pav. Veiksmų ir įrankių aplikacijoms kurti apibendrintas modelis

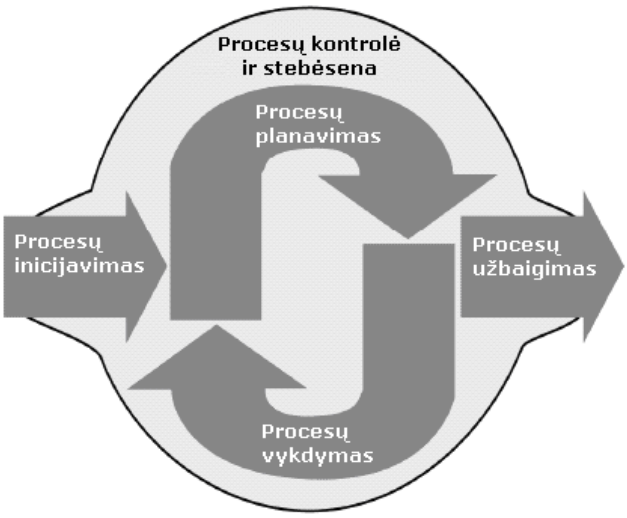
1.4. Išvados

Laikui bėgant sistemos didėjo, augo poreikis, sudėtingėjo uždaviniai. Naujos architektūros, didelis platformų pasirinkimas, išaugęs funkcionalumo poreikis, vis sudėtingesnių reikalavimų įgyvendinimo galimybės, saugumo užtikrinimas, gebėjimas prisitaikyti – tai dalis iššūkių, su kuriais susiduriama kuriant aplikacijas. Šiandieninėje visuotinėje rinkoje, plačiai taikant darbo padalijimo (angl. *Outsourcing*) galimybes sėkmingam aplikacijos sukūrimui nepakanka tik kokybiško realizavimo, programavimo. Aktualu bendradarbiavimas, dalijimasis informacija, prisitaikymas ir integracija. Tinkamai pritaikytas ir valdomas darbo organizavimo ir sprendimo įgyvendinimo modelis bei metodikos padeda išvengti ir iškilus spręsti naudotojų ar veiklos poreikių neatitikimo, sumaišytų reikalavimų, neintegruotų modulių, sunkaus palaikymo, vėlaus defekto suvokimo/radimo, prastos kokybės, produktų defektų, prastos eigos (angl. *Performance*), sunkaus testavimo palaikymo (angl. *Testable*), nekoordinuotų komandos veiksmų, neplanuotų išlaidų problemas. Užduoties ar globaliau žiūrint projekto sėkmė priklauso nuo tinkamo sprendimo priėmimo, darbo, procesų koordinavimo. Šiandien kuriant sudėtingus programinius sprendimus, ašimi tampa veiklos ar verslo procesas ir jo valdymas, prisitaikymas prie rinkos ir pokyčių, teikiamos paslaugos, kuriamos žiniatinklio paslaugos (angl. *Web Services*), įgyvendinama į paslaugas orientuota architektūra SOA, ir sudėtingasprocesų, uždavinių, projektų **valdymas bei tinkamų sprendimų priėmimas, koordinuotas procesų ir viso projekto valdymas.**

2. IT projektų valdymo metodai

2.1. Įvadas

Norint pasiekti geriausių praktikų ir verslo (veiklos) reikalavimų suderinimo, reikalingas apgalvotas procesų formalizavimas, priežiūra ir kontrolė. IT projektų valdymo disciplina apibrėžia tikslus ir koordinuoja sprendimui įgyvendinti naudojamus išteklius (laiką, pinigus, žmones, technologijas). Projekto lygmenyje iškeliamas tikslas - pristatyti tinkamą produktą. Galimi skirtingi darbo projekte organizavimo variantai, žinomas ne vienas formalizuotas modelis pačiam projektui įgyvendinti. IT projektams projekto valdymas apima visų kūrimo ciklo etapų ir susijusių situacijų valdymą projekto aspektu. Igyvendimo ciklas gali būti skirtingas, pvz., programinei įrangai kurti klasikiniu yra laikomas krioklio (angl. *Waterfall*) modelis, kuriame visi etapai vyksta paeiliui. Praktikoje yra taikomas šiuolaikinei dinaminei visuomenei ypač tinkantis iteracinis modelis, kai darbas vykdomas fazėmis, tuo įgalinant bendradarbiavimą su klientu, lengvesnį apimties (angl. *Scope*) ir pokyčių (angl. *Change*) valdymą. Nepriklausomai nuo pasirinkto programos įgyvendinimo modelio ar projekto valdymo metodologijos (pvz., pagal JAV standartą, ar UK PRINCE2, arba Scrum) projekto valdymo disciplina išlieka aktuali, keliami panašūs uždaviniai, susiduriama su panašiais klausimas, aktualus darbo dalijimas. Akcentuojamos kuriamų IT sprendimų judrumo (angl. *Agile*), prisitaikymo prie aktualių (angl. *On-demand*) poreikių galimybės. Projekto valdymo apibendrintas procesas (inicijavimas, planavimas, vykdymas, kontrolė, užbaigimas), susietas su planuoti-realizuoti-tikrinti-naudoti ciklu, pavaizduotas 2.1 pav. (PMI 2004).



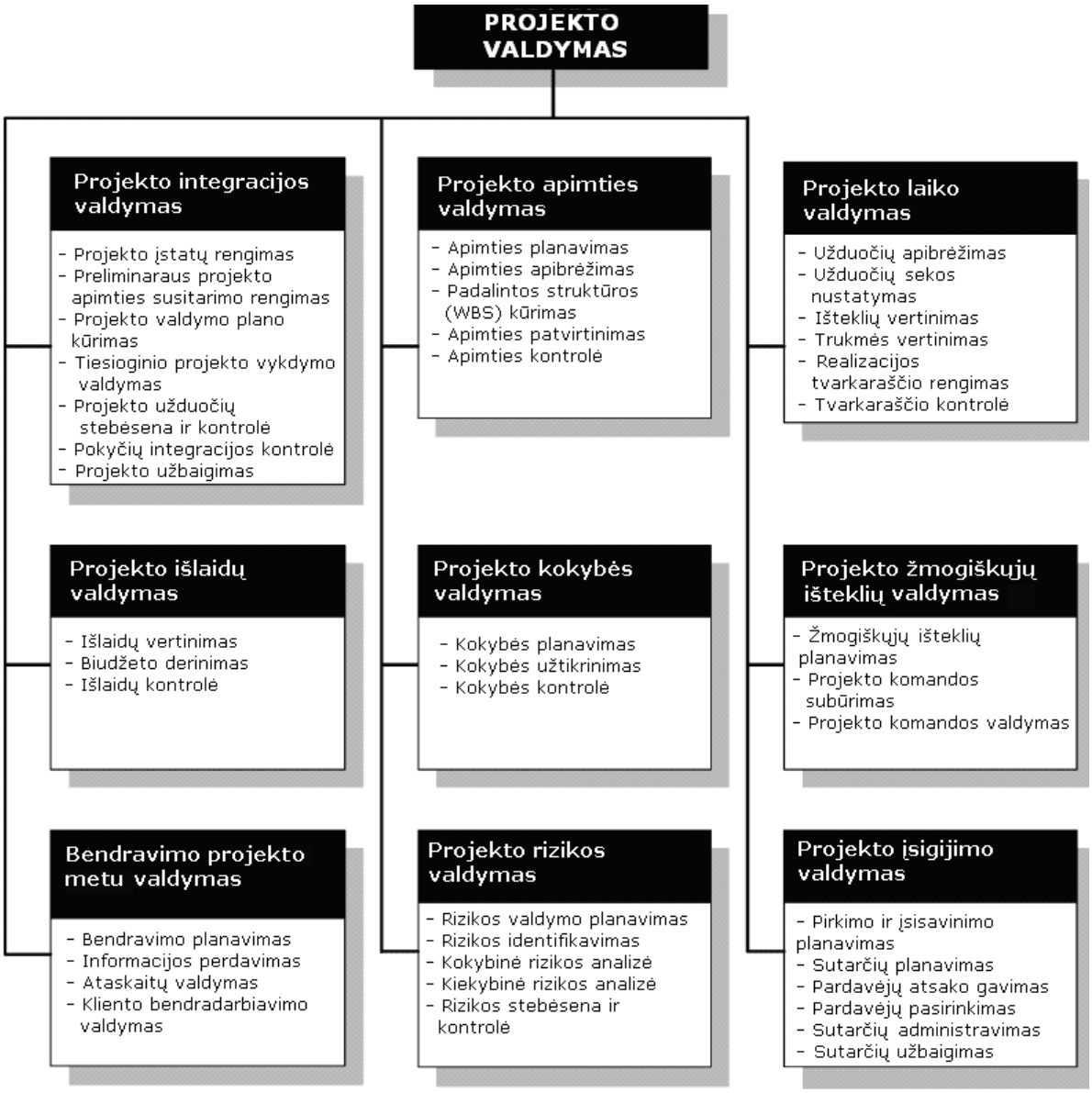
2.1 pav. Projekto valdymo procesas

Projektus koordinuoja projekto vadovas, kurio pirminė atsakomybė yra:

- planuoti ir organizuoti darbą;
- vadovauti kasdieninėms projekto užduotims (angl. *Day-to-day Activities of a Project*);
- pristatyti projekto sprendimą (angl. *Project Deliverables*) klientui;
- identifikuoti galimus klientus.

IT projektuose, projektų valdymo disciplina skirta identifikuoti, stebėti, sekti ir kontroliuoti skirtingus IT aplinkos komponentus bei panaudoti turimą informaciją priimant sprendimus. Projektų valdymas susijęs su skirtingomis darbo grupėmis, susijusia informacija bei ryšiais. Sėkmingam rezultatui svarbus kiekvieno etapo sėkmingumas, taip pat etapų suderinamumas, vienas kito papildymas. Projektų valdymas – plati sąvoka, taigi norint suvaldyti visumą, ji skaidoma į procesus, funkcijas, roles ir pan.

Taigi, projektų valdymo uždaviniai apima projekto koordinavimą įvairiais aspektais, kuriuos galima suskirstyti į 9 sritis ir virš 40 projekto valdymui reikalingų procesų (2.2 pav.) (PMI 2004). Kiekvienas etapas, užduotis reikalauja tinkamų žinių, reikalingų technikų ir/arba darbą palengvinančių programinių sprendimų.

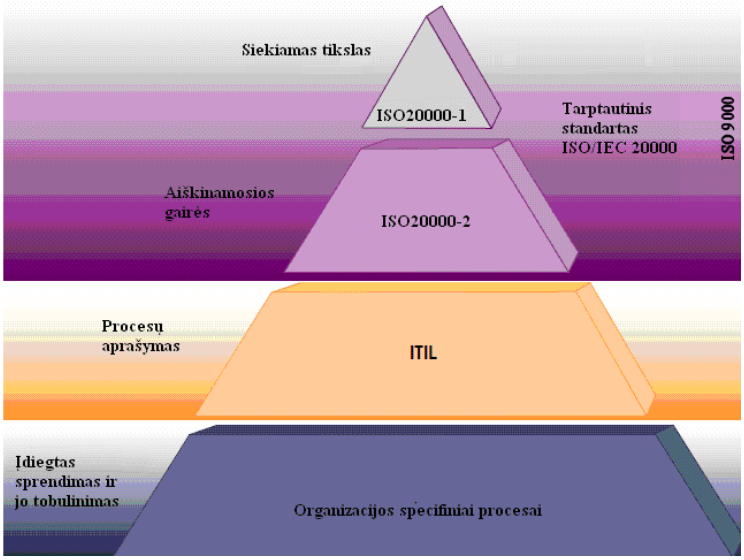


2.2 pav. Projektų valdymo sferos ir procesai

2.2. ITIL reikalavimai

Šiandien viena iš diskutuojamų temų – ITIL įgyvendinimas. **ITIL** (angl. *Information Technology Infrastructure Library*) – tai knygų biblioteka, aprėpianti įvairias įmonių IT paslaugų valdymo sritis. ITIL įvardijama informacinių technologijų ir telekomunikacijų (ITT) verslo valdymo metodologija, orientuota į darbo optimizavimą bei kokybės užtikrinimą IT kompanijose ar įmonių IT padaliniuose. Santrumpa kilusi iš to, kad pirminis ITIL variantas susidėjo iš kelių dešimčių skirtingus IT įmonių procesus aprašančių knygų - IT valdymo bibliotekos. ITIL yra pripažintas Didžiosios Britanijos BS-15000 bei tarptautiniu ISO-20000 standartais, kartu ITIL yra

suderinamas ir su visais ISO-9000 (skirtas nusakyti įmonių vadybos sistemas bei apibendrintus reikalavimus įmonėms, kurioms keliami produktų (prekių ar paslaugų) kokybės kontrolės reikalavimai) reikalavimais. Teisė į ITIL prekinį ženklą priklauso Didžiosios Britanijos vyriausybinei organizacijai - Office of Government Commerce (OGC). Abstrakčiai ryšiai tarp dviejų ISO 20000 standartų ir ITIL gali būti atvaizduojami diagrama - piramide (2.3 pav.). Piramidės viršūnėje yra ISO paslaugų valdymo standarto aprašas, kitas lygmuo yra ITSM praktikos. ITIL biblioteka yra trečiame lygmenyje ir individualios procedūros (angl. *In-house procedures*) pavaizduotos apatiniame lygmenyje (ITIL & ITSM World 2009). Abu papildantys vienas kitą ISO standartai, tiksliau dalys, (*The Specification for Service Management: ISO 20000-1* ir *Service Management Code of Practice ISO 20000-2*) apibūdina serviso valdymo procesus ir formuoja pagrindą valdomų IT servisų vertinimui. Jie buvo paskelbti 2005 metų pabaigoje pagal ankstesnius BS15000 servisų valdymo standartus (angl. *Service Management*).



2.3 pav. Ryšiai: ISO 20000, ITIL ir procedūros

IT servisų valdymo (ITSM) poreikis ir vystymasis laikui bėgant buvo natūrali eiga tobulinant (angl. *Developing*) technologijas. Iš pradžių, dėmesys buvo sutelktas į naujas galimybes kuriant aplikacijas, kurios sprendė problemą, ir galiausiai tapo matomi privalumai pristatyti aplikaciją kaip dalį siūlomų servisų, kurie palaiko patį verslo ar veiklos procesą. **ITIL** pradžia siejama su 1980-aisiais, kai augant servisų valdymo praktikai, augo ir priklausomybė nuo verslo ar veiklos. Tuo pat metu Didžiosios Britanijos vyriausybė dėl efektyvumo poreikio inicijavo komplektuotai dokumentuoti sėkmingas ir geriausias IT servisų valdymo praktikas. Taip iki 1990 m. pradžios apie IT paslaugų valdymo metodiką efektyviai verslui palaikyti buvo pristatyta knygų

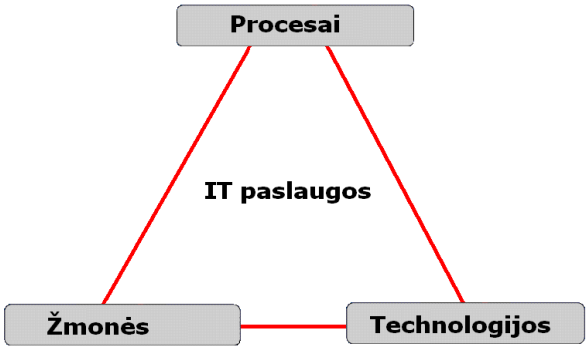
serija, praktikų biblioteka buvo pavadinta ITIL. Pradinė biblioteka augo iki virš 40 knygų ir pradėjo palaikyti reakcijų į interesus grandinę.

Terminas „**IT paslaugų valdymas“** (angl. *IT Service Management*) įsivyravo apie 1990-ųjų vidurį augant ITIL populiarumui. 1991 m. buvo sukurtas IT informacijos valdymo forumas – ***IT Information Management Forum*** (ITIMF) – skirtas susiburti naudotojams ir dalintis informacija, idėjomis bei patirtimi. Laikui bėgant forumo pavadinimas buvo pakeistas į ***IT Service Management Forum*** (itSMF) ir forumo populiarumas auga (ITIL 2007). 2002 m. rugsėjo 27 d. buvo publikuotas formalus ITSM skirtas standartas ***The British Standard 15000*** (BS 15000), pagrįstas ITIL. Šiuo pavyzdžiu pasekė ir kai kurios kitos šalys priimdamos savus nacionalinius standartus. Nors ITIL buvo sukurtas Didžiojoje Britanijoje, jis buvo tinkamas kitoms Europos ir pasaulio šalims. Galiausiai, 2005 m. gruodžio 15 d. buvo publikuotas tarptautinis **ISO/IEC 20000** standartas, pakeitęs nacionalinį BS 15000. ISO/IEC 20000 standartas aprašo tarpusavyje susijusių valdymo procesų pagal ITIL karkasą rinkinį.

ISO20000 susideda iš dviejų dalių:

- ISO20000-1 sudaro 10 sekcijų: apimtis (angl. *Scope*), terminai ir reikšmė (angl. *Terms and Definitions*), valdymo sistemų reikalavimai (angl. *Requirements for a Management System*), servisų valdymo planavimas ir vykdymas (angl. *Planning and Inplementing Service Management*), naujų ir pakeistų servisų planavimas ir vykdymas (angl. *Planning and Implementing New or Changed Services*), serviso pristatymo procesas (angl. *Service Delivery Process*), servisų procesai (angl. *Relationship Processes*), procesų sprendimai (angl. *Resolution Processes*), procesų kontrolė (angl. *Control Processes*), procesų skelbimas (angl. *Release Process*);
- ISO20000-2 padeda organizacijoms įgyvendinant ISO20000-1 (The ISO 20000 Directory 2009).

Atsiradus ISO standartui ITIL metodika ir sertifikacija pripažįstama tarptautiniu mastu. Laikui bėgant ITIL sudedamosios dalys yra tobulinamos, jų skaičius auga. Paskutinė su ITIL susijusi informacija iš Didžiosios Britanijos OGC organizacijos (UK Office of Government Commerce) pateikiama paskelbiant ją puslapyje http://www.best-management-practice.com/. Gartner Group išvadose teigiama, kad 80% neplanuotų prastovų nutinka dėl žmonių ir procesų. Remiantis tyrimais, visa IT veikla gali būti suskirstyta į tris dalis, kurių svarba dažniausiai pasiskirsto taip: 20% – technologijos (technika, programos ir pan.), 40% – darbuotojai (kvalifikacija, atsakingumas ir pan.), 40% – procesai (procedūros, instrukcijos, įvykių registravimas ir t.t.) (2.4 pav.).



2.4 pav. ITIL suvienija žmones, procesus ir technologijas pristatant IT paslaugas (angl. *IT Services*)

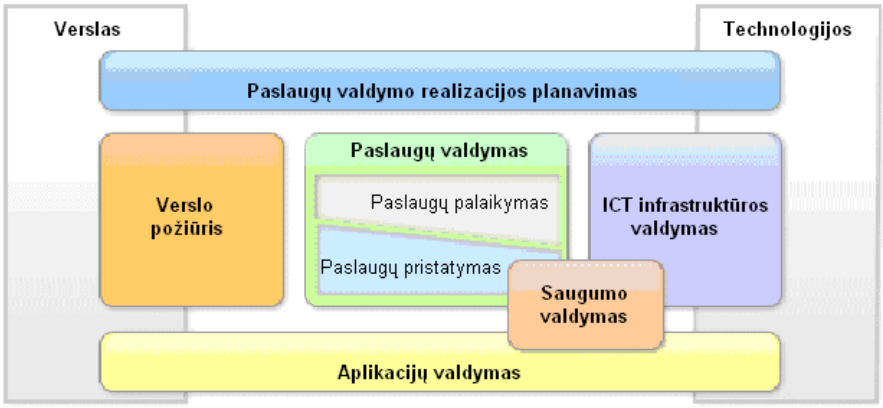
ITIL mažinamos tarp verslo ir technologijų, siekamas efektyvus bendradarbiavimas. Tikslo centre yra **IT paslaugos,** servisai (angl. *IT Services*) ir jų valdymas. Serviso sąvoka ITIL apibrėžiama kaip būdas, skirtas palengvinti kliento norimo rezultato gavimą (angl. *Means of delivering value*), nesusiduriant su papildomomis išlaidomis ar rizika. Serviso prasmė skirtinguose kontekstuose gali būti šiek tiek praplėsta, tačiau iš esmės tai vertės pristatymas (angl. *Delivering value*) (ITIL 2007).

ITIL antroji peržiūra, prasidėjusi 1990-aisiais, baigėsi 2004 m. išleidžiant ITIL Versiją 2 (ITIL Version 2 arba ITIL V2). Antrajai versijai priklauso dar 9 knygos, skirtos sumažinti barjerus tarp verslo ir technologijų, koncentruojantis į reikalingus procesus, norint pristatyti efektyvią paslaugą (servisą) verslo naudotojui. ITIL V2 pagrindinės disciplinos skirstomos į dvi grupes:

- Serviso palaikymas (angl. *Service Support*). Disciplinos: *Configuration Management, Problem Management, Change Management, Help Desk, Software Control and Distribution*;
- Serviso pristatymas (angl. *Service Delivery*). Disciplinos: *Service Level Management, Capacity Management, Contingency Planning, Availability Management, Cost Management for IT Services*).

Be pagrindinių dalių bibliotekai taip pat priklauso: paslaugų valdymo realizacijos planavimas (angl. *Planning to Implement Service Management*), ICT infrastruktūros valdymas (angl. *Information and Communication Technology (ICT) Infrastructure Management*), aplikacijų valdymas (angl. *Applications Management*), verslo požiūrio (angl. The Business Perspective) moduliai, pagrindui išliekant paslaugų pristatymo (angl. *Service Delivery*) ir palaikymo (angl. *Service Support*) dalyse. Pirmoji apima paslaugų pardavimus, sutartis ir pan., o antroji – techninių dalykų valdymą. Šios dvi sritys kartu sudaro ITSM (IT Service Management) ir jos formuoja ITIL pagrindą (dalį, kur dauguma verslų pradeda realizaciją). Kitos papildomos ITIL sritys susijusios su paslaugų valdymo metodologijų diegimu, susiejimu su bendraisiais verslo tikslais ir t.t. Kiekviena

sritis apibūdina priklausančių disciplinų rinkinį (itsm.lt 2009). Visos disciplinos kartu pristato rinkinį disciplinų, palengvinančių sujungti verslą ir IT (2.5 pav.) (Researchcreations 2009).

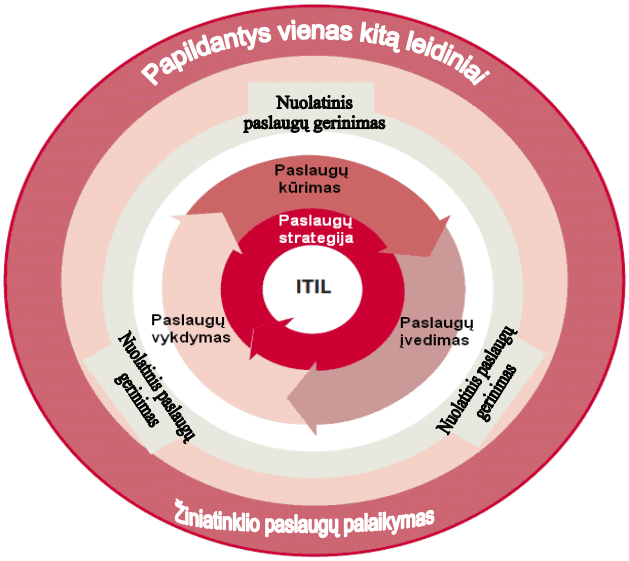


2.5 pav. Tiltas tarp verslo (veiklos) ir IT technologijų valdant paslaugas

2007 m. gegužės mėn. buvo išleista trečioji ITIL versija (**ITIL Version 3**). 3 versija sudaryta iš 5 pagrindinių dalių (knygų), palaikymui skirtų knygų, brošiūrų, dokumentų ir informacinių pranešimų. Pagrindinės dalys:

- paslaugų strategija (angl. *Service Strategy*);
- paslaugų kūrimas (angl. *Service Design*);
- paslaugų įvedimas (angl. *Service Transition*);
- paslaugų vykdymas (angl. *Service Operation*);
- nuolatinis paslaugų gerinimas (angl. *Continual Service Improvement*).

Pirma ir paskutinė dalys - *Service Strategy* ir *Continual Service Improvement* - aprašo įvairiuose gyvavimo ciklo etapuose galimas naudoti praktikas. Kitos trys dalys skirtos griežtam procesų aprašymui (angl. *Rigorously-defined*). Tradicinis skirstymas į dvi pagrindines dalis (*Service Support, Service Deliver*) pagal procesus, dabar ITIL pateikiamas iš šiek tiek naujo požiūrio taško koncentruodamasis į patį gyvavimo ciklą (angl. *Lifecycle*) ir ne tiek į pačius procesus (angl. *Less process-driven*) (2.6 pav.). Pats itSMF vadovas Aidan Lawes teigia, kad labai pasikeitė ITIL koncepcija: „...pagrindinės ITIL knygos dabar yra sudarytos, remiantis ne procesų grupėmis, o procesų gyvavimo ciklu...“ (Itsm.lt 2009.).



2.6 pav. ITIL procesų gyvavimo ciklo schema

Paslaugų gyvavimo ciklo šerdimi laikoma paslaugų strategija (angl. *Service Strategy*), t.y. strategija, kai pateikiami paslaugų valdymo patarimai. Čia bendrai apibrėžiamas (angl. *High Level Approach*) IT organizacijos paslaugų pristatymas. Nustatoma paslaugų rinka, kuri lemia paslaugų pasiūlą, strategiją. Pirmiausia atrenkamos įsivaizduojamos, menamos paslaugos, jos analizuojamos, kol, galiausiai, atitinka tikslą ir tampa projektuojamos, vykdomos *Service Design* etape. *Service Strategy* etapas apima rinkos, situacijos, vidinių ir išorinių tiekėjų charakteristikų tyrimą, paslaugų aprašymą, realizacijos strategijos peržiūrą. Etapas iš esmės skirtas identifikuoti klientą ir rinką, suvokti prioritetus, kad būti tikriems dėl išlaidų ir rizikos suvaldymo. Etapo pagrindinis akcentas – tikslo atskleidimas.

Siekiant, kad sukurtos paslaugos tikrai duotų naudos, jos turi tinkamai atitikti veiklos ar verslo ir kartu naudotojo, kliento poreikį. *Service Design* etape kuriamos naujos paslaugos arba modifikuojamos jau esančios. Užduotis apima tikslių reikalavimų apibrėžimą, sprendimo dizainą, alternatyvų vertinimą, integravimą į visumą. *Service Level Management* disciplinos tikslas įsitikinti, kad būtent sutarta paslauga, atitinkanti užsakovo norus, yra pristatoma klientui. Prižiūrima skirtingais aspektais: tinkamumo (angl. *Availability Management*), pajėgumo (angl. *Capacity Management*), informacijos saugumo (ang. *Information Security Management*), paslaugos tęstinumo (angl. *IT Service Continuity Management*). Taip pat turimų paslaugų visumai kuriamas paslaugų katalogas (angl. *Service Catalog*).

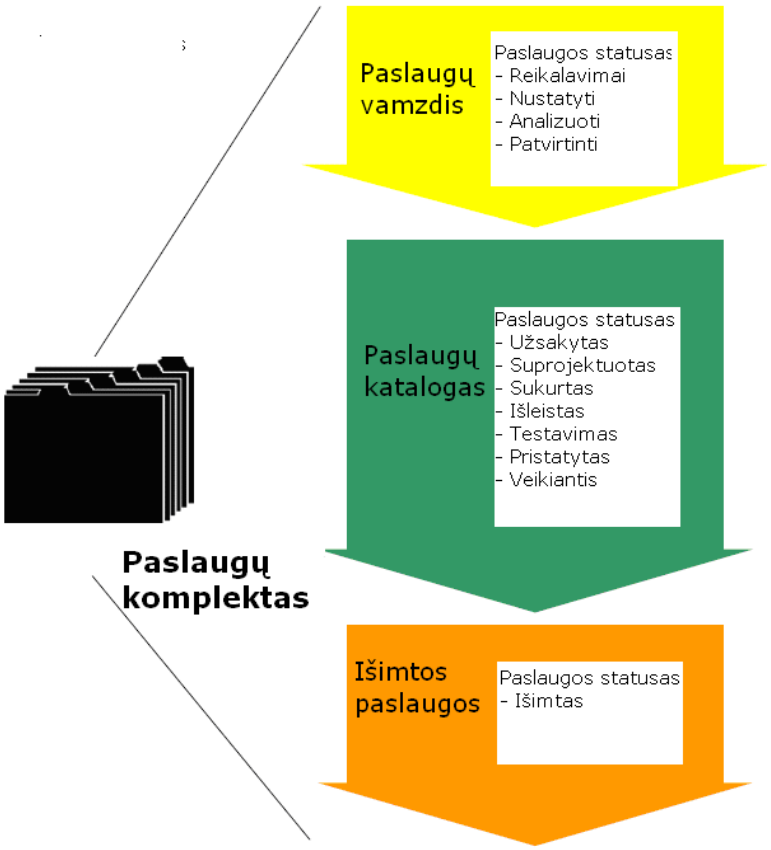
Toliau seka naujos ar pakeistos paslaugos įkėlimas (planavimas, derinimas) į produktą (*Service Transition* etapas) ir galimų paslaugų vykdymas, praktinis taikymas (angl. *Service Operation*). Nuolat tiriamos paslaugų gerinimo galimybės, nes kiekvienu momentu reikia suprasti ir

valdyti aktualų poreikį, funkciją ar savybę. Tarp procesų galimi grįžtamieji ryšiai. Kiekvienu momentu labai svarbi specializacija ir situacijos koordinacija (ITIL 2007).

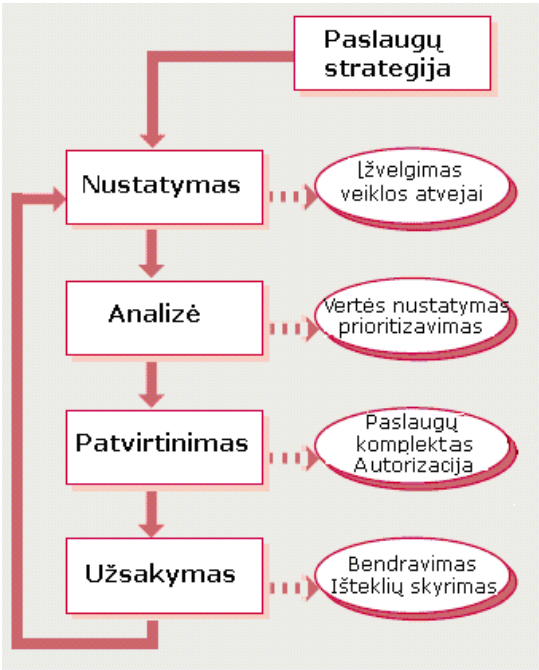
Jau turimos ir dar kuriamos paslaugos komplektuojamos į srities paslaugų komplektus (angl. *Service Portfolio*) (2.7 pav.) (Long 2008). Komplekto tikslas - susisteminti investicijas surinkus turimas paslaugas ir tas, su kuriomis dirbama (t.y. kurios kuriamos arba keičiamos). Be finansinio matomumo, analizuojant komplektą matomi užimti ištekliai, matomos užduotys, ši informacija naudojama planuojant. Komplekte jau egzistuojančios paslaugos sukauptos į paslaugų katalogą (angl. *Service Catalog*), paslaugos, su kuriomis dirbama yra paslaugų vamzdyje (angl. *Service Pipeline*). Išimtų iš apyvartos paslaugų statusas „Išimtas“ (angl. *Retired*). Paslaugų katalogas dažnai naudojamas ir paslaugų strategijos dalyje, nes būtent iš katalogo matomi jau turimi sprendimai, taigi katalogas padeda priimant sprendimus strateginiame lygmenyje. Kataloge galimi paslaugų statusai: užsakytas (angl. *Chartered*), suprojektuotas (angl. D*esigned*), sukurtas (angl. *Developed*), išleistas (angl. *Built*), testavimas (angl. *Test*), pristatytas (angl. *Released*), veikiantis (angl. *Operational*).

Svarbus aspektas yra paslaugų komplekto kontrolė (angl. *Service Portfolio Management*). Reikia suderinti jau turimas paslaugas, jų kiekį su kuriamomis paslaugomis, planuojamų darbų apimtimi, išlaidomis. Paslaugų komplekto kontrolė – dinaminis procesas, apimantis užduotis, reikalingas priežiūrai ir sprendimams priimti (2.8 pav) (ITIL 2007):

- nustatymas (angl. *Define*) - paslaugų išskyrimas, apžvalga veiklos (verslo atvejų) ir turimų duomenų komplekte;
- analizė (angl. *Analyse*) – komplekto vertės analizė, poreikio ir teikimo naudos lyginimas, prioritetų nustatymas;
- patvirtinimas (angl. *Approve*) – siūlymo tvirtinimas, paslaugų ir išteklių pavedimas;
- užsakymas (angl. *Charter*) – sprendimo aptarimas, išteklių ir dokumentų skyrimas.

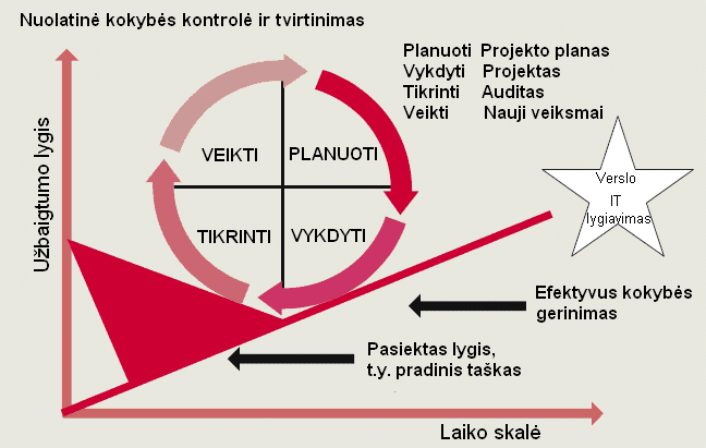


2.7 pav. Paslaugų komplekto sandara



2.8 pav. Paslaugų komplekto kontrolė – priežiūra ir sprendimų priėmimas

ITIL paslaugų gyvavimo ciklo traktavimas apima ir papildo Demingo kokybės ciklą planuoti-vykdyti-tikrinti-veikti (angl. *Plan-Do-Check-Act*) (2.9 pav.) (ITIL 2007). Tai padeda suvienyti ITIL praktikas su COBIT, ISO/IEC 20000 ar kitomis struktūromis.



2.9 pav. Demingo kokybės ciklas – tęstinė kokybės kontrolė ir stiprinimas

Apibendrintai galima pasakyti, kad ITIL orientuojamas į kokybės gerinimą, procesų apibrėžimą ir valdymą, ne į taupų lėšų panaudojimą, finansinį verslo skaidrumą. Norint realizuoti ITIL susiduriama su techniniais, fundamentalaus supratimo, verslo vertės nustatymo, praktinės patirties iššūkiais.

2.3. COBIT atitikimai ITIL

IT specialistams, atsakingiems už informacinių technologijų ir sistemų saugumą, IT auditoriams naudingos COBIT (angl. *Control Objectives for Information and Related Technologies*) žinios. Sritis apima informacinių sistemų audito procesą, IT valdymo struktūros vertinimą, IT sistemų ir infrastruktūros gyvavimo ciklą, IT paslaugų teikimo ir palaikymo valdymo atitikimą teikiamų paslaugų lygiui, vertinimą, ar pakankamai apsaugotos IT priemonės, kad nelaimės atveju užtikrintų nenutrūkstamą veiklos procesą, taip pat pristatomi geriausios praktikos pavyzdžiai.

COBIT yra visame pasaulyje žinomas informacinių sistemų audito ir kontrolės asociacijos ISACA (angl. *Information Systems Audit and Control Association*) ir IT valdymo instituto ITGI (angl. *IT Governance Institute*) standartas. COBIT skirtas pristatyti vadovams, auditoriams ir IT naudotojams bendrai pripažįstamus matavimus, indikatorius, procesus ir efektyvias praktikas IT sričiai ir padėti kompanijoje sukurti atitinkamą valdymą ir kontrolę, įgalinti kokybiškesnį auditą. COBIT pristatomas rinkinys dokumentų, kurie gali būti klasifikuojami, kaip bendrai pripažintos

efektyvios IT priežiūros (angl. *Governance*), kontrolės (angl. *Control*) ir užtikrinimo (angl. *Assurance*) praktikos. Daugelis dokumentų (išskyrus, pvz., audito gidą) yra viešai prieinami internete, tarptautinėje ISACA svetainėje. COBIT misija yra tirti, kurti, skelbti ir aukštinti (skatinti) patikimus, aktualius, tarptautinius kontrolės tikslus tuo padedant profesionalams (*The COBIT Mission: To research, develop, publicise and promote an authoritative, up-to-date, international set of generally accepted information technology control objectives for day-to-day use by business managers, IT professionals and assurance professionals*) (ISACA 2009).

Pirmosios standarto versijos leidimą ISACA pristatė 1996 metais. Organizacijos veiklos pradžia galima laikyti 1967 m., kai susibūrė nedidelė grupelė audito ir informacinių sistemų specialistų. Dabar tai pasaulinio lygio organizacija, jungianti specialistus iš įvairių pasaulio šalių ir atsakinga už IT audito kokybės tobulinimą bei CISA, CISM sertifikacijas. ISACA skyrius Lietuvoje buvo įkurtas 2002 m. Iš pradžių COBIT buvo kurtas kaip IT procesų ir kontrolės karkasas, skirtas priartinti IT ir verslą (veiklą). 1998 m. antrasis leidimas buvo papildytas ir pridėtos *Management Guidelines* „valdymo gairės“. COBIT tapo labiau naudojamas kaip IT priežiūros (angl. *Governance*) karkasas, panaudojant pateikiamas metrikas ir kontrolės modelius. 1998 m. vis aktualėjančią IT kontrolės tematiką, atkreipiant dėmesį į COBIT siūlomą metodiką, tirti buvo įsteigta ne pelno siekianti ITGI. Vėliau ITGI ir CISA apsijungė ir 2000 m. abi institucijos bendrai išleido trečią versiją. 2003 m. versija tapo prieinama interaktyviai (angl. *Online*). 2005 m. gruodžio mėn. buvo išleista ir naujesnė 4.0 versija, o dabartinė 4.1. peržiūra patvirtinta nuo 2007 metų gegužės.

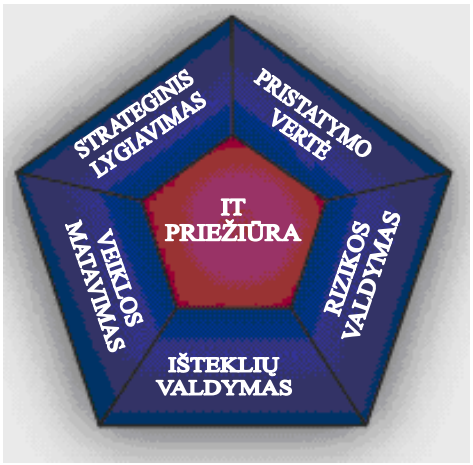
COBIT kombinuojami komponentai, norint užtikrinti organizacijos informacijos nuoseklumą, pastovumą, patikimumą, saugumą, suvokiant verslo procesus, galimą riziką, kontrolės poreikį, technines galimybes. Taigi, standartas yra IT priežiūros (angl. *IT Governance*) įrankis, padedantis verslui įgyvendinti tikslus ir sugebantis palaikyti valdymo svarbos ir IT rizikos koncepcijas. Augant sprendžiamų uždavinių sudėtingumui, o taip pat ir standartų bei siūlomų praktikų pavyzdžių skaičiui susiduriama su iššūkiais (ITGI 2008):

- tinkamai įsisamoninti verslo (veiklos) tikslus ir praktikų naudas;
- sprendimų priėmimą palaikyti su naudojamomis praktikomis ir sugebėti naujus sprendimus integruoti su vidine politika ir procedūromis;
- tinkamai pritaikyti standartus ir geriausias praktikas specifiniams organizacijos reikalavimams.

IT priežiūrai (angl. *IT Governance*) sutelktos sritys (2.10 pav.) (CISA 2009):

- Strateginis lygiavimas (angl. *Strategic Alignment*). Sutelktas ties verslo ir IT planų suderinimu, sujungimu.

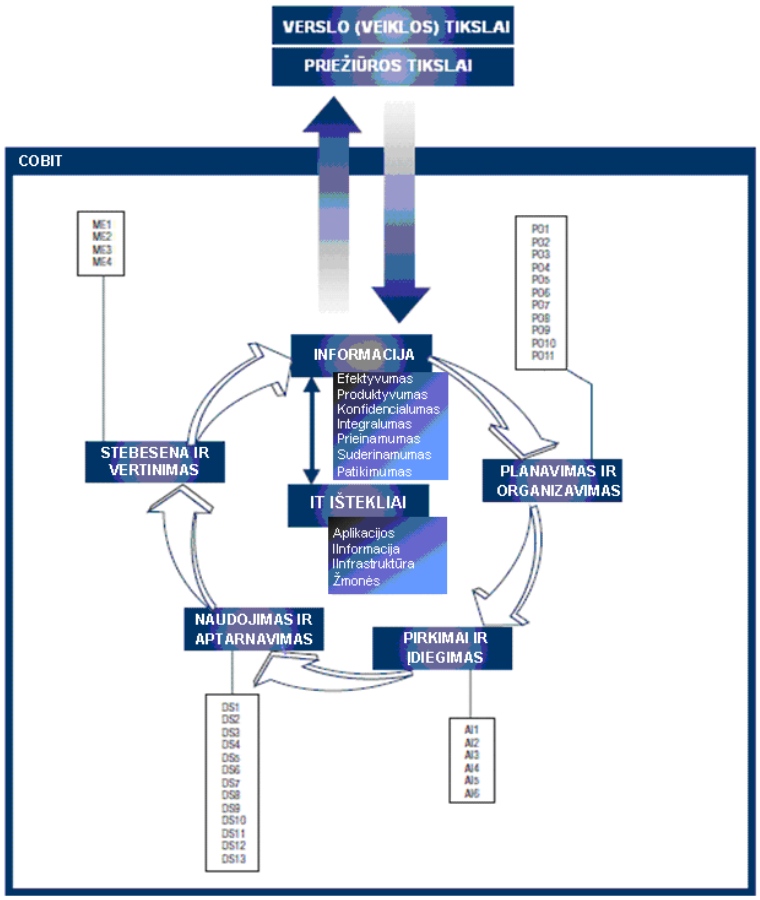
- Pristatymo vertė (angl. *Value Delivery*). Sutelkta ties plano vykdymu visame pristatymo cikle. Įsitikinimas, kad pristatoma paslauga yra tinkama, atitinka strategiją, kreipiamas dėmesys į išlaidų optimizavimą ir IT kokybės gerinimą.
- Išteklių valdymas (angl. *Resource Management*). Analizuoja optimalų invertavimą ir tinkamą išteklių (procesų, žmonių, aplikacijų, infrastruktūros, informacijos) valdymą.
- Rizikos valdymas (angl. *Risk Management*). Reikalaujamas galimos rizikos suvokimas, rizikos esamai padėčiai ir kompanijai vertinimas ir rizikos valdymo atsakomybių nustatymas.
- Veiklos matavimas (angl. *Performance Measurement*). Stebimas strategijos įgyvendinimas, projekto vykdymas bei užbaigimas, procesų atlikimai, paslaugų pristatymas, skaičiuojami ir vertinami veiklos rezultatai įvairiais aspektais.



2.10 pav. IT priežūrai sutelktos sritys

COBIT cikle apjungiami IT ištekliai, procesai, verslo ir priežiūros tikslai. IT procesai grupuojami į keturias pagrindines sritis (angl. *Domains*) (2.11 pav.) (ITGI, OGC 2008) :

- planavimas ir organizavimas (angl. *Plan and Organise*);
- pirkimai ir įdiegimas (angl. *Acquire and Implement*);
- naudojimas ir aptarnavimas (angl. *Deliver and Support*);
- stebėsena ir vertinimas (angl. *Monitor and Evaluate*).



2.11 pav. COBIT IT procesai

COBIT procesai organizuojami pagal verslą (veiklą) (1 lentelė).

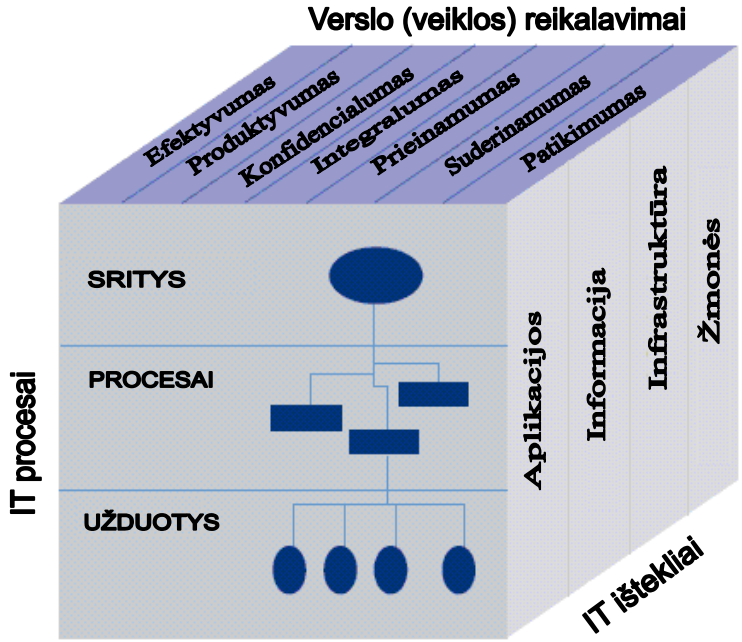
2.1 lentelė COBIT procesų ir verslo poreikių sąsajos lentelė (ISACA 2009)

Procesas		Verslo poreikis
Planavimas ir organizavimas		
PO1	Apibrėžti strateginį IT planą	Rasti balansą tarp IT galimybių ir realių verslo poreikių
PO2	Apibrėžti informacinę architektūrą	Užtikrinti optimalų informacinių sistemų sąryšį
PO3	Pasirinkti technologinę kryptį	Panaudoti esamas ir naujas technologijas verslo strategijos realizavimui
PO4	Apibrėžti IT organizacinę struktūrą ir roles	Užtikrinti vidinių IT paslaugų teikimą
PO5	Kontroliuoti IT investicijas	Užtikrinti IT finansavimą ir finansų panaudojimo kontrolę
PO6	Komunikuoti vadovybės tikslus ir kryptį	Užtikrinti, kad darbuotojai žino ir supranta vadovybės iškeltus IT tikslus
PO7	Valdyti žmogiškuosius išteklius	Įdarbinti ir išlaikyti kompetetingą IT darbuotojų komandą
PO8	Užtikrinti išorinių reikalavimų vykdymą	Užtikrinti teisinių ir sutartinių reikalavimų vykdymą IT srityje
PO9	Įvertinti ir valdyti riziką	Užtikrinti, kad rizikos faktoriai yra identifikuoti ir pašalinti
PO10	Valdyti IT projektus	Nustatyti uždavinius, baigti projektus laiku ir biudžeto rėmuose
PO11	Kontroliuoti kokybę	Patenkinti naudotojų poreikius
Pirkimai ir įdiegimas		

Procesas		Verslo poreikis
AI1	Automatizavimo sprendimų paieška	Rasti tinkamus ir optimalius būdus patenkinti naudotojų poreikius
AI2	Įsigyti ir prižiūrėti programinę įrangą	Teikti automatizuotas funkcijas, padedančias verslo procesams
AI3	Įsigyti ir prižiūrėti technologinę infrastruktūrą	Suteikti platformas, būtinas programinės įrangos funkcionavimui
AI4	Sukurti ir atnaujinti IT naudojimo procedūras	Užtikrinti, kad naudotojai panaudoja programinę įrangą ir technologijas tinkamai, pagal paskirtį
AI5	Įdiegti ir akredituoti sistemas	Užtikrinti, kad programinė įranga tinka pageidaujamai paskirčiai
AI6	Kontroliuoti IT sistemų pokyčius	Minimizuoti neautorizuotų pakeitimų galimybę
Naudojimas ir aptarnavimas		
DS1	Apibrėžti ir užtikrinti tinkamą IT paslaugų lygį	Pasiekti bendrą supratimą apie tai, koks aptarnavimo lygis yra reikalingas
DS2	Kontroliuoti trečiųjų šalių teikiamas paslaugas	Užtikrinti, kad trečiųjų šalių vieta ir uždaviniai yra aiškiai apibrėžti, yra vykdomi ir tenkina naudotojus
DS3	Kontroliuoti sistemų pajėgumus ir apkrovą	Užtikrinti adekvatų sistemų pajėgumą, optimalų panaudojimą
DS4	Užtikrinti nuolatinį sistemų funkcionavimą	Užtikrinti, kad informacinės sistemos veikia, ir netgi didelės problemos atveju nuostoliai bus minimizuoti
DS5	Užtikrinti sistemų saugumą	Apsaugoti informaciją nuo neautorizuoto priėjimo, paskleidimo, pakeitimo, praradimo
DS6	Identifikuoti ir paskirstyti IT kaštus	Užtikrinti suvokimą apie realią IT paslaugų kainą
DS7	Apmokyti naudotojus	Užtikrinti, kad naudotojai naudojasi technologija, yra supažindinti su rizika ir atsakomybe
DS8	Teikti pagalbą naudotojams	Užtikrinti, kad naudotojų patirtos problemos yra sprendžiamos
DS9	Kontroliuoti sistemų konfigūraciją	Identifikuoti visus IT komponentus, patikrinti jų fizinį egzistavimą, užkirsti kelią neleistiniems pakeitimams.
DS10	Sekti, spręsti problemas ir skundus	Užtikrinti, kad problemos yra sprendžiamos, priežastys identifikuojamos ir išvengiama problemos pasikartojimo
DS11	Prižiūrėti duomenis	Užtikrinti, kad duomenys išlieka pilni, tikslūs ir galiojantys įvedimo, atnaujinimo, saugojimo, išvedimo metu
DS12	Prižiūrėti patalpas, kuriose yra įranga	Užtikrinti tinkamą fizinę aplinką, saugančią darbuotojus ir įrangą nuo natūralių ir žmogaus sukeltų nelaimių
DS13	Užtikrinti kasdienį sistemų panaudojimą	Užtikrinti, kad svarbios IT funkcijos yra atliekamos reguliariai ir tvarkingai
Stebėsena ir vertinimas		
ME1	Prižiūrėti IT funkcijų atlikimą	Užtikrinti, kad IT procesams keliami reikalavimai yra vykdomi
ME2	Įvertintinti vidinės kontrolės adekvatumą	Užtikrinti nuolatinį iškeltų tikslų pasiekimą, procesų laikymąsi
ME3	Užtikrinti trečiųjų šalių auditą, garantą	Didinti pasitikėjimą įmone partnerių, klientų tarpe
ME4	Padėti organizuoti audito procesą	Didinti pasitikėjimą įmone, pasinaudoti pasaulinės IT praktikos pasiekimais

Informacijai vertinti, apdoroti pasirenkami kriterijai: produktyvumas (angl. *Efficiency*), efektyvumas (angl. *Effectiveness*), konfidencialumas (angl. *Confidentiality*), integralumas (angl. *Integrity*), prieinamumas (angl. *Availability*), suderinamumas (angl. *Compliance*), patikimumas (angl. *Reliability*). IT naudojami ištekliai grupuojami: aplikacijos (angl. *Applications*), t.y. automatizuotos sistemos, apdorojančios informaciją; informacija (angl. *Information*), t.y. įvairiai

pateikti duomenys, įvestis, apdoroti tarpiniai duomenys ar išvestis; infrastruktūra (angl. *Infrastructure*), t.y. technologijos ir įrankiai, kaip techninė įranga, operacinės sistemos, duomenų bazių valdymo sistemos, multimedia, tinklas ir pan., viską palaikančios aplinkos, kurios įgalina aplikacijų veikimą; žmonės (angl. *People*), t.y. vidinis, iškeltas (angl. *Outsourced*) ar sutartinis (angl. *Contracted*) personalas, kuris turi būti suplanuotas, organizuotas, prižiūrimas vykdo užduotis. Visi trys „varantieji“ komponentai – verslo reikalavimai, IT procesai, ištekliai – yra tarsi trijų dimensijų, taigi COBIT gali būti atvaizduotas kaip kubas. Modelis ir jo principai turi įvairius kriterijus, KPI, naudojamus situacijai nustatyti (2.12 pav.) (ITGI, OGC 2008).



2.12 pav. COBIT kubas

Tiek ITIL, tiek COBIT pristato **apibendrintą vienareikšmiai suvokiamą praktiką**: iškeliami tikslai, su paaiškinimais įvedamas vienodai interpretuojamas žodynas, metrikos, skirstomos funkcijos, išskiriami procesai, nustatomi ryšiai, komentuojamas valdymas ir valdymo galimybės. Kiekvienu atveju siekiama darbo efektyvumo, maksimaliai mažinti ir, reikalui esant, valdyti riziką, viską atlikti kokybiškai ir atitikti standartus. COBIT, besiremiantis ITGI tyrimais ir apibendrinimais (kad IT procesų modeliai būtų tinkami kiekvienai organizacijai), pateikdamas bendrą kontrolės rekomendaciją gali būti naudingas aukščiausiame IT priežiūros lygmenyje. Pagal ideologiją, Informacinių sistemų auditas pagal CobIT atliekamas siekiant suteikti įmonės akcininkams bei vadovams aiškų, nešališką, ir pasauline praktika paremtą eksperto įvertinimą apie tai, kiek jų organizacijos turimos informacinės sistemos yra saugios, tikslingos, efektyviai panaudojamos, valdomos ir pateikia patikimą valdymo informaciją. Taigi, dėmesys labiausiai sutelkiamas ties **finansiniu** aspektu, apskaita. Žemesniame lygmenyje (gilinantis detaliau į

situaciją) be tikslų reikia ir detalizuotų praktikų, smulkesnių aprašymų konkrečiam atvejui, kurių ne visi yra COBIT'e. Tam reikalingi kiti modeliai, praktikos, standartai. Priimant individualius sprendimus galima naudoti siauresnėms sritims sukurtus standartus (pvz., saugumui skirtas ISO/IEC 27002, procesų valdymo rekomendacijos įvairiais aspektais pateiktos ITIL bibliotekoje). ITIL spręsdama efektyvumo, našumo, efektyvios verslo ir IT integracijos uždavinį sprendžia koncentruojantis į organizavimą, IT realizacijos detales. Kadangi pagrindiniai tikslai nepriklauso nei nuo pasirinkto modelio, nei nuo kūrimo etapo – reikalingas atitinkantis poreikį kokybiškas sprendimas laiku ir tinkamo biudžeto – kiekvienos metodikos atveju yra vienodų aspektų, sprendimų ir skiriasi tiesiog požiūris, kur sutelktas atskaitos taškas (ar tai audito išvada, ar paslaugų valdymas, administravimas ir pan.).

Galiausiai, kiekvienas verslas, kompanija turi pats pritaikyti standartus ir praktikas, kad jie atitiktų individualią situaciją, poreikį. Tiek ITIL, tiek COBIT dėmesys sutelktas ties verslo ir IT bendradarbiavimu tikslui psiekti, apjungiant poreikius ir reikalavimus, žmones, išteklius ir įvedant procesus, leidžiančius optimaliai, efektyviai organizuoti darbą. Galima pritaikyti kombinacija yra COBIT, ISO/IEC 27002 saugumo standartas nusakantys tikslą, „**ką reikia daryti**" aspektu ir tikslui pasiekti panaudojant ITIL praktiką, t.y. „**kaip valdyti paslaugas**" aspektu. Valdant sudėtingas sistemas ir jau turint sukaupus ne vieną praktiką, standartą, sukūrus veikiantį modelį dabar dėmesys **krypsta į praktikų, modelių apjungimą, optimizavimą**. Pastebima, kad kiekvieno modelio (metodo) atstovai ruošia modelių palyginimus, atskaitos tašku imdami savo atstovaujamą modelį, sistemą (pvz., ITIL atstovai vertina CobIT iš ITIL pusės, o CobIT atstovai pateikia ITIL procesus iš CobIT pusės).

2.4. Korporaciniai projektų valdymo modeliai

IT kompanijos, remdamosis praktikomis, kuria programinius sprendimus, darbo aplinkas, „griaučius" (angl. *Framework*), padedančius IT verslo atstovams dirbti savo darbą bei įgyvendinti IT sprendimą. Kartu su programų paketu (angl. *Portfolio*) kompanijos pateikia ir savo komercines metodikas, konkretizuotus kompanijai modelius, orientuotus į efektyvų darbą ir žinomas praktikas, standartus bei konkrečios kompanijos produktų naudojimą, vykdomą ciklą. Žemiau pateikta keletas pavyzdžių.

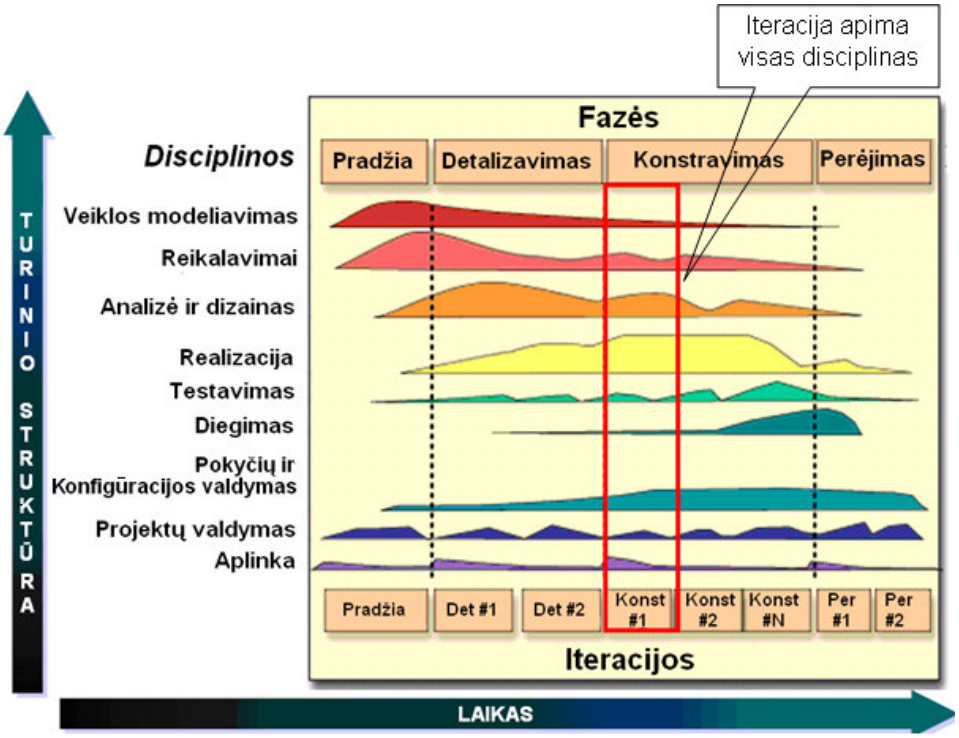
Vienas turinčių ilgą patirtį ir plačiai žinomų koncepcijų yra **RUP** (angl. *Rational Unified Process*) – t.y. programinės įrangos inžinerijos karkasas, pateikiantis lankstų užduočių paskirstymo ir darbo organizavimo metodą programinio sprendimo kūrimo metu. RUP tikslas – palaikyti aukštą kuriamo produkto kokybę, įgyvendinti sprendimą atitinkantį poreikį laiku ir telpant į biudžetą. RUP pradžia siejama su 1980-1990 metais, nuo 2003 m. priklauso **IBM** kompanijai. Programinės įrangos

kūrimas – sudėtingas procesas, susiduriantis su daug rizikingų situacijų. RUP palaiko žinomas praktikas:

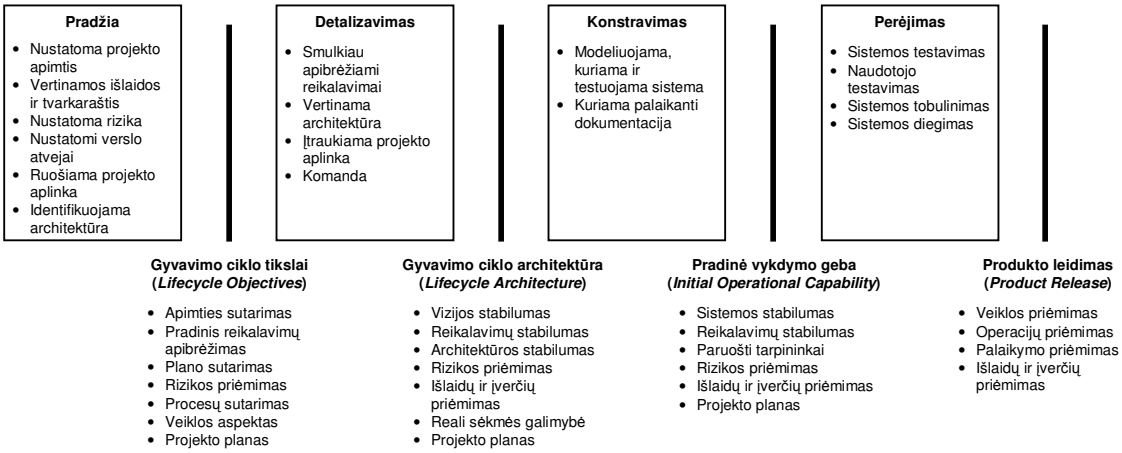
- dinaminė struktūra (vykdymas fazėmis ir iteracijomis);
- svarbi projektų valdymo disciplina, apibrėžianti darbo vykdymą fazių ir iteracijų metu;
- Use Case modeliai ir reikalavimų rizikos sąrašas padeda nustatyti funkcionalumą kiekvienam etapui;
- reikalavimų disciplinos darbo srautų apibrėžimas parodo, kokie galimi įrankiai ir uždaviniai nustatant ir valdant reikalavimus;
- iteracinis metodas leidžia nustatyti reikiamų komponentų eiliškumą, galimą daugkartinį panaudojimą, komponentų pardavimą ar pirkimą;
- UML naudojama procesų atvaizdavimui, apibrėžimui, konstravimui, dokumentavimui. UML tampa „de facto“ standartu.

RUP turinį (angl. *Content*) skirsto į disciplinas, t.y. užduočių, susijusių su pagrindiniu tikslu, rūpima sritimi (angl. *Area of Concern*), rinkinį. RUP apima veiklos modeliavimo (angl. *Business Modeling*), reikalavimų (angl. *Requirements*) analizę ir dizainą (angl. *Analysis & Design*), realizaciją (angl. *Implementation*), testavimą (angl. *Test*), diegimą (angl. *Deployment*), pokyčių ir konfigūracijos valdymą (angl. *Change & Configuration Mgmt*), projektų valdymą (angl. *Project Mgmt*), aplinkos (angl. *Environment*) disciplinas. Projekto ciklas apima keturis etapus (2.13, 2.14 pav.) (Kroll 2003):

- pradžia (angl. *Inception*) – sužinoma, ką reikia sukurti;
- detalizavimas (angl. *Elaboration*) – sužinoma, kaip tai realizuoti;
- konstravimas(angl. *Construction*) – vykdoma produkto realizacija;
- perėjimas (angl. *Transition*) – pridavimas, įdiegimas.



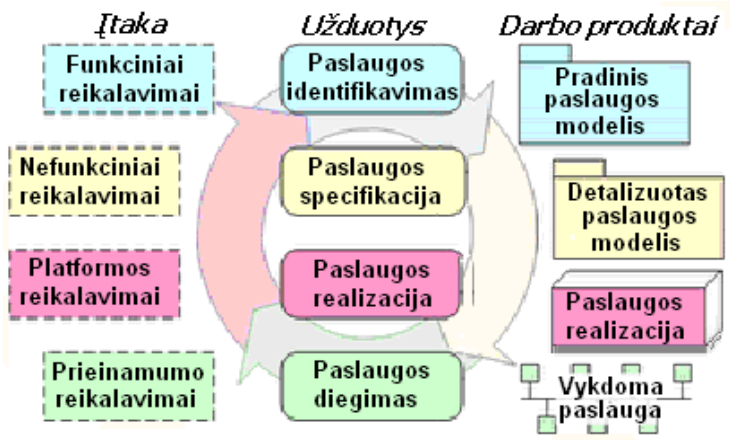
2.13 pav. RUP 2003 iteracinio ciklo grafikas



2.14 pav. RUP fazės ir jų etapai (angl. *Milestones*)

Šiandien jau tapo norma iteracinis, augantis programinės įrangos kūrimas, neabejojama greitos reakcijos į situaciją, verslą poreikiu. Organizuojant darbą, išskiriamos keturios komandų grupės – verslo ar veiklos analitikai (angl. *Business Analysts*), programuotojai (angl. *Developers*), kokybę užtikrinantis padalinys (angl. *Quality Assurance*), projektų valdymo grupė (angl. *Project Management*). IBM pristato *Rational Method Composer 7.1.1*, kuris susideda iš RUP į paslaugas orientuotam modeliavimui ir architektūrai (SOA). Naujesnė RUP 7 taip pat palaiko verslo

modeliavimo ciklą, skirtą apibūdinti ryšį tarp verslo modelio ir paslaugų. Dauguma SOA projektų naudoja verslo procesų modelius verslo, funkciniams ir reikalingų paslaugų (angl. *Services*), galinčių tinkamai palaikyti procesus, reikalavimams suprasti. Norint į paslaugas orientuotam sprendimui realizuoti atliekamas veiksmų ciklas (paslaugų identifikavimas, apibūdinimas, realizacija, diegimas). Veiksmai inicijuojami turint reikalavimus (2.15 pav.) (Steeghs 2009). Didžiausia atsakomybė ir sprendimai tenka sistemų architektui ir verslo (veiklos) analitikui, kuris renka ir pateikia reikalvimus bei inicijuoja įgyvendinimo procesą.



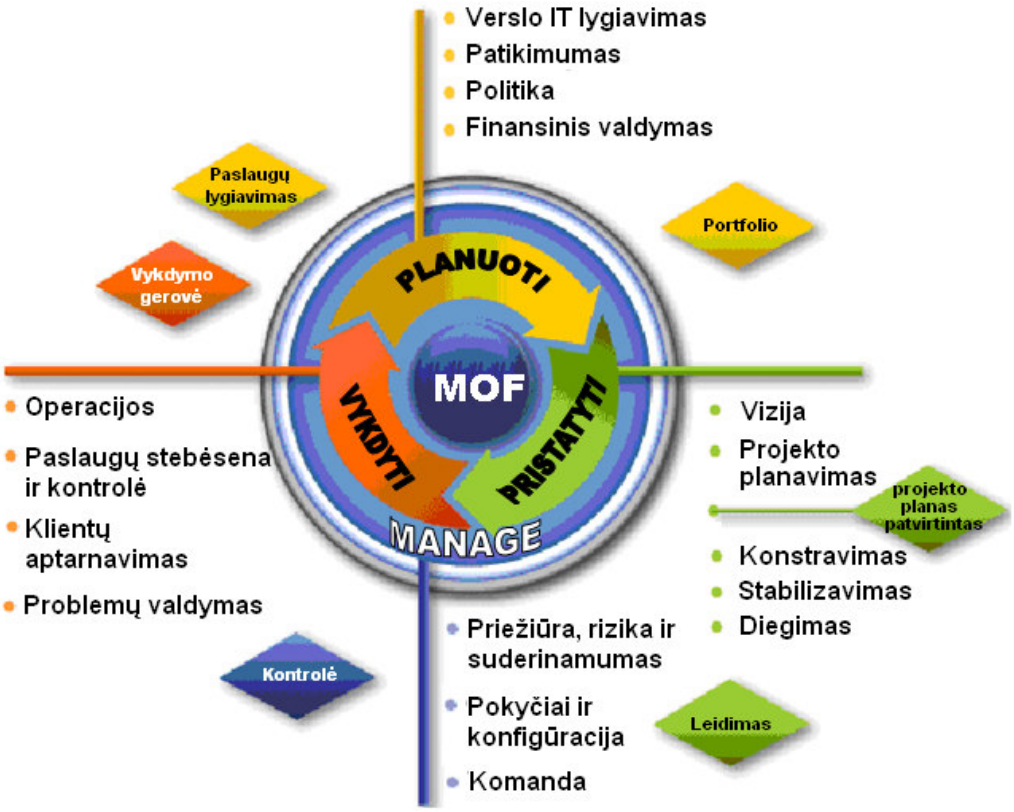
2.15 pav. Į paslaugas orientuoto sprendimo įgyvendinimo užduotys

RUP siejamas ir su WebSphere veiklos integracijos sprendimais (angl. *WebSphere Business Integration solutions*), taip sujungiant verslo ar veiklos modeliavimą su realizacija (WebSphere Business Integration Modeler 5.1 versija palaiko WebSphere Business Modeler 6.0).

Microsoft patikimumui, pasiekiamumui, valdymui pasiekti pateikia savo **MOF** (angl. *Microsoft Operations Framework*) modelį. MOF sukurtas remiantis ITIL, adaptuotas Microsoft produktams ir naudojamoms technologijoms. Pirmasis MOF v1.0 buvo sukurtas1999 vasarą. 2008 m. balandžio 25 d. buvo publikuota dabartinė 4.0 versija. MOF apima IT gyvavimo ciklą integruodamas planavimą, pristatymą, vykdymą ir valdymą su priežiūros, rizikos, suderinamumo užtikrinimu; Microsoft Solutions Framework (MSF) geriausiom praktikom bei valdymo apžvalgom (angl. *Management reviews*). MOF taip pat remiasi ISO 15504 (kitaip Software Process Improvement and Capability Determination, SPICE), skirtu procesų vertinimui (Pultorak D., Bon J. ir kt. 2009). MOF IT ciko procesus dalina į keturias dalis, vadinamas kvadrantais (angl. *Quadrants*) (2.16 pav.) (Microsoft 2009):

- optimizavimas (angl. *Optimizing*);
- keitimas (angl. *Changing*);

– veikimas (angl. *Operating*);
– palaikymas (angl. *Supporting*).



2.16 pav. MOF modelis

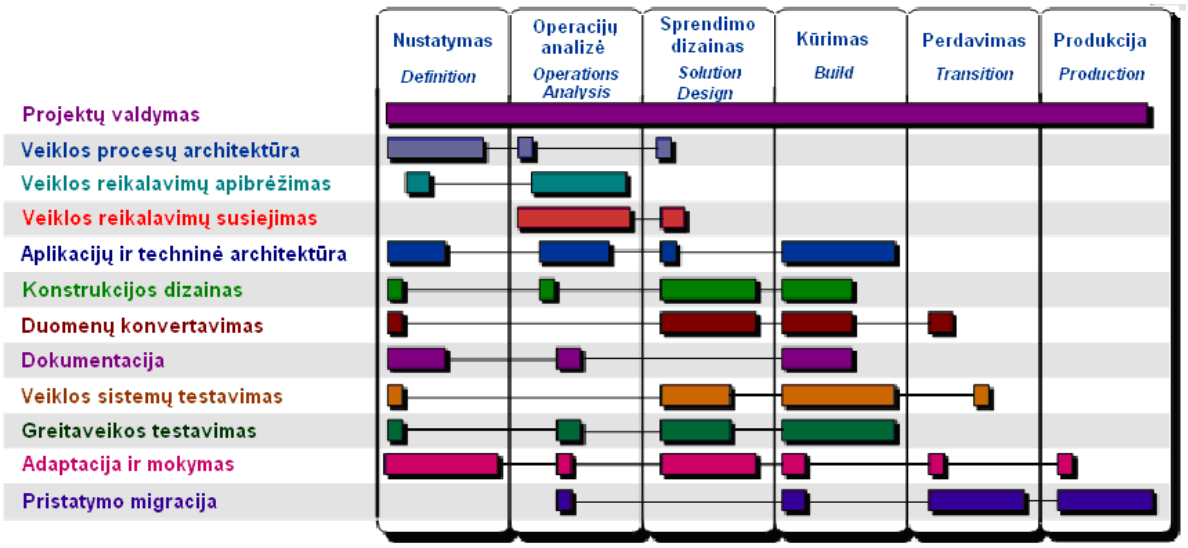
Kadangi MOF sukurtas pagal ITIL, tai kiekvienas kvadrantas susijęs su ITIL procesais. Optimizavimo kvadrantui priskiriami ITIL Availability, Capacity, Financial, Security, Service Level ir Continuity management procesai bei papildomi Infrastructure Engineering ir Workforce Management procesai. Keitimo kvadrantui galima priskirti ITIL Change, Configuration ir Release Management procesus. Veikimo kvadrantui priklauso Service Desk bei Incident ir Problem Management procesai, o palaikymo kvadrantui – stebėsenos bei įvairūs specifiniai, orientuoti daugiau į Microsoft programinę įrangą procesai. MOF detaliai analizuoja programinės įrangos diegimo ir panaudojimo procesus, tačiau yra Microsoft specifiška. MOF galima apibūdinti kaip su ITIL suderinama metodologija, nagrinėjanti procesus, susijusius su Microsoft programinės įrangos diegimu, aptarnavimu ir t.t. (itsm.lt 2009).

Oracle projektų valdymo metodologijai ir realizacijos strategijai pateikti sukūrė aplikacijų realizacijos metodologiją **AIM** (angl. *Application Implementation Methodology*). AIM apima

projektų valdymo metodologiją ir dokumentacijos šablonus, palaikančius valdymui ir realizavimui reikalingas užtuotis. Oracle programinė įranga apima dokumentacijos šablonus, naudojimo instrukcijas ir šablonų valdymui skirtą tinklapį. AIM suderinta ir glaudžiai susijusi su Oracle projektų valdymu PJM (angl. *Oracle's Project Management Method*). PMJ ir AIM užduotys atsispindi darbo paskirstymo struktūroje WBS (Work Breakdown Structure). Su AIM susiję metodai ir įrankiai yra: Oracle veiklos modeliai (angl. *Oracle Business Models*, OBM); sprendimo vertės vertinimas (angl. *Solution Value Assessment*, SVA); Oracle palaikymo vertinimas (angl. *Oracle Support Assessment*, OSA); mokymosi poreikio vertinimas (angl. *Learning Needs Assessment* , LNA); verslo duomenų valdymo sistemos (angl. *Enterprise Data Management System*, EDMS), Oracle vadovas (angl. *Oracle Tutor*) (OracleApps Epicenter 2009).

AIM apimtis (angl. *Scope*) koncentruojasi į verslą ar veiklą kaip į visumą. AIM apima 11 realizacijos procesų ir projektų valdymą, atliekamą viso ciklo metu. AIM disciplinos (2.17 pav.) (Resource Center for Oracle E-Business Suite 2009):

- veiklos procesų architektūra (angl. *Business Process Architecture*) [BP] – bendrai nusako verslo praktikas;
- veiklos reikalavimų apibrėžimas (angl. *Business Requirement Definition*) [RD] – paaiškina apie pradinį klausimyną ir reikalavimų rinkimą;
- veiklos reikalavimų susiejimas (angl. *Business Requirement Mapping*) [BR] – verslo reikalavimai sujungiami su standartiniu Oracle funkcionalumu;
- aplikacijų ir techninė arcitektūra (angl. *Application and Technical Architecture*) [TA] – infrastruktūros reikalavimai Oracle aplikacijų realizavimui;
- konstrukcijos dizainas (angl. *Build and Module Design*) [MD] – pabrėžia klientui reikalingą naują funkcionalumą;
- duomenų konvertavimas (angl. *Data Conversion*) [CV] - duomenų konvertavimas į Oracle aplikaciją;
- dokumentacija (angl. *Documentation*) [DO];
- veiklos sistemų testavimas (angl. *Business System Testing*) [TE] – QA tikrinimas;
- greitaveikos testavimas (angl. *Performance Testing*) [PT] ;
- adaptacija ir mokymas (angl. *Adoption and Learning*) [AP];
- pristatymo migracija (angl. *Production Migration*) [PM].



2.17 pav. AIM procesai arba disciplinos ir jų ryšys su programinės įrangos gyvavimo ciklu

Vienas žinomų komercinių supaprastintų ir konkretizuotų ITIL metodologijos variantų yra **APM** modelis (angl. *Alignability Process Model*). Modelis orientuotas į vieną sritį. APM dalinamas į du modulius, kurie palaiko moduliui skirtus procesus. Paslaugų palaikymo (angl. *Service Support*) moduliui priklauso konfigūracijos (angl. *Configuration Mgmnt*), perspėjimo (angl. *Alarm Mgmnt*), įvykio (angl. *Incident Mgmnt*), problemos (angl. *Problem Mgmnt*), pokyčio (angl. *Change Mgmnt*), versijos (angl. *Release Mgmnt*) valdymo procesai. Paslaugų pristatymo (angl. *Service Delivery*) moduliui priklauso paslaugos lygio (angl. *Service Level Mgmnt*), prieinamumo (angl. *Availability Mgmnt*), gebos (angl. *Capasity Mgmnt*), tęstinumo (angl. *Continuity Mgmnt*), versijos (angl. *Release Mgmnt*) valdymo procesai (Infravision 2009).

Dalis APM modelio privalumų (itsm.lt 2009):

- dėl pavyzdinių procedūrų, apspręstos CMDB ir supaprastinimo modelis įdiegiamas gerokai lengviau, nei klasikinis ITIL;
- neprireikia daryti didelių, kai kada - neįgyvendinamų darbų, pritaikant įmonei ITIL;
- pirmiausiai diegiami svarbiausi ITIL procesai;
- bendra APM įdiegimo savikaina yra bent kelis kartus mažesnė, nei pilnaverčio ITIL;
- yra išsami ir labai patogi dokumentacija.

Esminiai APM modelio trūkumai:

- modelis gana smarkiai (nors ir ne besąlygiškai) susietas su HP OpenView Service Desk programa;
- įmonei plečiantis, modelį gali tekti modifikuoti, priartinant jį prie ITIL, o tam reikia papildomų išlaidų;
- dėl gilesnio detalizavimo modelis yra ne toks lankstus, kaip ITIL;

- APM modelis mažiau, nei ITIL pritaikytas komerciniams IT paslaugų teikėjams;
- kai kurie procesai pernelyg supaprastinti, pvz., Incident Management procesas pilnai apjungtas su ITIL Service Desk funkcija.

APM 1999-aisiais pradėtas kurti Service Management Partners kompanijoje. APM labiau, nei ITIL orientuotas į didelių kompanijų IT aptarnavimo skyrius, mažiau dėmesio kreipia į programinės įrangos kūrimą ir pan., pvz., Release management procesai nagrinėjami labai paviršutiniškai. Dalis skirtumų nuo ITIL: pakeista daug terminų (pvz., Problem Management pakeistas į Root Cause Analysis); kai kurie procesai modifikuoti, t.y. dalis jų yra apjungti, dalies atsisakyta (Incident Management ir Service Desk apjungti į Support Management). Metodologija stipriai konkretizuota - pritaikyta vidiniam įmonės IT naudotojų aptarnavimui. Pradedant APM versija 1.0, išleista 2001 m., ir baigiant naujausia versija 4.2, išleista 2006 m., APM modelis vis labiau artėja prie ITIL - po truputį grąžinami tradiciniai ITIL terminai, procesų struktūra it t.t.

ITIL surinkus geras praktikas ir siūlant gaires (ne konkrečius procesus ir realizacijos detales) paslaugų valdymo procesams, skirtingos kompanijos pagal ITIL siūlo praktikų įgyvendinimą. Apibendrinant, komerciniams modeliams būdingas konkretizavimas (skirtumas nuo ITIL), priklausomai nuo situacijos vienais atvejais yra privalumas, kitais – trūkumas.

2.5. Išvados

Sėkmingam IT projektų valdymui svarbu struktūrizuotas darbas, aiškiai suskirstyti ir apibrėžti vaidmenys, užduotys, procesai. Sprendimų priėmimas turėtų būti pagrįstas esamos situacijos suvokimu, taikant geriausias praktikas ir metodus. Tinkamai pritaikytos žinomos praktikos ir standartai padeda efektyviai valdyti individualias situacijas. Praktikos pateikiamos skirtingomis metodologijomis, priklausomai nuo požiūrio taško. Paslaugų valdymo rekomendacijų rinkinys pateikiamas ITIL metodologija, kuri pagrįsta aksiomos, kad kas neapibrėžta, negali būti kontroliuojama; kas nekontroliuojama, negali būti išmatuota; kas neišmatuota, negali būti tobulinama. COBIT pristatomi kontroliuojami procesai labiau žiūrint finansiniu aspektu. Tinkamai parinkta kombinacija gali duoti geriausius rezultatus, pvz., COBIT ir ISO/IEC 27002 gali padėti apibrėžiant, ką reikia padaryti, o ITIL – kaip padaryti, žiūrint paslaugų valdymo aspektu. Kompanijos pagal pripažintus standartus bei žinomas praktikas siūlo modelius, pritaikytus savo sukurtai programinei įrangai ir technologijoms.

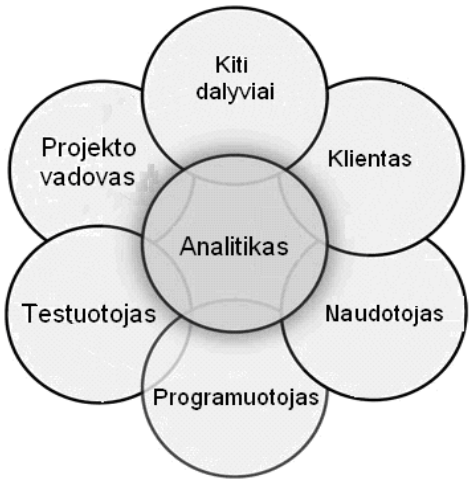
Visais modeliais siekiama tinkamai padalinti projekto visumą į funkcijas, optimaliai skirti atsakomybes ir užtikrinti darnų visumos funkcionavimą. Dėl to svarbus procesų išskyrimas, apibrėžimas, darni sąveika, aiški tvarka, kuri leidžia veikti greičiau, sutelkti dėmesį į konkretų sprendimą, o ne į visumos valdymą. Sėkmingas rezultatas pasiekiamas, kai sujungiamos įvairios sritys (valdymas, veiklos valdymas, auditas), kurioms atstovauja profesionalai. Skirtingų aspektų sujungimo dėka galima pasiekti darbo efektyvumo, išmatuojamumo, lėšų optimizavimo, matomumo ir kontrolės galimybes. Konkreti organizacija turi sukurti sau tinkamus procesus, sugebėti parinkti žmones, technologijas, priemones, pritaikyti modelius ir žinomas sėkmingas praktikas, tam paruošto šablono nėra.

Nustatant projekto tikslus, kuriamas funkcijas ir savybes, valdant projekto apimtį ir siekiant užtikrinti kokybę (ypač, jeigu testavimas yra organizuojamas remiantis dokumentacija ir už kokybę aspektu, kiek tai atitinka verslo ir naudotojo poreikį, atsako ne testuotojų padalinys) **projekto valdymas kertasi su reikalavimų valdymu**. Reikalavimų išaiškinimas ir tikslo suvokimas – pati pirminė bet kurio projekto darbo stadija, kur dirba verslo analitikas.

3. Reikalavimų valdymo problemos

3.1. Įvadas

Efektyvus, produktyvus, inovatyvus verslo ar veiklos procesas galimas tik tinkamai naudojant informacines technologijas ir pasirinkus tinkamiausią metodologiją procesams tinkamu būdu valdyti. Viso projekto esmė – pristatyti laiku ir kokybišką tinkamo biudžeto sprendimą, atitinkantį verslo ir naudotojų poreikius. Būtent verslo ir naudotojų poreikiai, koreguoti techninės realizacijos salygoto apribojimo ir išimčių, yra nustatomi aplikacijų reikalavimų. Be to, laiku tinkamai suprasti ir organizuotai pateikiami bei dokumentuojami reikalavimai padeda kuriančiai komandai efektyviai dirbti. Analitiko darbas glaudžiai susijęs su visais projekte dalyvaujančiais tarpininkais (angl. *Stakeholders*), tiek individualiais asmenimis, tiek tam tikrą etapą atstovaujančiomis grupėmis (3.1 pav.) (Wiegers 2006).



3.1 pav. Analitiko vaidmens ryšys su kitais projekto dalyviais

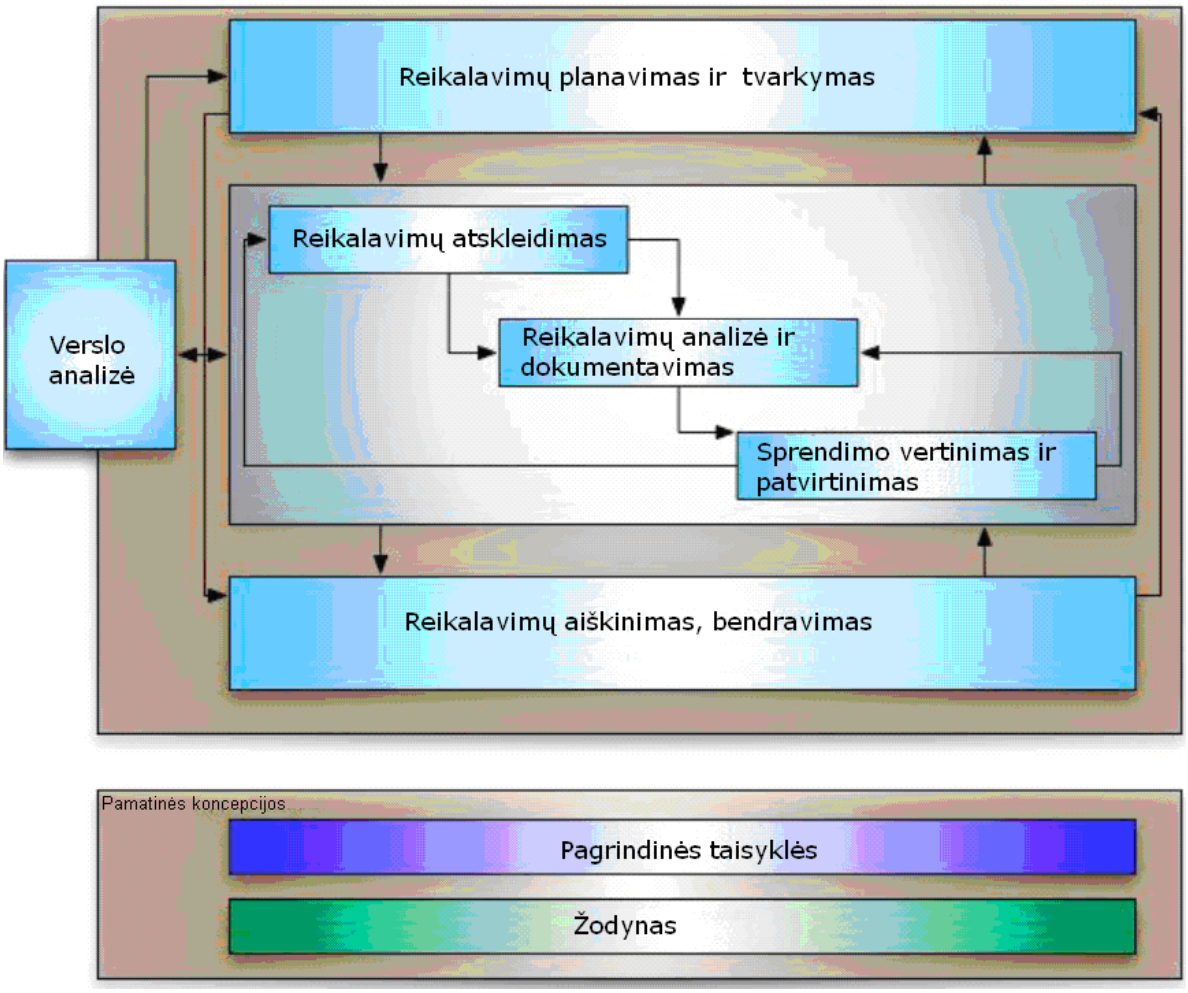
Reikalavimų analitikas atlieka pagrindinį vaidmenį reikalavimų suderinimo ir komunikacijos tarp vidinės projekto komandos (programuotojų, testuotojų ir kt.) narių ir galutinio naudototojo. Reikalavimų atskleidimas ir apibrėžimas yra ilgas ir sudėtingas etapas kuriant programinį sprendimą, todėl XXI a., norint viską atlikti greitai ir kokybiškai, kuo mažesnėmis sąnaudomis ir kuo didesne apimtimi, ieškomos galimybės kuo efektyviau išnaudoti laiką, turimą informaciją, įdirbį ir kiekvieną užduotį atlikti kuo kokybiškiau.

Apibendrintai, verslo analitiko pirminės atsakomybės yra:

- efektyvus reikalavimų rinkimas, atskleidimas;
- aiškus, tikslus reikalavimų išreiškimas (dokumentavimas);
- reikalavimų koordinavimas tarp komandos narių ir kliento.

3.2. Reikalavimų valdymo procesų uždaviniai

Reikalavimų valdymo procesas apima veiklos sferos suvokimą, vartojamą specifinį žodyną, verslo analizę (angl. *Enterprise Analysis*), reikalavimų atskleidimą (angl. *Requirements Elicitation*), dokumentavimą, sprendimo įvertinimą ir patvirtinimą (angl. *Solution Assessment And Validation*), reikalavimų išaiškinimą komandos nariams – tiek kūrėjams, tiek naudotojams (3.2 pav.) (IIBA 2006).



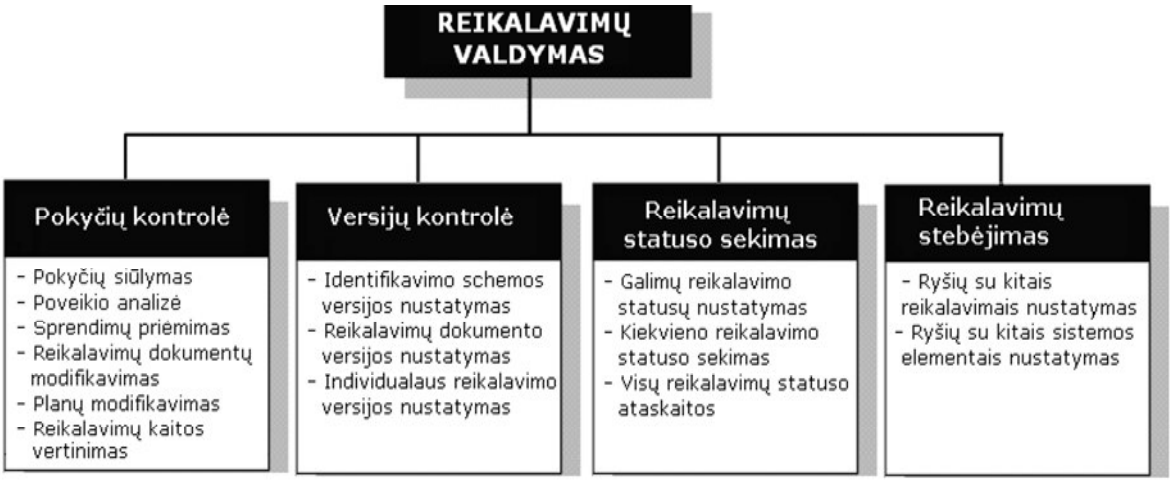
3.2 pav. Reikalavimų valdymo proceso etapai

Projekto kontekste sprendimo reikalavimų kūrimas (atskleidimas, analizė, dokumentavimas, vertinimas) ir kontrolė dar yra vadinama reikalavimų inžinerija (angl. *Requirements Engineering*). Kalbant apie verslo analitiko funkcijas ir projekto reikalavimus, gali būti atskiriamas reikalavimų derinimo ir nustatymo procesas, apimantis informacijos suvokimo, rinkimo, bendravimo su klientais ir komandos nariais technikas, dokumentavimo metodus, ir paties valdymo procesas, sukoncentruotas į turimų reikalavimų, jų versijų kontrolę, pokyčių bei poveikio, gavus RFC,

sekimą, stebėjimą, vertinimą. Kiekviename etape norint atlikti uždavinius kokybiškai ir efektyviai naudojamos įvairios metodikos, reikalaujančios atitinkamų verslo analitiko kompetencijų (IT žinių, logikos, verslo išmanymo, bendravimo ir pan.). Reikalavimams atskleisti svarbu komunikacija ir minčių išgirdimas: skiriami susitikimai, struktūrizuotos peržiūros (angl. *Structured Walkthrough*), analitikai analizuoja turimus praeities duomenis (angl. *Historical Data*), apžvalgas (angl. *Surveys*), taiko „proto šturmo“ (angl. *Brainstorming*) metodą, JAD sesijų (angl. *Joint Application Development Session, JAD*) arba RAD sesijų (angl. *Rapid Application Development Session, RAD*) galimybes, kuria prioritetų matricas (angl. *Development of Priority Matrix*) ir kt. Atliekant analizę ir ruošiant BRD ar kitokią dokumentaciją pasitelkiamas modeliavimas. Struktūrinės analizės (angl. *Structured Analysis*) metu analizuojami verslo procesai (angl. *Business Process*), duomenų srautai (angl. *Data Flow*), darbo srautai (ang. *Workflow*), ERD diagramos, identifikuojama esama būsena (angl. *as-is state*) ir analizuojamos gerinimo galimybės (angl. *to-be state*) (ESI International 2006).

Reikalavimai turi atitikti poreikį, turi būti aiškūs ir vienareikšmiškai suvokiami. Šiam tikslui pasiekti naudojami skirtingi šablonai, vizualizavimo priemonės, įvedami bendri susitarimai (Withall 2007). Reikalavimų atžvilgiu, reikalavimų valdymo užduotys gali būti apibrėžtos (3.3 pav.) (Wiegers 2003) taip:

- pokyčių kontrolė reikalavimams;
- projekto planų pagal esamus reikalavimus palaikymas;
- versijų kontroliavimas ir individualių reikalavimų, ir reikalavimų dokumentų priežiūra;
- reikalavimų statuso projekte sekimas;
- loginių ryšių tarp individualių reikalavimų ir kitų veikiančio projekto produktų valdymas.



3.3 pav. Reikalavimų valdymo uždaviniai

Aplikacijos reikalavimai kuriami elgsenai apibrėžti skirtingais aspektais, t.y. funkcionalumui, pateikimui (ang. *Look & Feel*), veikimui, saugumui ir t.t. Reikalavimai, atsiradę dėl techninių galimybių ar išimčių, koreguoja pradinius reikalavimus, tačiau nežymiai. Techninėms detalėms susitarti, nefunkciniams reikalavimams detalizuoti dažnai pateikiama papildoma dokumentacija. Tačiau, kuriant sprendimą, sunkiausia reikalavimų nustatymo dalis yra apibrėžti klientui ir galutiniam naudotojui reikalingus reikalavimus, atitinkančius funkcionalumą ir darbo organizavimą (vėliau jau koreguojamos tik detalės). Reikalavimus galima skirstyti į daug tipų pagal pasirinktus kriterijus ir smulkumo lygį. Dažnai reikalavimai skirstomi į tipus:

- verslo (angl. *Business*) – apibrėžia tikslus, kaip sistema paveiks veiklos sritį, verslą;
- naudotojų (angl. *User*) – apibūdina naudotojo poreikius ir problemas;
- funkciniai (angl. *Functional*) – apibūdina sistemos elgseną, programinės įrangos darbą;
- kokybės (angl. *Quality of Service*) – taikomi visai sistemai ir yra „galimybės“, pvz., prieinamumas (angl. *Accessibility*), greitaveika (angl. *Performance*), keičiamumas (angl. *Scalability*) ir kt.

Efektyvesniam reikalavimų valdymo procesui svarbus vykdomų procesų suvokimas, naudojami šablonai (metodai, kaip tam tikro tipo reikalavimai), standartai, metodologijos ir palengvinanti atitinkama programinė įranga (pvz., Telelogic DOORS palengvina žvalgą, ryšių tarp reikalavimų radimą, pokyčių stebėjimą ir kt.). Be to, įvairiais aspektais kuriamos ataskaitos esamai situacijai matyti ir vertinti.

3.3. Universalios modeliavimo metodologijos

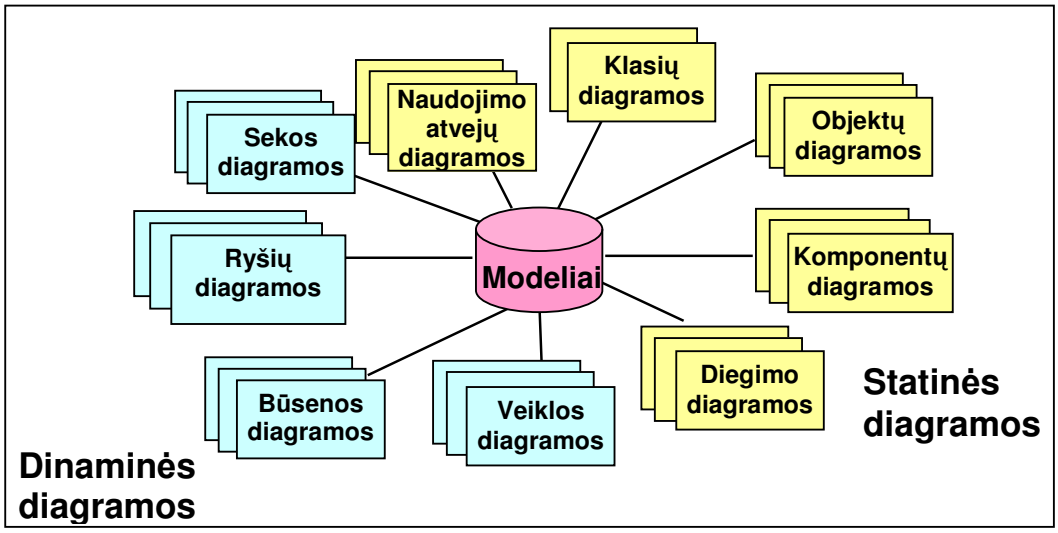
Nepriklausomai nuo to, koks būdas pasirinktas programinio sprendimo kūrimui, kokia pasirinkta procesų valdymo metodika, reikalavimų vaidmuo išlieka nepakitęs, t.y. reikalavimai turi aiškiai apibrėžti, koks turi būti teisingas „produktas“. Reikalavimų rinkimui, sisteminimui ir analizei palengvinti padeda **modeliai** (grafinis realybės supaprastinimas atvaizduojant). Teisingas modelis beveik garantuoja reikalavimų teisingumą. Modeliavimo tikslai (IBM 2003):

- padeda vizualiai atvaizduoti sistemą taip, kaip norima (modeliai palengvina skirtingų atstovų vienodą situacijos supratimą, bendravimą);
- leidžia specifikuoti sistemos struktūrą/elgseną (kartu palengvina susitarimą tarp skirtingų projektų atstovų);
- šablonas sistemos konstravimui (kartu ir testavimui);
- dokumentuoja priimtus sprendimus.

Skirtingiems tikslams ir disciplinoms (etapams) kuriami įvairūs modeliai. Pavyzdžiui, programinės įrangos kontekste žinomi Use Case modeliai (poveikiui išoriniams naudotojams,

sistemos tikslui atvaizduoti), dizaino modeliai (aprašoma, kaip programa realizuos servisus, aprašytus UC), realizacijos modeliai (vaizduoja fizinius programinės įrangos elementus ir juos palaikančios sistemos realizaciją) ir t.t. Be to žinomi reikalavimų modeliai (angl. *Requirements Models*), naudojami programinės įrangos ir verslo sistemų funkcinių bei duomenų reikalavimų atskleidimui, rinkimui ir analizei.

Įvairioms praktikoms naudojama **UML** kalba, kuria skirtingi modeliai kuriami skirtingiems uždaviniams spręsti. Populiarios naudojimo atvejų (angl. *Use Case*), klasių (angl. *Class*), objektų (angl. *Object*), komponentų (angl. *Component*), diegimo (angl. *Deployment*), veiklos (angl. *Activity*), būsenos (angl. *Statechart*), ryšių (angl. *Collaboration*), sekos (angl. *Sequence*) UML diagramos (3.4 pav.) (IBM 2003). Naudojant šiuos modelius, matoma struktūra ir elgsena, skirtingų elementų sąryšiai, dizainas ir realizacija (angl. *Design and Implementation*) išlieka nuoseklūs, pagal poreikį galima išryškinti ar paslėpti reikiamas detales ir vienareikšmiškai interpretuoti situaciją.

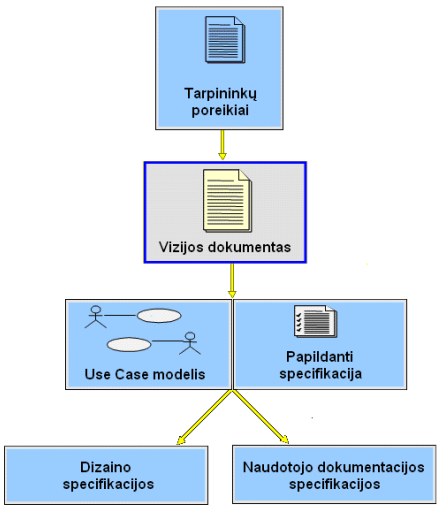


3.4 pav. UML praktikos skirtingiems tikslams pasiekti

Modeliu pagrįsta architektūra (angl. *Model-Driven Architecture*) nurodo, kokie ir kaip paruošti modeliai yra reikalingi. Pats pavadinimas „modeliu pagrįsta“ (angl. *Model-driven*) parodo, kad modeliai naudojami srities supratimui, konstravimui, diegimui, valdymui, palaikymui ir modifikavimui. Sistemos architektūra yra sistemos dalių ir ryšių, taip pat sąveikos tarp dalių per jungtis aprašymas. UML pateikia įrankių, kurie turi būti valdomi ir keičiami, standartą (modeliai ir diagramos, tai nėra kūrimo standartas). Sėkmingam sprendimo įgyvendinimui reikalingas tinkamas procesų valdymo pasirinkimas (pvz., IBM yra siūlomas RUP, geriausios praktikos‘ (angl. *Best Practices*).

Reikalavimų derinimui, analizei, sprendimo vertinimui, testavimo idėjų radimui, testavimo atvejų kūrimui palengvinti gali būti naudojamas veiklos modelis (angl. *Business Model*). Veiklos (verslo) objektų modelis (angl. *Business Object Model*) sukoncentruotas ties verslo esybių, tokių kaip produktas ar pristatoma dalis (angl. *Deliverable*), svarbių veiklos sričiai įvykių (angl. *Events*), veiklos atstovų ir jų atsakomybių, ryšių tarpo verslo objektų išaiškinimu. Yra daug verslo objektų modelio aspektų. Skirtingi požiūriai gali būti atvaizduojami reikiamomis UML diagramomis. Veiklos objektų modelis yra naudingas atramos taškas reikalavimų rinkimo ir analizės darbo sraute, nes identifikuojami verslo objektai ir jų manipuliacija, elgsena. Šie objektai ir procesai galiausiai atsiduria dokumentuotuose sistemos reikalavimuose.

Kiekvieno naujo projekto ar net mažiausio funkcionalumo pradžia – užduoties suvokimas. Iš pradžių turimi reikalavimai yra poreikio pavidalu (angl. *Stakeholders Requests*). Pradinius poreikius formuluoja klientas, analitikas juos gauna skirtingais būdais (elektroniniu paštu e-mail, susirinkimų metu, pokalbių metu, BRD ar pan.). Sužinotas kliento poreikis inicijuoja programinės įrangos funkcijas ir savybes, iš pradžių sukuriama vizija, pradinis dizainas, orientuojantis į galutinį naudotoją galimas prototipų panaudojimas (Stappers 2009, Lin Chyun-Chau, Luh Ding-Bang 2009). Susidarius viziją, diskutuojama, derinami konkretesni reikalavimai. Šiame etape dėmesys yra sutelktas į problemos apibūdinimą, sistemos tikslų nustatymą, naudotojų analizę, poreikių supratimą, aplinkos susitarimą (angl. *User environments and platforms*), produkto savybių suderinimą (angl. *Product Features*). Turint atramos tašką, tolesniam darbui kuriami UC (3.5 pav.) (IBM 2003).



3.5 pav. Apibendrinta reikalavimų apibrėžimo struktūra, naudojant reikalavimų kaupimo įrankius

Konkretizuoto Use Case tikslai:

- nustatyti dalyvius (angl. *Review and Refine Actors*);
- nustatyti ir trumpai abibūdinti naudojimo atvejus (angl. *Discover and briefly describe Use Cases*);
- identifikuoti sąryšius (angl. *Identify communicates-associations*);
- sukurti UC diagramas (angl. *Create a use-case diagram*).

Use Case modelio nauda:

- bendravimas (angl. *Communication*) – bendravimas lengvesnis, konkretesnis, pateikiamas vizualiai ir kartu pradedama dokumentacija;
- identifikavimas (angl. *Identification*) – nustatomi dalyviai;
- vertinimas (angl. *Verification*).

Use Case – naudojimo atvejis. Tai yra viena pamatinių sąvokų su reikalavimais susijusių klausimų kontekste. Kitaip tariant, tai sąveikos tarp sistemos ir vieno ar daugiau veikėjų (angl. *Actor*) aprašymas. Šis specifikavimo būdas yra grįstas naudojimu (angl. *Usage-centric*), todėl dažnai palengvina bedradarbiavimą su verslo naudotojais, kuriems toks formatas lengviau suvokiamas. Naudojimo atvejis gali būti apibendrintai parodytas modeliu ir/arba aprašytas. Karlo Wiegerso pateiktas *Use Case* aprašymo šablonas pavaizduotas 3.6 pav.

Use Case ID:	
Use Case Name:	
Created By:	Last Updated By:
Date Created:	Date Last Updated:

Actors:
Description:
Trigger:
Preconditions:
Postconditions:
Normal Flow:
Alternative Flows:
Exceptions:
Includes:
Priority:
Frequency of Use:
Business Rules:
Special Requirements:
Assumptions:
Notes and Issues:

3.6 pav. Use Case dokumentavimo šablonas (Wiegers 2006)

Naudojantis šablonu atvejui priskiriamas unikalus identifikavimo numeris *Use Case ID*, aprašomas pavadinimas, sukūrimo ir paskutinio atnaujinimo data ir autorius. Ši informacija leidžia **matyti reikalavimų apimtį, sekti jų atsiradimą ir keitimą**. Šablone aprašomas funkcionalumas, veikėjai, trigeriai, pradinės sąlygos, normali eiga, alternatyvūs atvejai, specialieji reikalavimai, verslo taisyklės, pastabos, išimtys, jei yra, ir prielaidos. Prioritetas parodo svarbą (ypač aktualu planuojant). Pasirinkus šį reikalavimų aprašymo metodą, analogiškai dokumentuojamas ir papildomas funkcionalumas – *Extended Use Case*. Aprašomus pagrindinės eigos ir alternatyvius atvejus patogu numeruoti ir įterpti stulpelį veikėjui, taip kiekvienam etapui matomas vykdytojas (asmuo ar sistema). Šablonas patogus bendraujant su verslo klientais. Jis lengvai suprantamas, tačiau įgyvendinimo atžvilgiu toks dokumentavimo būdas nėra pakankamai detalus. Konkretizuojant veiksmus, kartu yra pateikiamos įvairios UML diagramos, papildoma specifikacija funkcionalumui detalizuoti. Prie *Use Case* aprašymo patogu turėti grafinės naudotojo sąsajos (angl. *Graphical User Interface*, GUI), naudojamos veiksmui įvykdyti, vaizdą ir aprašymą. GUI pavyzdinis vaizdas ir elementų lentelė yra lengvai suvokiamas pateikimo būdas verslo vartotujui ir vienareikšmis įvykdymo nurodymas (tai gali būti pakeista prototipu). Vertinant reikiamą laiką ir analizuojant situaciją, šis modelis nėra pakankamai informatyvus verslo analitikui ir projekto vadovui, kurie turi patys vertinti remdamiesi savo metodika ir patirtimi. Naudojant Use Case (ir, prireikus, papildančią specifikaciją) galima aiškiai išskirti vykstančius atvejus, t.y. atvejus, kuriuos turi sugebėti palaikyti ir apdoroti kuriama aplikacija.

Use Case modeliu ir aprašu išaiškinami ir dokumentuojami atskiri reikalavimai. Visgi funkcionalumą (projektą ar dalį) apibūdina atskirų tarpusavyje suderintų reikalavimų visuma. Dokumentacijai paruošti, pateikiami įvairūs dokumentų šablonai (angl. *Software Requirements Template*) su paruoštų skyrių sąrašu (turinys, dokumento paskirtis, bendri reikalavimai, bendri teiginiai, naudojamos terminologijos paaiškinimas, vieta modelio, lentelių įterpimui ir pan.). Dar labiau palengvinti matomumą, sąryšių matymą, bendradarbiavimą padeda specialios programos (pvz., Telelogic DOORS). Žiūrint projekto lygmenyje, verslo analitiko vykdomi procesai apima visą reikalavimų ciklą (nuo atradimo iki dokumentacijos ir aiškinimo), procesai lengviau valdomi, kai elgsena struktūrizuota, darbui naudojama patirtis ir patariantys šablonai.

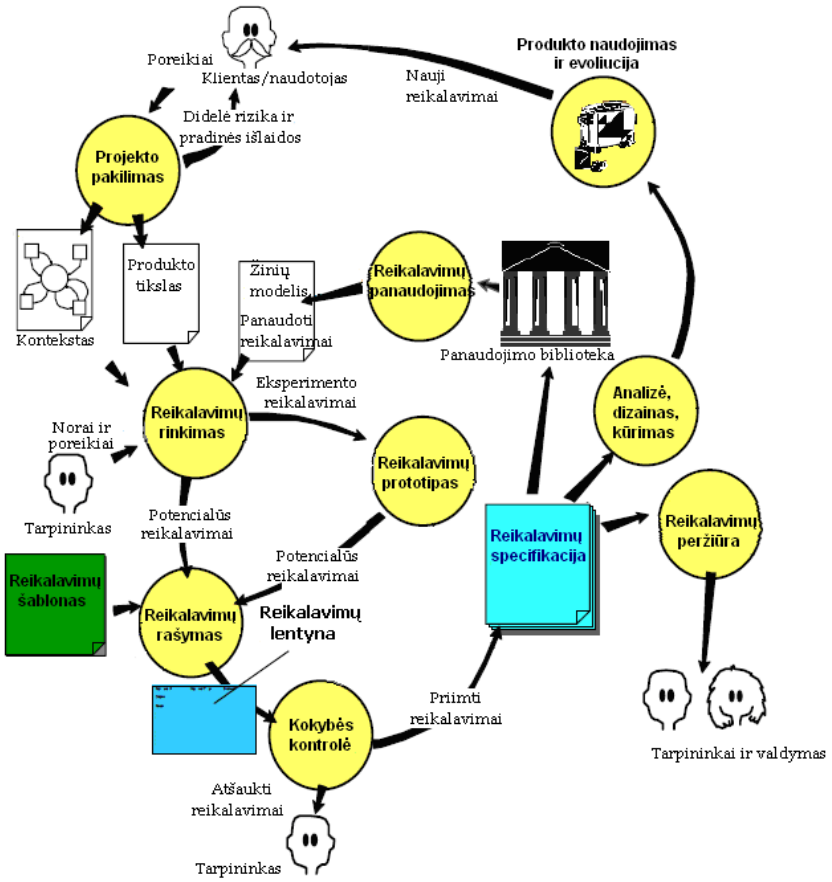
Suzanne ir James Robertsonai darbui su reikalavimų ciklu siūlo susitemintą **Volere** šabloną. Pavadinimas Volere yra iš itališko veiksmažodžio „norėti“. Šis šablonas pateikiamas kaip formalus „gidas“, padedantis, surinkti, suprasti, dokumentuoti ir pateikti reikalavimus komandos nariams, nieko nepraleidžiant, o taip pat ir valdyti tuos reikalavimus. Volere padeda visuose etapuose prisiminti, į ką turėtų būti atkreiptas dėmesys, kokie aspektai įvertinti ir kokios galimos technikos.

Šis šablonas yra tarsi žinomų praktikų, kurias pravartu prisiminti, susisteminimas. Konkrečiam sprendimui reikia pasirinkti reikiamas galimybes ir sukurti sau tinkantį darbo modelį. Atskiri reikalavimai yra saugomi taip vadinamose „reikalavimų lentynose“ (angl. *Requirements Schell*), kur surenkama informacija apie reikalavimą. Galimas ir visas Volere dokumento šablonas. Panaudojant dokumento skyrius jis tampa ruošiniu dokumentacijai.

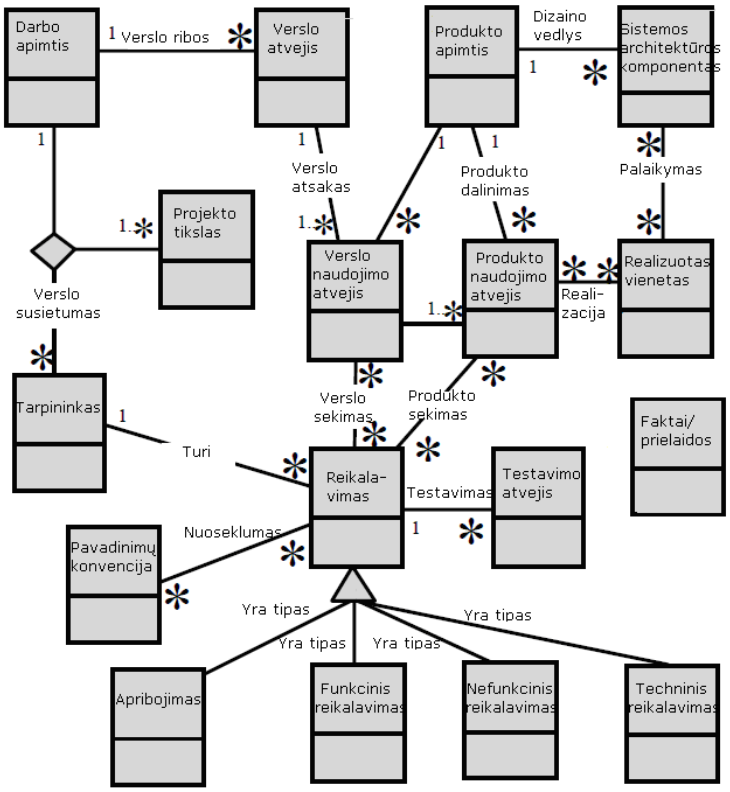
Apibendrintai žiūrint, šablonas apima trijų tarpusavyje sąveikaujančių komponentų aprašymą (The Atlantic Systems Guild 2009):

- reikalavimų žinių struktūra (angl. *The Requirements Knowledge Structure*) – apie skirtingų darbo su reikalavimais etapų saryšį, kaip sekami reikalavimai skirtinguose lygmenyse;
- reikalavimų procesas (angl. *Requirements Process*) – skirtas užduotims ir vykdomoms procedūroms, reikalingoms reikalavimams atskleisti, įgyvendinti (įtraukti į produktą), paskleisti, pristatyti;
- reikalavimų tarpininkai (angl. *Requirements Stakeholders*) – tai įvestis, nusakanti, kaip reikalavimų žinių struktūra turi būti pritaikoma projektui. Informacija iš tarpininkų (užsakovų, klientų ir pan.) naudojama reikiamam darbui nustatyti ir poreikiams išsiaiškinti.

Kiekvienas komponentas susideda iš jį sudarančių dalių. Reikalavimų žinių struktūrai priklauso sistemos reikalavimams atskleisti naudojamas reikalavimų žinių modelis, reikalavimų šablonas, individualių reikalavimų (angl. Atomic Requirement) struktūra, reikalavimų sekimas (angl. *Requirements Traceability*), reikalavimų lygiai. Reikalavimų procesui priklauso bendras procesas reikalavimų žinioms atskleisti, kaip išrinkti projekto charakteristikas, reikalavimų atradimo, rinkimo technikų patarimai, tikslo analizės technikos, reikalavimų vertinimas (angl. *Estimation*), reikalavimų specifikacijos audito technikos, prioriteto nustatymas. Darbui su reikalavimų tarpininkais priklauso tarpininko, jo vaidmenys (dar vadinamos rolės), žinių, atsakomybių analizė, grįžtamojo ryšio gavimo technikos, konfliktų valdymas, reikalavimų požiūrio supratimas. Taigi apibendrintai šablonas apima visą reikalavimų ciklą (3.7, 3.8 pav.) (Robertson 2006).

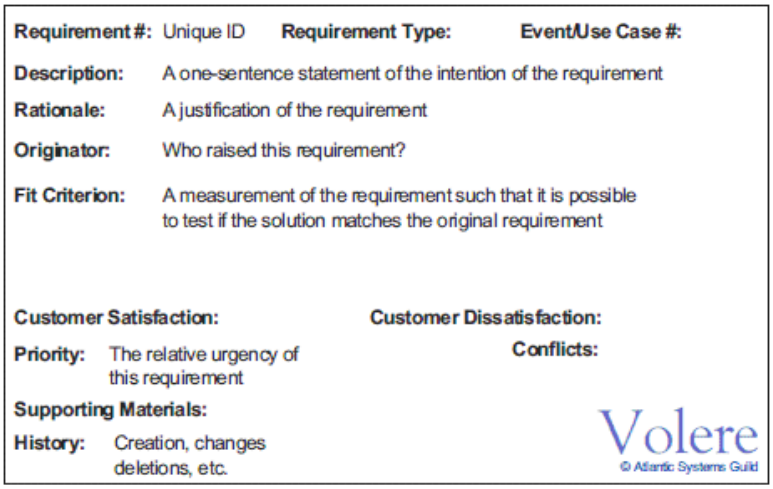


3.7 pav. Volere šablono modelis



3.8 pav. Volere principo modelis (angl. *Volere Requirements knowledge model*)

Žiūrint reikalavimo atskleidimo aspektu, Volere remiasi naudojimu grįstu (angl. *Usage-centric*) požiūriu į reikalavimus ir apibūdina visą reikalavimo ciklą nuo jo atskleidimo iki dokumentavimo ir daugkartinio panaudojimo (angl. *Reuse*). Use Case diagramos (grafinis produkto funkcionalumo atvaizdavimas) arba, jei daug UC, Use Case sąrašas (angl. *Use Case List*) „gaunami“ atskleidžiant funkcinius reikalavimus ir bendravimo su klientu ir galutiniais naudotojais, naudojant įvairius pagalbinius „įrankius“, pvz. pokalbių pastabas (angl. *Interview notes*), aprašus, video ar audio įrašus, sekos diagramas (angl. *Sequence diagram*), procesų modelius (angl. *Process model*), prototipus (angl. *Prototype*), GUI pavyzdžius (angl. *Mock-ups*). Individualūs reikalavimai kaupiami vadinamose lentynose „*Requirements Shell“*. Kiekvieną reikalavimą apibūdina išskirtos pagrindinės charakteristikos: identifikavimo numeris, reikalavimo tipas (funkcinis, nefunkcinis, apribojimas, technologinis), UC, kuriam būdingas reikalavimas, numeris, aprašymas, priežastis, reikalavimo kūrėjas, tikimo kriterijus (angl. *Fit Criterion*), t.y. įvertinimas ar testuojamas, suprantamas, įvertinimas, ar klientas patenkinas/nepatenkintas, prioritetas, konfliktai (jei reikalavimas prieštarauja kitiems), papildantys dokumentai, kur pateikiama daugiau informacijos, kūrimo, keitimo data (3.9 pav.) (Robertson 2006).



3.9 pav. „Reikalavimo lentynos“ originalo pavyzdys

3.4. BPM metodologija

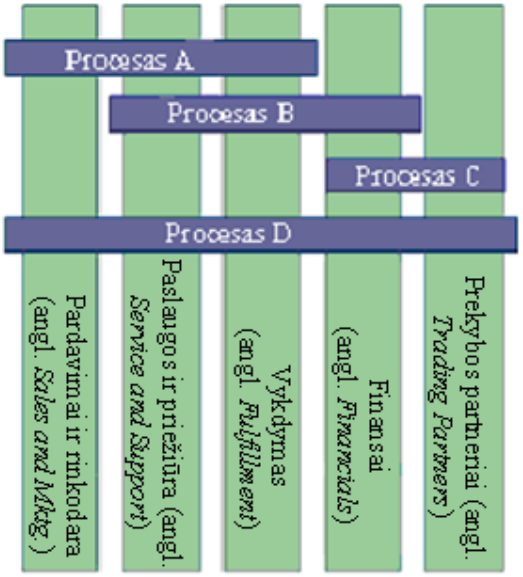
Jau keletą pastarųjų metų veiklos procesų valdymo (angl. *Business Process Management*, BPM) įvaldymas ir tobulinimas ne vienai kompanijai yra prioritetas. BPM privalumas tas, kad metodologija įdedant mažiau pastangų įgalina apdoroti procesus (angl. *Process*) daugiau ir yra geresnė kokybė. Pagrindinė nauda: našumas (angl. *Efficiency*), efektyvumas, judrumas (angl. *Agility*) (3.1 lentelė).

3.1 lentelė. Produktyvumo gerinimas įvairiose projekto fazėse (Lombardi 2008)

Tipinės BPM projekto fazės	% projekte	Produktyvumo pagerinimas
Funkciniai reikalavimai ir funkcinis dizainas	25%	50%
Realizacija (angl. *Implementation*)	50%	20-25%
Kokybės užtikrinimas / testavimas	25%	30%
Verslo atskleidimas	N/A	N/A

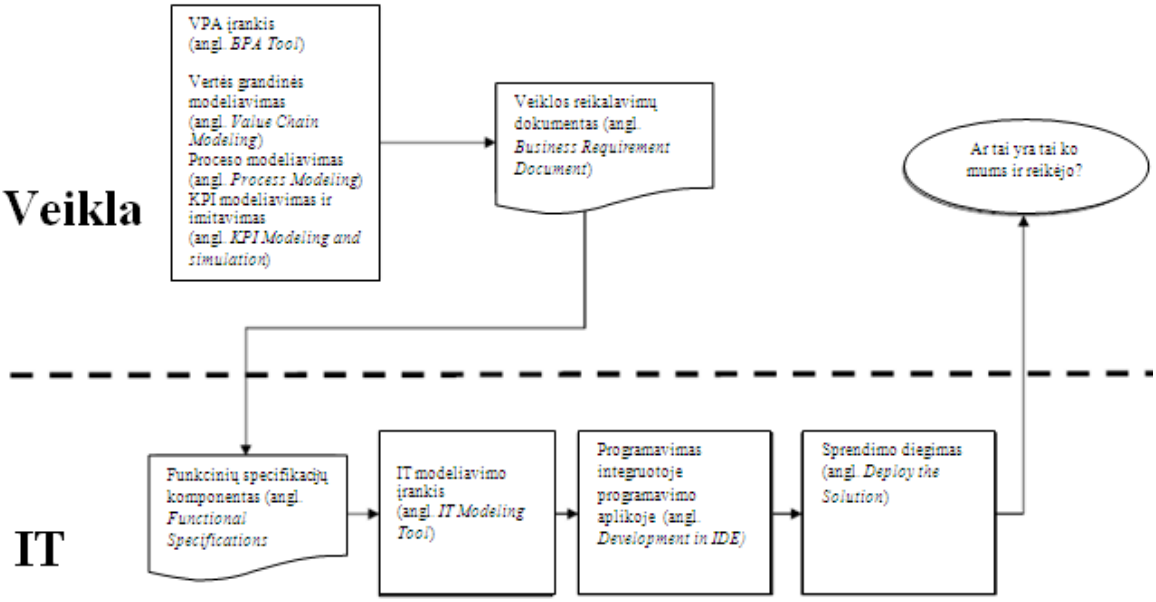
Veiklos procesų valdymas, kaip valdymo disciplina, atsirado prieš 20 metų kaip naujas požiūris į veiklą (kaip planuoti, analizuoti, vertinti ne atskiras dalis, bet procesus visame veiklos kontekste) (Silver 2007). BPM – plati sąvoka, susijusi ir su veiklos modeliavimu, rodiklių vertinimu, ir su technologija, ir su verslo ir IT bendradarbiavimu, t. y. ir veiklos analizė, ir paskirstymas, ir integracija, ir rodiklių vertinimas, *KPI* (angl. *Key Performance Indicators*), ir darbo srautų (angl. *Workflow*) automatizavimas, ir į paslaugas orientuota architektūra (angl. *Service-oriented Architecture*, SOA), ir kt. BPM esmė – naujas požiūris į veiklą, kai atskirai analizuojama ne atskiros veiklos, duodančios rezultatą, bet veikiantys keliose veiklose procesai, kurie leidžia susikoncentruoti ties procesu nuo jo pradžios iki pabaigos (angl. *End-to-End*). Daug privalumų jau žinoma naudojant verslui pritaikytas aplikacijas, pvz., ERP, CRM, kurios individualių funkcijų ar

sistemų lygmenyje automatizuoja veiklą, daro ją efektyvesnę, suteikia galimybę duomenų integracijai ir kontrolei. Be to, tai ištakos besikuriačio *End-to-End* procesų valdymo, kuriam vykstant dėmesys kreipiamas į galutinį tikslą (angl. *Mission-Critical*) ir naikinamos ribos tarp atskirų funkcijų (3.10 pav.).

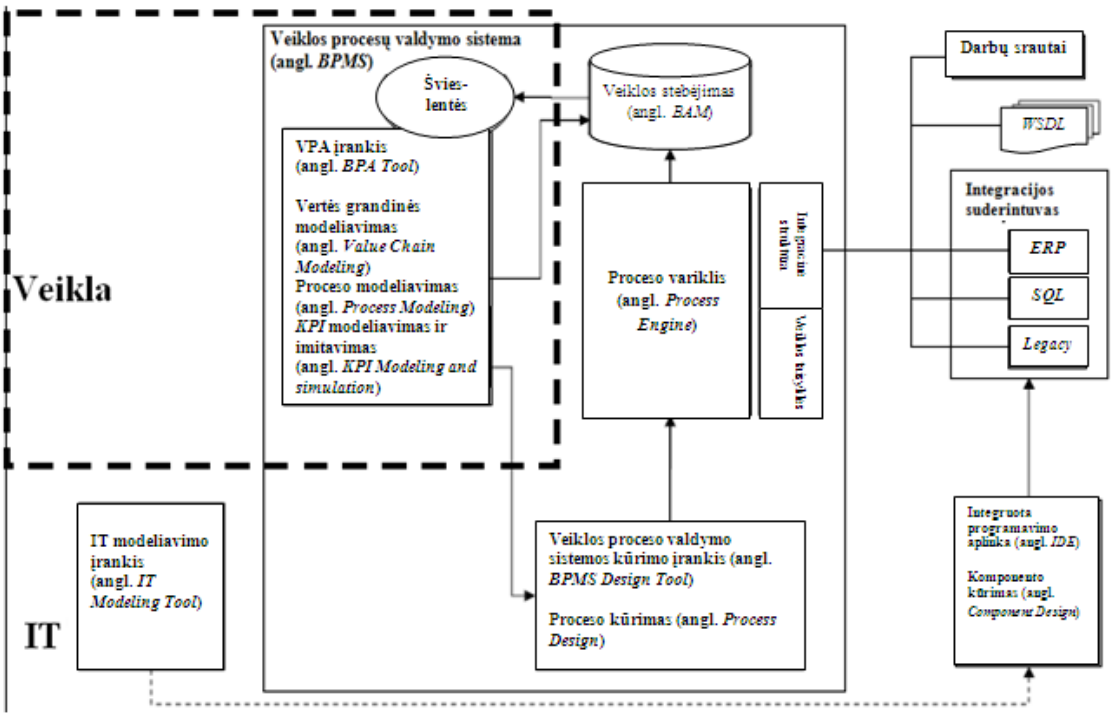


3.10 pav. Kompanijos ir verslo aplikacijos, organizuojamos pagal atskiras veiklos funkcijas, kai dėmesys sutelktas į tikslą, procesai naikina ribas (Silver 2006)

BPM disciplina reikalinga ne tik dėl kitokios veiklos analizės galimybės. Ši naujo požiūrio metodika leidžia, akcentuojant modeliavimą ir analizę, paversti modelį vykdomuoju (angl. *Executable*), matuojamu ir galimu optimizuoti. Tam tikslui naudojamas veiklos procesų valdymo komplektas – *BPMS*, kuris naikina barjerus tarp atskirų funkcijų ir padidina efektyvumą (automatizuojant), paslankumą pokyčiams, suderinamumą (logika pagrįsta taisyklėmis) ir aiškų proceso visumos, t. y. viso funkcijų derinio, matomumą. Taip pat keičiasi visas darbo organizavimas. BPMS visą veiklos procesą padaro vientisą, paremtą veiklos specialistų ir IT specialistų bendradarbiavimu (3.11 pav., 3.12 pav.).



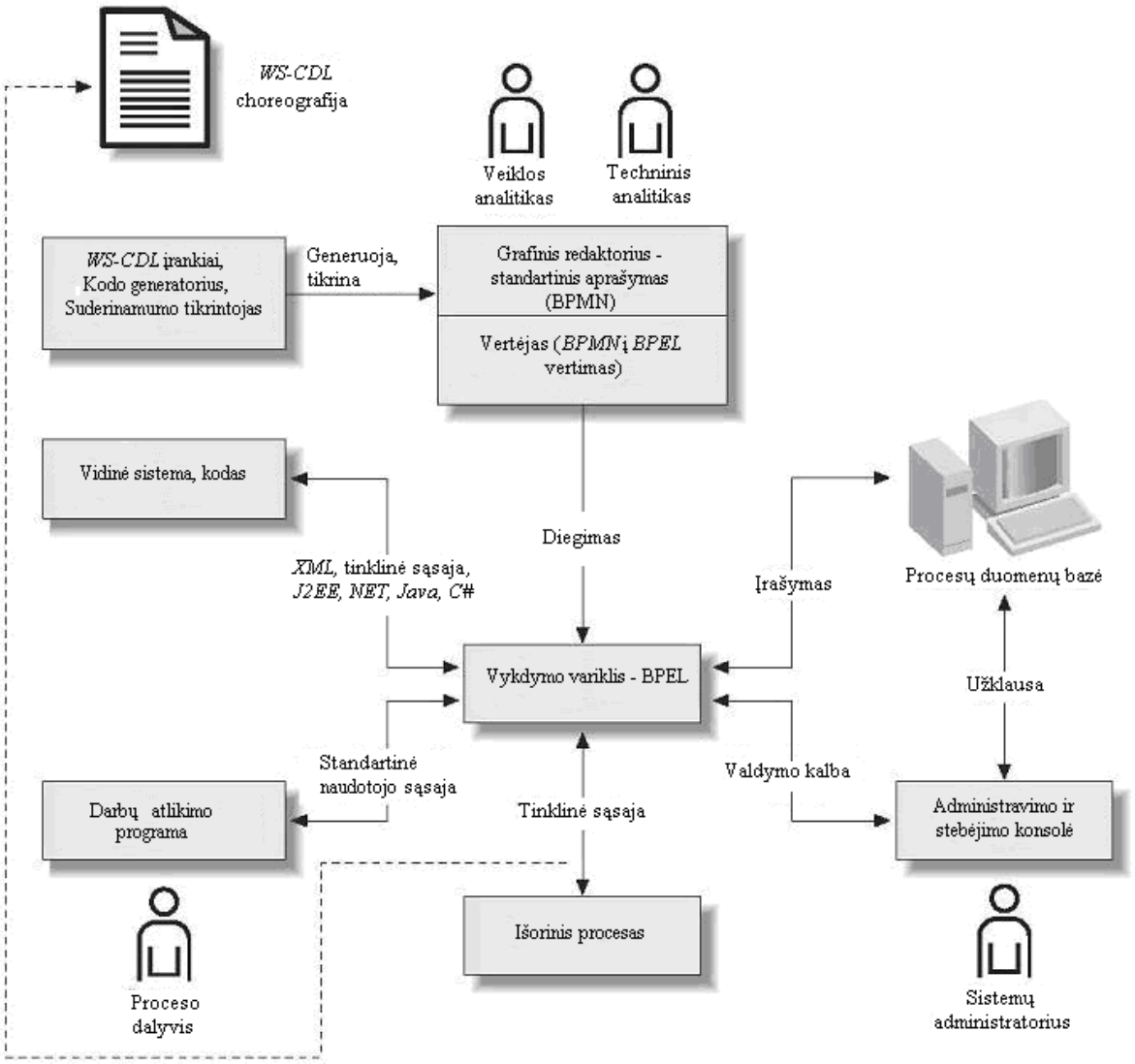
3.11 pav. Seno požiūrio *BPM* įgyvendinimas (Silver 2006)



3.12 pav. Naujo požiūrio *BPM* įgyvendinimas – *Clean Handoff* stilius (Silver 2006)

BPM turi daug žinomų standartų, iš kurių renkamasi geriausios kombinacijos. Dažnai veiklos procesų modeliavimo standartų grupės analizė pradedama veiklos procesų modeliavimo pastabomis BPMN (angl. *Business Process Modeling Notation*), kurių pagrindinis siekis yra suteikti ženklų sistemą, kurią lengvai suprastų visi procese dalyvaujantys žmonės: ir verslo analitikai, kuriantys juodraštinius veiklos procesų modelius, ir techniniai darbuotojai, kurie atsakingi už

veiklos procesų vykdymą, reikiamų technologijų diegimą, ir galiausiai tie atstovai, kurie valdys ir prižiūrės šiuos procesus. Veiklos procesų diagramų braižymas yra loginė veiklos procesų valdymo įgyvendinimo pradžia. BPMN vaizduoja nuoseklų veiklos proceso etapų ir veiksmų srautą (3.13 pav.) (Havey 2005).



3.13 pav. *BPM* architektūra

BPMN yra susieta su veiklos procesų vykdomąja kalba BPEL4WS arba tiesiog BPEL (angl. *Business Process Execution Language*) ir yra vykdoma BPEL variklio. BPEL, kaip procesų vykdomoji kalba, suformuoja esminį veiklos procesų valdymo sluoksnį. Egzistuoja ir kitos panašaus pobūdžio vykdomosios kalbos, pavyzdžiui, XPDL, BPML ir kt., tačiau šiuo metu labiausiai dėmesys atkreiptas į didžiųjų gamintojų naudojamą BPEL, kuri, prireikus gali būti modifikuojama, derinant ją prie kitų kalbų ar perimant kai kurias reikiamas jų savybes.

Tinklinių paslaugų choreografija (angl. *Choreography*) paremta W3C (angl. *World Wide Web Consortium*) konsorciumo WS-CDL (angl. *Web Services Choreography Description Language*) kalba. Choreografija apibūdina procesus globaliu požiūriu. Choreografijos globalus požiūris yra kontrastas BPEL orkestravimui (angl. *Orchestration*). BPEL procesas yra vieno dalyvio procesas, o choreografija yra dalyvių grupės sąveikos modelis.

WSDL (angl. *Web Services Description Language*) arba tinklinių sąsajų apibūdinimo kalbą pristatė Microsoft kartu su IBM 2000 m. spalio mėn.. Ji buvo paremta IBM tinklu pasiekiamų paslaugų aprašymo kalba ir Microsoft paprastų objektų pasiekimo protokolų (SOAP) sutartine kalba. WSDL apibūdina verslo funkcijas nuo diegimo nepriklausomu būdu, kuris leidžia kitoms taikomosioms programoms jas surasti ir panaudoti. Taigi WSDL yra kalba, pagal kurią nusakomos tinklinės sąsajos ir aprašomas jų naudojimas. Programa, naudojanti tinklines sąsajas, remiasi WSDL tam, kad surastų paslaugų vietą ir taip būtų galima prisijungti prie jų. SOAP pristatė Microsoft korporacija 1999 m., dalyvaujant IBM tam, kad būtų praplėstas pasiūlymo mastas. 2000 m. kovo mėn. abi kompanijos pateikė 1.1 versiją konsorciumui – W3C. Šiuo jungtiniu produktu labai pasitiki pramonės šakos, nes jis suteikė galimybių dirbti kartu su dviem šių organizacijų labiausiai paplitusiais kompiuteriniais modeliais, tokiais kaip J2EE ir NET.

BPEL specifikacija yra laikoma jau esamų tinklinių sąsajų standartų veiklos procesų plėtiniu. Anksčiau tinklinės sąsajos buvo apribotos iki neturinčių būklės sąveikų. BPEL, kiti procesai ir choreografijos kalbos rodo, kaip sukurti turinčius būklę, sąveikaujančius veiklos procesus tinklinėms sąsajoms. BPEL yra griežta kalba, kuri sukuria ir išplečia žiniatinklio paslaugas sąveikaujantiems procesams.

BPMN ryšys su BPEL – BPMN pateikia mechanizmą, kaip iš veiklos lygio pastabos sukurti vykdomą veiklos procesą. **BPMN diagramos nebūtų tiek naudingos, jei jų nebūtų galima vykdyti**. BPEL sąsaja su BPMN leidžia iš BPMN diagramų gauti BPEL kodą, o tai savo ruožtu suteikia galimybę vykdyti BPMN BPEL sistemose.

Taigi visa veiklos procesų valdymo metodikos sąvoka apimama ir veiklos proceso modeliavimą/dizainą, ir vykdymą, ir kontrolę/stebėseną. Pradinis etapas yra skirtas **reikalavimams nustatyti ir apibrėžti** modeliuojant veiklos ar verslo procesus. **Veiklos proceso modeliavimas** (angl. *Business Process Modeling)* **–** modelių kūrimo ir lyginimo modeliai, kaip yra dabar (angl. *As-Is*) ir kaip būtų (angl. *What-If*) ateityje išanalizavus ir patobulinus. Modeliavimo įrankiais galima įvertinti esamą situaciją (angl. *As-Is Process*) ir nustatyti patobulintą (angl. *To-Be Process*), šias situacijas įvertinti be techninių realizacijos detalių, naudojantis tik modeliavimo įrankiu. Šiame etape modeliuojama įvykių seka, svarbus reikalavimas, kad kiekvienas procesas turėtų trukmę, aiškią įvestį (angl. *Input*) ir rezultatą (angl. *Output*). Veiklos proceso modeliavimo pagrindiniai terminai (Havey 2005):

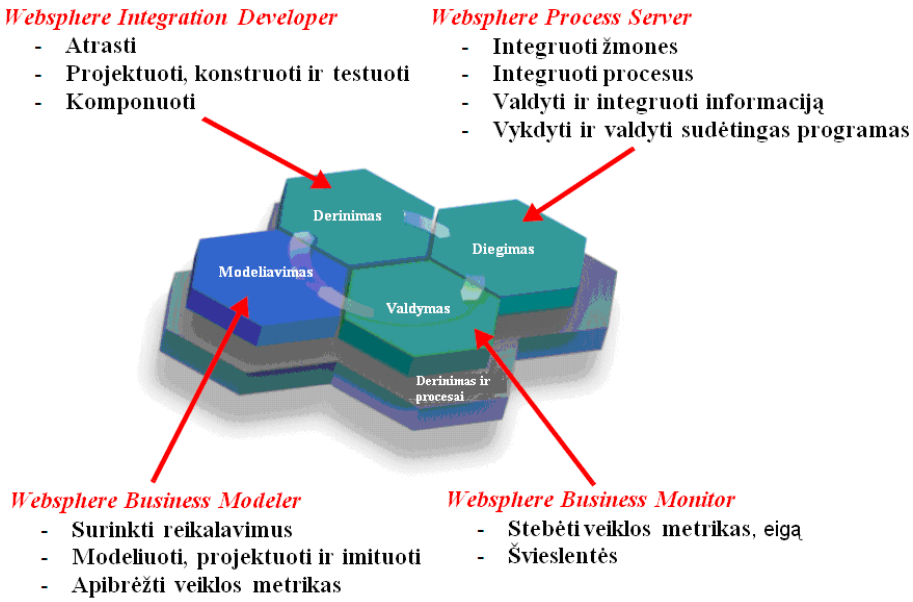
- proceso apibūdinimas (angl. *Process Definition)* – pagrindininis proceso elgesio algoritmas;
- proceso atvejis *(*angl. *Process Instance)*, turintis specifinę įvestį (*input*);
- veiksmas ar užduotis (angl. *Activity Or Task*) – proceso etapas;
- proceso etapas (angl. *Automated Activity Or Automated Task*), kurį automatiškai atlieka vykdymo variklis;
- proceso etapas *(*angl. *Manual Activity Or Manual Task)*, kurį atlieka dalyvaujantis procese asmuo.

Veiklos procesų valdymui skirtingi gamintojai siūlo savo produkciją. BPM sprendimus siūlo ir labai žymūs gamintojai, pvz., *IBM WebSphere BPM Suite* sprendimas, *Oracle BPEL Process manager*, ir mažiau girdėti, pvz., *BONAPART® BPM Suite*.

IBM yra užėmusi pagrindinę poziciją veiklos procesų valdymo sistemoje (BPMS) prekiautojų horizonte: tai pirmas programinės įrangos infrastruktūros tiekėjas, siūlantis **pilną BPMS**, palaikantis pilną (angl. *End-to-End*) egzistavimo ciklą nuo analitinio modeliavimo iki vykdymo valdymo (angl. *Performance Management*) ir optimizavimo, remiantis BPEL ir servisui orientiuota architektūra (*SOA*). *WebSphere BPM* pabrėžia kritiškos misijos intensyvios integracijos procesus bankininkystės ir finansų servisuose, draudime, telekomunikacijose ir pan.

WebSphere BPM Suite, palaikantis nuosekliai tęstinį veiklos proceso gyvavimo ciklą nuo pradžios iki pabaigos (3.14 pav.), susideda iš keturių pagrindinių komponentų (Wahli, et. al. 2006):

- *WebSphere Business Modeler Version 6;*
- *WebSphere Integration Developer Version 6;*
- *WebSphere Process Server Version 6;*
- *WebSphere Business Monitor Version 6.*



3.14 pav. IBM WebSphere Business Suite komponentai ir paskirtis

Pirmam etapui naudojamas *WebSphere Business Modeler.* Įrankis skirtas: analitiniam proceso eigos modeliavimui, išteklių, išlaidų, duomenų ir vykdymo (angl. *Performance*) valdymui; proceso scenarijų simuliavimui; dokumentavimui ir eksportavimui į proceso *Design*‘o įrankį. Savybė *Business Measures Editor* apibrėžia vykdymo indikatorius (*KPI*) ir kitas proceso metrikas, naudojamas vykdymo valdyme (angl. *Performance Managenent*). IBM *Websphere Business Modeler* – patogus ir verslo eigos aprašymui. Tai privalumas prieš, pvz, *Active BPEL Designer* produktą, kuris reikalauja modeliavimo etape nustatyti atskirų veiksmų tipus, kaip pvz., *Invoke* („Iškviesti“), *Receive* („Gauti“) ir pan., tai atitraukia dėmesį nuo pačios eigos logikos aprašymo, įrankis tampa nepatogus, jei naudojamas tiesiog analizei. Kiekvienam procesui gali būti priskiriamas išteklius, vaidmuo (angl. *Role*), darbo laikas, tvarkaraštis, išlaidos. Modelis lengvai koreguojamas, galimas sekos ar proceso daugkartinis panaudojimas. Sukurtas modelis yra „vykdomasis“ (angl. *Executable*), t.y. galima „simuliuoti“ (angl. *Simulate*) veiklą ir matyti rezultatą. Ši savybė patogi sprendimo vertinimui ir verslo analitikui ieškant gerinimo galimybių, *to-be* modelių. Be to, lengvai generuojamos ataskaitos reikiamu aspektu. BPM‘e naudojama BPEL kalba lengvai generuojama į UML. Nors pakeitimai greitesni, mažesnis dėmesys pokyčių sekimui ir kaitos registravimui, kas yra reikalinga reikalavimų valdymui. Naudojantis BPMS pirmame etape sukurtas modelis toliau naudojamas realizacijai, kitaip nei *Use Case*, kur dokumentaviams ir realizacija yra atskiri procesai.

Pasirinkus bet kurį kūrimo modelį, reikalavimų pateikimo formą verslo analitiko tikslai ir užduotys išlieka. Atskleidus ir išanalizavus reikalavimus, priklausomai nuo metodologijos keičiasi komandos sudėtis – analitikas, dirbantis atskirai nuo programuotojų, ar analitikas, pateikiantis

programuotojams paruoštą realizacijos pradžią reikalavimus jau su realizacijos įdirbiu (tokiu atveju analitikas dirba su mažesne programuotojų komanda, nes jis pats atlieka dalį realizacijos). Keičiasi reikalavimų pateikimas, atributai gali būti priskirti tiesiai modeliui, taip išvengiant kai kurių reikalavimų atskiros dokumentacijos. Informacija viename BPMN modelyje apima skirtingomis UML diagramomis atvaizduojamus aspektus. BPM naudojama BPEL gali būti lengvai konvertuojama į UML, BPEL modeliai, pateikti vykdančiai komandai, naudojami tolesniam kūrimui taip apjungiant darbą į visumą ir keičiant universalią iki tol žinomą koncepciją, kad reikalavimai ir vykdymas yra atskiri etapai.

3.5. Išvados

Kompanijos neturi kito pasirinkimo kaip susitvarkyti su vis sudėtingesniais procesais ir augančiais bei greitai kintančiais poreikiais. Efektyvumui ir kontrolei pasiekti naudojamos žinomos technikos, siekiama atitikti pripažintus standartus, tuo pat metu ieškoma naujų galimybių. Veiklos procesų valdymo paradigma pristato kitokį požiūrį į procesus, ypač daug dėmesio skiriama modeliavimui, vykdomųjų modelių naudojimui, procesų tęstinumui. Nors galutiniai reikalavimų tikslai išlieka tie patys (reikalavimus reikia atskleisti, analizuoti, vertinti, laiku ir vienareikšmiai pateikti ir dokumentuoti), reikalavimų kaupimas su Use Case ir UML modeliais bei BPM veiklos modeliu skiriasi. BPM modelis plačiau panaudojamas, informatyvesnis. Visgi labiausiai nauda matoma taikant visą BPM ciklą, ne tik modeliavimą. Reikalavimų dokumentavimas su Use Case išlieka patogus, kai produktas kuriamas konkrečiam stambiam klientui. Jiems naudojimo atvejais pateikiami reikalavimai dažnai vis dar lieka labiau įprasta forma, verslo naudotojai dažnai nori prie modelių turėti daugiau dokumentacijos.

4. Reikalavimų valdymas naudojant programinius sprendimus

4.1. Įvadas

Viena iš pagrindinių sričių, kur IT taikymas labai svarbus, tobulinami sprendimai yra draudimas (tiek gyvybės, tiek P&C). Draudimo kompanijoms programiniai sprendimai ne tik padeda taupyti laiką, matyti patogias ataskaitas, bet ir padeda parduodant draudimo polisus ar registruojant draudžiamuosius įvykius – Claim‘us. Kuriant sprendimą labai daug dėmesio skiriama verslo logikai, taisyklių ir išimčių taikymui, apribojimų tikrinimui ir pan. Norint pasiekti gero rezultato akivaizdus tinkamų, tikslių ir laiku pateiktų reikalavimų vaidmuo. Reikalavimams apibrėžti, saugoti, valdyti galimas ne vienas programinis sprendimas.

Analizei pasirinktas P&C draudimo Claim komponento reikalavimas – draudžiamojo įvykio (Claim‘o) registracija sistemoje ir užklausos perdavimas kitai sistemai. Draudžiamųjų įvykių registracija ir duomenų apdorojimas – vienas pačių aktualiausių ir atsakingiausių draudimo kompanijos ir jos naudojamos sistemos uždavinių ir tikslų. Reikalavimai gali būti fiksuojami ir valdomi naudojant Use Case ir jį palaikančius programinius sprendimus (pvz., IBM Rational Requirements Composer) arba BPM metodologija grįstus įrankius (pvz., IBM Websphere Business Modeler).

4.2. Reikalavimų apibrėžimo su Use Case praktikų taikymas

Draudžiamąjį įvykį registruojant galima taikyti skirtingus būdus, logiką priklausomai nuo kompanijos pasirinkimo. Draudžiamųjų įvykių veiklos komponento reikalavimų labai daug. Pateikiamas pavyzdys aprašo patį pradinį etapą – įvykio registraciją sistemoje.

Norėdamas užregistruoti draudžiamąjį įvykį, sistemoje vadinamą Claim, naudotojas tai gali atlikti arba iš išplėstinės paieškos arba jau būdamas pasirinkęs konkretų draudimo polisą. Išplėstinės paieškos naudojimas patogus greitam susijusio poliso suradimui, taip pat dublikatų išvengimui. Tuo atveju, jeigu kompanijos darbuotojas jau dirba su kliento informacija, galimas atvejis, kad poliso informacija jau rodoma. Tokiu atveju Claim gali būti pradedamas kurti tiesiai iš esamo lango. Naudotojui registruojant įvykį sistema įsitikina, ar galima registruoti (pagal susitarimą sistema palaiko Claim tik tiems polisams, kurie integruoti su kita sistema - System2), ar polisas apskritai egzistuoja, ar įvykio laiku jis buvo tikrai galiojantis, ar neatšauktas. Visi atliekami veiksmai registruojami duomenų bazėje. Naudojimo atvejai pateikiami Use Case šablonuose 4.1, 4.2 pav.

Use Case ID:	***Claim001***		
Use Case pavadinimas:	***Claim inicijavimas po paieškos***		
Sukūrė:	Julija Skatulaite	**Paskutinį taisymą atliko:**	Julija Skatulaite
Sukūrimo data:	9/16/2008	**Paskutinio taisymo data:**	9/16/2008

Aktoriai	Naudotojas (pvz., draudžiamųjų įvykių specialistas)	
Sistemos	Aplikacija	
Aprašymas	Naudotojas prieš registruodamas draudžiamąjį įvykį (vad. Claim) įsitikina, kad nekuria dublikato.	
Pradinės sąlygos	Naudotojas autentifikuotas ir turi reikiamas privilegijas, reikalingas ieškoti, matyti ir inicijuoti Claim. Naudotojas sistemoje dirba išplėstinės paieškos lange ir yra pasirinkęs įvykio registracijos veiksmą.	
Rezultatas	Pasirinktas aplikacijoje esantis polisas, kuriam bus kuriamas Claim. Veiksmų įrašai užregistruoti į įvykių duomenų bazę (Ivykiu DB).	
Normalioji eiga	Naudotojas	1. Į išplėstinės paieškos langą įveda ir polisų paieškos kriterijus.
	Aplikacija	2. Registruoja veiksmo įrašą.
	Aplikacija	3. Atlieka paiešką, išveda rezultatus. Žr. alternatyvų atvejį ID=Polisas nerastas, jeigu sistema nerado poliso įrašų, atitinkančių įvestus kriterijus.
	Naudotojas	4. Iš rastų polisų sąrašo pasirenka reikiamą polisą. Žr. alternatyvų atvejį ID=Kurti, jeigu pasirenkama kurti Claim nežiūrėjus poliso detalių.
	Aplikacija	5. Rodo pasirinkto poliso informaciją naudotojui (su galimybe kurti Claim).
	Aplikacija	6. Registruoja veiksmo įrašą.
Alternatyvūs atvejai	ID=Polisas nerastas	
	Aplikacija	1. Rodo pranešimą, kad draudimo polisas, atitinkantis paieškos kriterijus, nerastas.
	Naudotojas	2. Gali pradėti procesą iš naujo. Žr. Išplėstinį UC ID=Claim005, įvykio registracijos atvejams, kai susijęs polisas nerastas, o reikalinga registracija.
	ID=Kurti	
	Aplikacija	1. Tikrina, ar poliso tipui galima registruoti Claim. Žr. alternatyvų atvejį ID=Netinkamas poliso tipas, jeigu sistema nerado poliso įrašų, atitinkančių įvestus kriterijus.
	Aplikacija	2. Atidaro Claim registravimo langą.
	Aplikacija	3. Registruoja veiksmo įrašą.
		4. Žr. Claim003 detalėms, kaip kurti Claim.
	ID=Netinkamas poliso tipas	
	Aplikacija	1. Rodo pranešimą, kad Claim sukurti neįmanoma, nes netinkamas poliso tipas.

Išimtys	Klaidos atveju parodomas pranešimas, užregistruojamas įrašas (angl. *log error*).
Apimtis	
Naudojimo dažnis	Kompanija užregistruoja apie 1000 pranešimų per mėnesį (pranešimai gali būti susiję su keliais žalos objektais, asmenimis. Dalis pranešimų būna atšaukta).
Prioritetas	
Svarba	
Verslo taisyklės	
Ypatingi reikalavimai	
Prielaidos	Claim galima inicijuoti naudojant išplėstinę paiešką arba tiesiogiai turint pasirinktą polisą.
Pastabos ir ginčai	

4.1 pav. Paieškos UC, naudojamas inicijuojant Claim dublikatams išvengti

Use Case ID:	***Claim003***		
Use Case pavadinimas:	***Claim kūrimas***		
Sukūrė:	Julija Skatulaite	**Paskutinį taisymą atliko:**	Julija Skatulaite
Sukūrimo data:	9/16/2008	**Paskutinio taisymo data:**	9/16/2008

Aktoriai	Naudotojas (pvz. draudžiamųjų įvykių specialistas)	
Sistemos	Aplikacija	
Aprašymas	Pasirinktam polisui registruojamas įvykis (vad. Claim).	
Pradinės sąlygos	Naudotojas autentifikuotas ir turi reikiamas privilegijas, reikalingas sukurti Claim. Naudotojas yra pasirinkęs draudimo polisą ir žiūri jo informaciją.	
Rezultatas	Claim užregistruotas aplikacijoje, užklausa išsiųsta į System2. Veiksmų įrašai užregistruoti į įvykių duomenų bazę (Ivykiu DB).	
	Naudotojas	1. Poliso informacijos lange atidaro Claim skirtą kortelę.
	Aplikacija	2. Tikrina, ar poliso tipui galima registruoti Claim. Žr. alternatyvų atvejį ID=Netinkamas poliso tipas, jeigu sistema nerado poliso įrašų, atitinkančių įvestus kriterijus.
	Aplikacija	3. Atidaro Claim registravimo langą.
	Aplikacija	4. Registruoja veiksmo įrašą.
	Naudotojas	5. Įveda datą.
	Aplikacija	6. Tikrina, ar polisas egzistavo registruojamo įvykio metu. Žr. alternatyvų atvejį ID=Poliso nera, jeigu įvykio data nepatenka į poliso galiojimo periodą.
	Aplikacija	7. Tikrina, ar polisas nebuvo atšauktas registruojamo įvykio metu. Žr. alternatyvų atvejį ID=Polisas atsauktas, jeigu įvykio metu polisas buvo atšauktas.
	Naudotojas	8. Įveda su žala susijusius duomenis.

	Aplikacija	9. Įsitikina, kad nėra įvesties klaidų. Žr. alternatyvų atvejį ID=Ivesties klaida, jeigu įvesti duomenys neatitinka įvesties formato.
	Aplikacija	10. Išsiunčia užklausą sąveikaujančiai System2. Žr. alternatyvų atvejį ID=Neissiusta, jeigu užklausa neišsiųsta.
	Aplikacija	11. Užregistruoja darbo su Claim trukmę.
	Aplikacija	12. Rodo Claim informacijos langą naudotojui.
Alternatyvūs atvejai	ID=Netinkamas poliso tipas	
	Aplikacija	2. Rodo pranešimą, kad Claim sukurti neįmanoma, nes netinkamas poliso tipas.
	ID=Poliso nera	
	Aplikacija	1. Rodo pranešimą, kad įvykio data nepriklauso draudimo poliso gyvavimo laiko intervalui. (Žr. Išplėstinį UC ID=Claim005, įvykio registracijos atvejams, kai polisas nerastas)
	ID=Polisas atsauktas	
	Aplikacija	1. Rodo pranešimą, kad polisas įvykio metu buvo atšauktas.
	ID=Ivesties klaida	
	Aplikacija	1. Rodo pranešimą, kad duomenys įvesti netinkamo formato.
	ID=Ivesties klaida	
	Aplikacija	1. Rodo pranešimą, kad duomenys neišsiųsti.
Išimtys	Klaidos atveju parodomas pranešimas, užregistruojamas įrašas (angl. *log error*).	
Apimtis		
Naudojimo dažnis	Kompanija užregistruoja apie 1000 pranešimų per mėnesį (pranešimai gali būti susiję su keliais žalos objektais, asmenimis. Dalis pranešimų būna atšaukta).	
Prioritetas		
Svarba		
Verslo taisyklės		
Ypatingi reikalavimai		
Prielaidos	Claim kuriami tik su System2 suderinamiems polisų tipams.	
Pastabos ir ginčai	Aplikacijoje matomas tik claim inicijavimas, verslo logika įgyvendinama System2.	

4.2 pav. Claim (galiojančiam polisui) registravimo UC

Duotu atveju į reikalavimą žiūrima veiksmų suvokimo aspektu: **šablonuose** ypatingas dėmesys skiriamas eigai, atliekamiems veiksmams ir jų sekai. Reikalavimą įkeliant į projektą, svarbi informacija reikalinga planavimui – prioritetas (angl. *Priority*), svarba (angl. *Severity*) ir pan. Reikiamu momentu ši informacija lengvai papildoma.

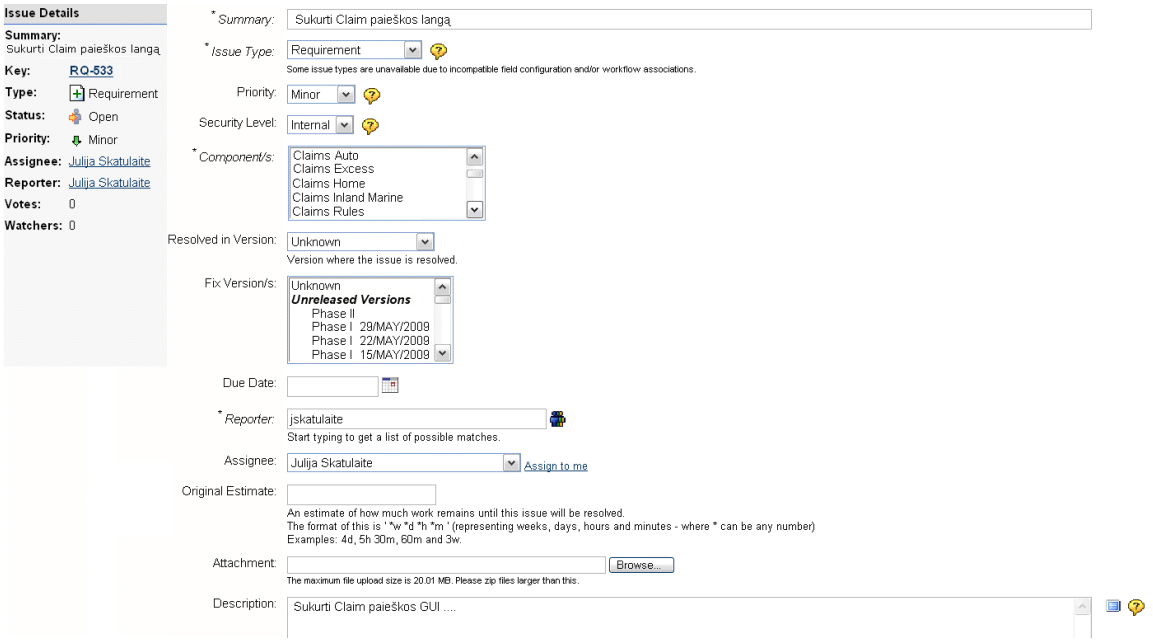
Visiems susijusiems su įvykių registracija reikalavimams apibrėžti kuriamas dokumentų rinkinys – susitartas žodynas, use case rinkinys, reikiami modeliai reikalavimams patikslinti (pvz.,

būsenos), naudotojo aplinkos GUI aprašai ir pavyzdiniai paveikslai (angl. *GUI mock-ups*). Kadangi reikalavimai susiję su sistemų integracija ir sąveika, techninė specifikacija reikalinga siunčiamos užklausos formavimui, duomenų susiejimui (angl. *Mapping*).

Sukurtus **dokumentus** reikia saugoti bendrai pasiekiamoje bibliotekoje, priskiriant **versiją**, **statusą** (pvz., patvirtintas, juodraštinis variantas ar pan). Versijoms naudojama vieninga informatyvi sistema. Paties dokumento apimtis (skyriai) gali būti organizuojami įvairiai, taikomi įvairūs šablonai, priklausomai nuo poreikio. Dažniausiai dokumente pateikiamas įvadas, atspindintis dokumento tikslus, terminologijos paaiškinimus, susitarimus, ryšį su kita dokumentacija. Toliau dokumente yra specifinė informacija konkrečiai sistemai apibūdinti - Use Case susistemintas sąrašas ir atskiri Use Case'ai arba taisyklių sąrašas ir detalės ir pan.

Turint aprašytą funkcionalumą pateiktu aukščiau būdu, uždaviniai, susiję su reikalavimais ir projektų valdymu - reikalavimų visumos dalinimas į atskiras užduotis, skyrimas komandos nariams realizuoti, ataskaitų generavimas ir padėties stebėjimas - gali būti patogiai sprendžiami projektų valdymui ir defekto kontrolei skirtu įrankiu – **Jira**.

Jira – kuriamas sprendimas, kuriam galima modifikuoti pradinį (bazinį) sprendimą ir kurti patiems procesus bei Jiros funkcijas, savybes. Reikalavimams saugoti atributus galima pridėti. 4.3 paveiksle pavaizduotas reikalavimas su jų apibūdinančiais atributais (pavadinimas, tipas, komponentas), planavimui naudotinas laikui vertinti skirtas atributas, leidimo numeris, užduotys skiriamos žmonėms, reikalavimai surišami su užregistruotomis sistemos klaidomis, susijusiomis užduotimis, matomas statusas (4.3 pav.). Pasirinkus šį metodą analogiškai koordinuojami ir nefunkciniai reikalavimai (skiriasi nuoroda į kitokio tipo dokumentus, užduoties atlikimas).

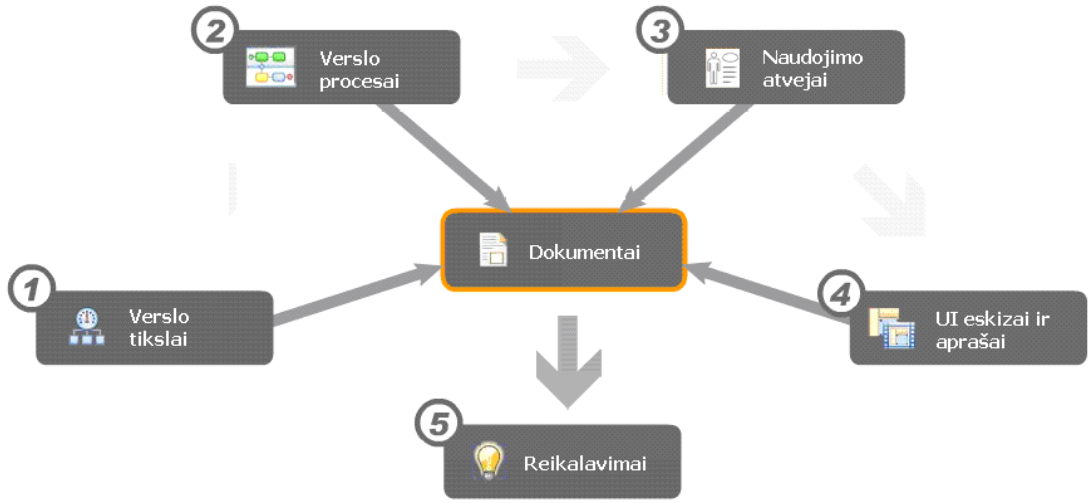


4.3 pav. Jira kortelės (angl. Jira *Ticket*) pavyzdys

Verslo analitiko darbui palengvinti savo sprendimą siūlo IBM Rational. IBM Rational programinės įrangos inžinerijoje įgyvendinamos praktikos, įgalinančios ir papildančios viena kitą, yra (Kroll 2003):

- iteracinis kūrimas (angl. *Develop Iteratively*);
- reikalavimų valdymas (angl. *Manage Requirements*);
- komponentų architektūra (angl. *Use Component Architectures*);
- modeliavimas (angl. *Model Visually (UML)*);
- kokybės tęstinumas (angl. *Continuously Verify Quality*);
- pokyčių valdymas (angl. *Manage Change*).

Projekto procesams, palaikantiems Rational geriausias praktikas, taikomas **IBM Rational Method Composer**, įrankis procesams kurti, konfigūruoti, žiūrėti ir publikuoti. Verslo (veiklos) analitikui pritaikytas **IBM Rational Requirements Composer**, įrankis reikalavimams aprašyti ir prototipams kurti. Grafiniai editoriai palengvina reikalavimų sisteminimą ir analizę, eskizų kūrimą. Dirbant su IBM Rational Requirements Composer sukuriamas projektas, pridedami Use Case, reikalavimai, dokumentai, GUI eskizai. Suderinamumas su IBM Rational RequisitePro palaiko pilną nuo pradžios iki pabaigos (angl. *End-to-end*) veiklos sprendimą. Vyksta suderintas kūrimo procesas, kur reikalavimai atskleidžiami palaipsniui ir pateikiami vykdymui: iškeliami verslo tikslai (angl. *Business Objectives*), konkretizuojami verslo procesai (angl. B*usiness Processes*), kuriami use case, pridedami UI eskizai ir aprašai (angl. *UI sketches and Storyboards*) ir viskas sudedama į dokumentus (angl. *Documents*) bei sukuriami reikalavimai (4.4 pav.).



4.4 pav. Reikalavimų procesas, palaikomas Rational Requirements Composer

Kita galimybė - naudoti **Telelogic DOORS.** Tai Rational komplekto (angl. *Portfolio*) plėtinys nuo 2008 m. balandžio mėn., skirtas reikalavimų fiksavimui (angl. *Capture*), susiejimui (angl. *Link*), sekimui (angl. *Trace*), analizei (angl. *Analyse*) ir valdymui (angl. *Manage*) (IBM 2009). Telelogic Doors pateikia centralizuotą sistemą, reikalavimai saugomi su reikalingais apibūdinančiais atributais (pvz., statusas, klientas), galimas skirtingas atvaizdavimas, filtrų taikymas, greitas ataskaitų generavimas, galimybė stebėti, vizualizuoti ir planuoti produkto leidimus (angl. *Releases*). Įrankio panaudojimas ir privalumai panašūs į auksčiau pavaizduotos Jira sistemos.

4.3. Reikalavimų valdymas naudojant BPEL

BPM metodikai naudoti pasirinktas IBM Websphere Business Suite komplektas, palaikantis kūrimo ciklą nuo pradžios iki pabaigos. Reikalavimų modeliavimui, veiklos vertinimui skirtas Business Modeler, veikiantis skirtingais režimais (angl. *Mode*) priklausomai nuo smulkumo poreikio. Kuriant veiklos modelį, reikia jį suvokti t.y. suprasti veiksmus. Kuriant modelį atkreipiamas dėmesys į:

- užduočių variacijas, kada įvyksta (angl. *Task variations and when those variations occur*);
- įvestį ir išvestį kiekvienai veiklai (angl. *Activity inputs and outputs*);
- alternatyvų elgesį (angl. *Alternative tasks*);
- pilną užduoties aprašymą (angl. *Complete task descriptions*);
- reikalingusvaidmenis kiekvienai užduočiai (angl. *Associated roles*).

Taigi kuriant modelį išskiriami **veiksmai, jų seka**. Modeliu atvaizduota chronologinė įvykių proceso eiga (angl. *Process flow*) Use Case šablone pateikiama normaliosios eigos ir alternatyvių atvejų skyriuose.

Įvykio registracijos pavyzdyje galima išskirti veiksmus (turinčius trukmę, priklausomus nuo įvesties duodančius rezultatą (išvestį)): duomenų įvedimas (paieškos kriterijų, įvykio registracija); įrašų radimas; rastų ar pasirinktų įrašų rodymas naudotojui; veiksmų registravimas duomenų bazėje; duomenų siuntimas susijusiai sistemai; pranešimų išvedimas naudotojui.

Proceso metu priimami sprendimai (tikrinama) ar naudotojas nori peržiūrėti polisą, ar polisas egzistuoja, ar tinkamas (terminas, tipas), ar suvesti duomenys teisingi, ar neįvyko klaida.

Būtent veiksmų ir sprendimų aprašo **visuma** atvaizduoja veiklą, funkciją.

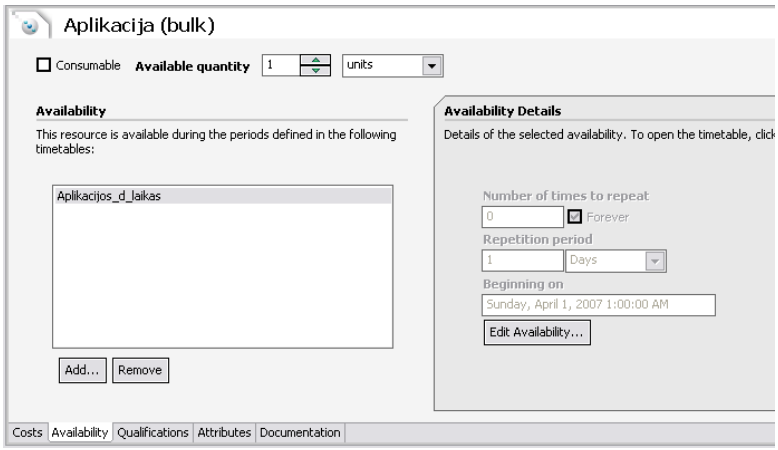
Kiekvienam veiksmui atlikti reikalingi **ištekliai**. Galimi ištekliai:

– žmogiškieji ištekliai arba žmonės (angl. *Individual Resources*), jie dar papildomai skirstomi pagal priskirtą rolę (angl. *Role*);
– įranga, medžiaga, mašina (angl. *Bulk Resources*) – kiti įrankiai, reikalingi veiksmui atlikti.

Užduotims atlikti ir ištekliams apibūdinti svarbūs **atributai**:

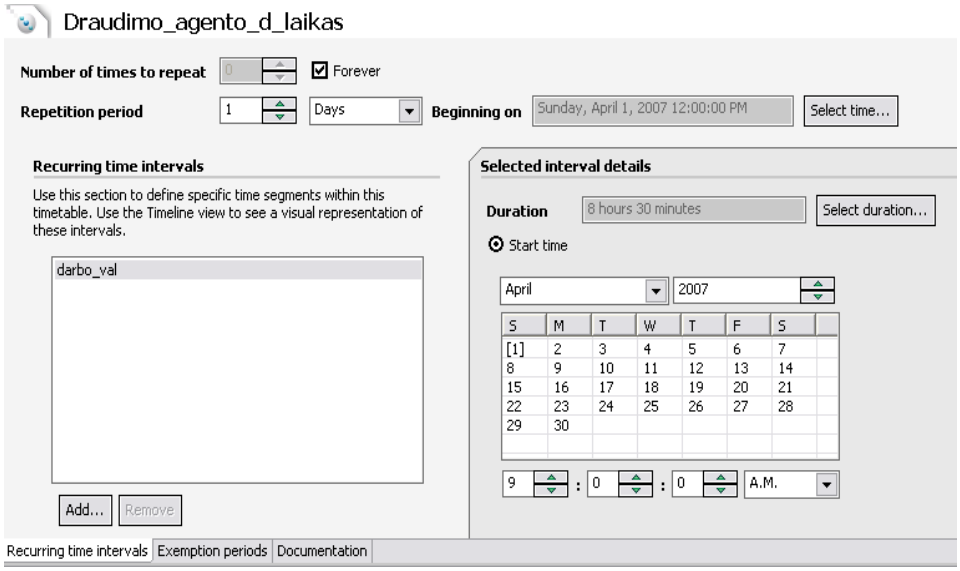
– vaidmenys (angl. *Role*) – žmonių skirstymas pagal atsakomybes, funkcijas ir pan.;
– tvarkaraštis (angl. *Timetable*) – laikas/valandos, kai išteklius yra prieinamas, naudojamas tiek žmogiškiesiems ištekliams, tiek įrangai.

Registruojant įvykį dalyvauja aplikacijos naudotojas, tiesiogiai dirbantis su sistema (apibrėžiamas *Individual Resources* tipu), aplikacija ir veiksmus registruojanti bazė (ištekliai, apibrėžiami *Individual Resources* tipu). Ištekliams priskiriamas darbo laikas (Business Modeler sukuriami *Timetable*), kada jis veikia ir yra prieinamas, naudotojui priskiriamas vaidmuo (kuriamame projekte sukuriama *Role*). Kiekvienam ištekliui apibrėžiamos išlaidos, jų tipas: vienkartinės (angl. *One-time cost*) ar laiko vienetui (angl. *Cost per time unit*) (4.5, 4. 6 pav.).



4.5 pav. Išteklio kūrimas IBM Business Modeleriu

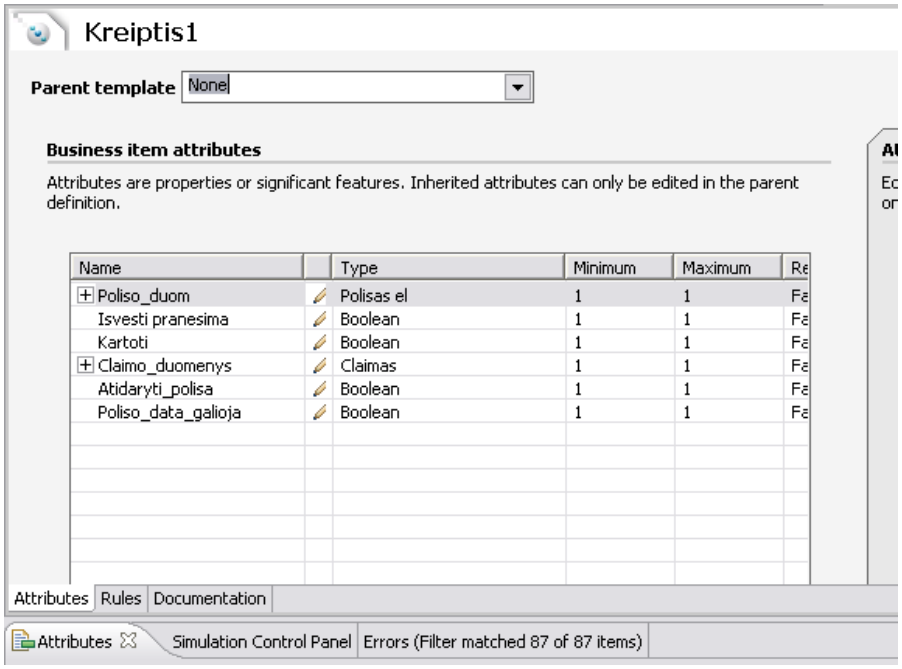
Kuriant tvarkaraštį nurodomos darbo dienos, valandos. Kortelėje *Exemption periods* nurodoma jau į kitą sukurtą tvarkaraštį, kuriame aprašyta, kuriuo laiku išteklius nedirba/neprieinamas (išeiginės, šventinės dienos, atostogos). Aplikacijai, duomenų bazei nurodomas darbo laikas 24val/parą. Žmogiškojo darbo išteklio darbo tvarkaraštis yra darbo valandos darbo dienomis.



4.6 pav. Tvarkaraščio kūrimas IBM Business Modeleriu

Procese dalyvauja ir modeliuojant kuriami veiklos objektai (angl. *Business Items*), t.y. dalyvaujantys procese elementai: darbo dokumentai, darbo produktai, kiti reikmenys, transformuojami į veiklos operacijas. Tai gali būti bet kas sukurta, surinkta, patikrinta, ištestuota, pakeista. Šie objektai yra perduodami iš vieno proceso etapo kitam, t.y. kas reikalinga gauti, kad etapas įvyktų; kas grąžinama, jam įvykus.

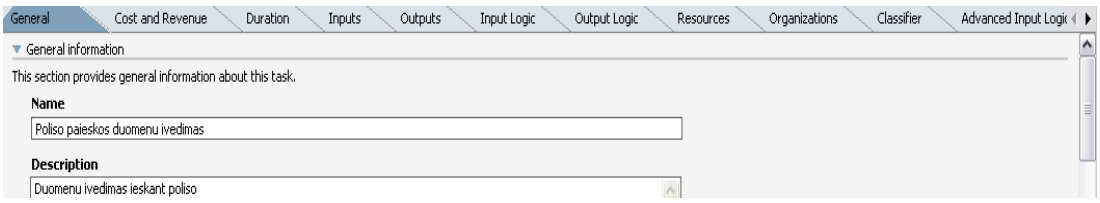
Pavyzdiniame draudimo įvykio registravimo modelyje proceso metu perduodami duomenys aprašyti Business Modeler sukurtu veiklos objektu Kreiptis1. Pagrindinis veiklos objektas Kreiptis1 sudėtinis. Be paprasto tipo kintamųjų (pvz., *Boolean*), naudojamas išvestinis tipas iš kitų objektų (tipai: Polisas el, Claimas), turinčių savo atributus. Įvedus neteisingus duomenis, procesas užbaigiamas, išvedamas į ekraną klaidos pranešimas naudotojui. Tai atvaizduota modelyje veiklos objektu Pranesimas (4.7 pav.).



4.7 pav. Veiklos objekto Kreiptis1 atributai

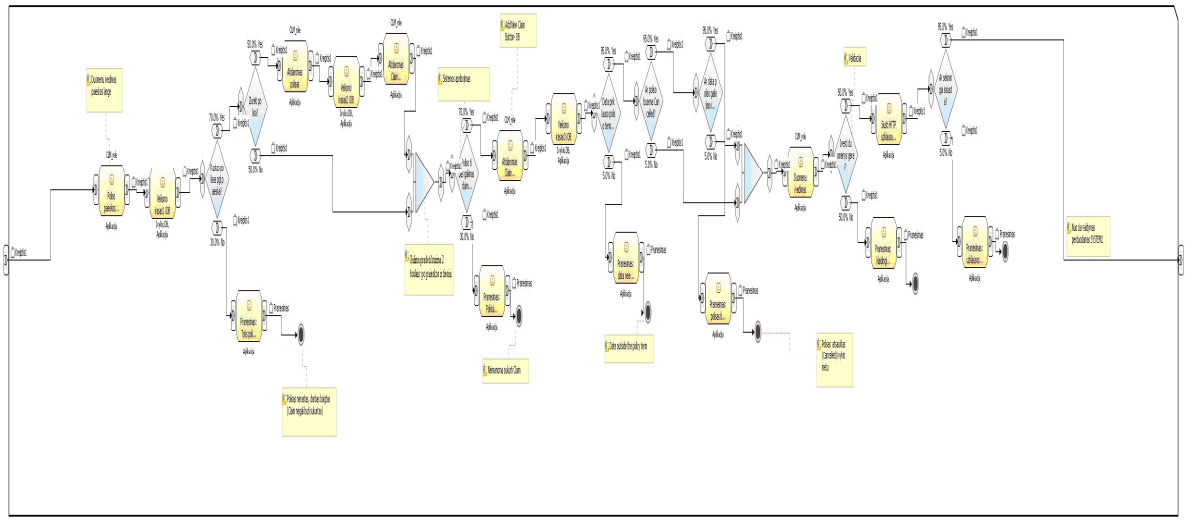
Taigi išskyrus veiklos veiksmus, juos sudėjus nuoseklia eiga, priskyrus išteklius, darbo tvarkaraščius, apibrėžus dalyvaujančius verslo objektus tampa aiškiai matomas ir tinkamas analizei bei vertinimui veikimo, **funkcionalumo principas**, **užduotys kurias reikia atlikti**.

Kiti svarbūs aspektai, kuriant sprendimus, yra **finansinis ir laiko** (trukmės) vertinimas. Business Modeler modelyje kiekvienam veiksmui norodoma trukmė (angl. *Duration*) – kiek užtrunka jį atlikti. Kiekvienas veiksmas vertinamas finansiniu aspektu ir yra susietas su atitinkamomis išlaidomis ir įplaukomis (angl. *Cost and Revenue*). Kiekvienas veiksmas susiejamas su reikalingais (jau sukurtais) ištekliais (*Bulk/Individial Resources*) su jų laiku (*Timetable*), organizacija (angl. *Organization*). Proceso kiekvieno veiksmo atributai nustatomi atitinkamose veiksmą aprašančiose kortelėse (4.8 pav.).

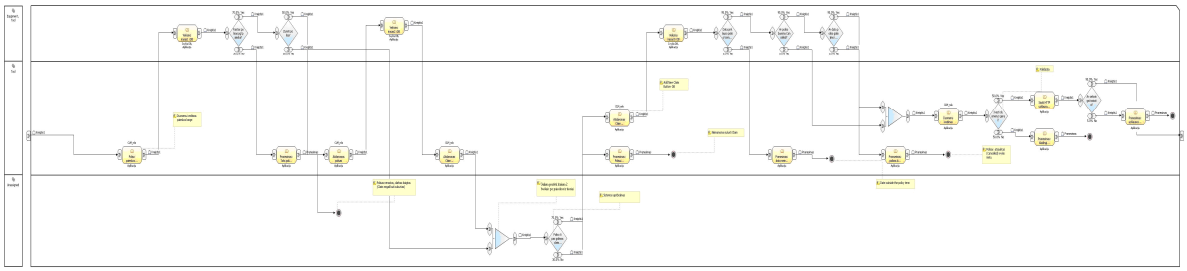


4.8 pav. Veiksmo atributų nustatymas Business Modeleryje

Sukūrus veiklos procese dalyvaujančius objektus, jiems priskyrus būdingus duomenis, atributus, gaunamas dar tikslesnis veiklos modelis (4.9, 4.10 pav.), atvaizduojantis veiklos procesą ir su juo susijusią informaciją.



4.9 pav. Draudžiamojo įvykio registracijos modelis



4.10 pav. Draudžiamojo įvykio registracijos modelis pagal pasirinktą kriterijų

Taigi iš chronologinės veiksmų sekos sukurtas modelis palengvina suvokimą, vienareikšmį procesų **supratimą**. Modelis lengvai ir greitai **modifikuojamas.** *IBM Business Modeler* leidžia peržiūrėti modelį keliais režimais (angl. *Switch Swimlane*), tuo pabrėžiant požiūrį **rūpimu aspektu**.

Priskyrus veiksmams susijusią informaciją (trukmę, išteklius, finansinius rodiklius ir pan.) galima **vertinti** situaciją (angl. *As -Is*), tobulinimo galimybes (angl. *What-If* situacijas). Vertinti padeda šios savybės:

- Modelio veiksmų **simuliacijos** (angl. *Simulation*) galimybė, t.y. dinaminės analizės (angl. *Dynamic Analysis*) pagal pasirinktą kriterijų galimybė, kurios metu sukuriamas momentinė modelio kopija iš esamų procesų, priskiriamas laikas, pradžia (angl. *Simulation Snapshot).*

– Veiklos apskaičiuoto vertinimo galimybė, pasinaudojant IBM Business Modelerio savybe „*Business Measure*“. Ja galima sukurti veiklos matavimo modelį (angl. *Business Measure Model*), kuris apibrėžia veiklos įvertinimą, kaip leidžia matuoti veiklos procesus, jų vykdymą remiantis **metrikomis** (angl. *Metrics*), vykdymo indikatoriumi KPI.

Informacija modelyje gali būti atvaizduota įvairiu pavidalu ir naudinga dokumentacijai ruošti, pagreitinti ruošimą (tiek aprašomojo pobūdžio, tiek testavimo atvejams). Duomenų **statinė analizė** (angl. *Static Analysis*) galima įvairiais būdais: išteklių analizė (angl. *Resources Analysis*), organizacijų analizė (angl. *Organization Analysis*), bendroji analizė (angl. *General Analysis*), pagal pasirinkimą, galima ir matricinė analizė. Reikiamai atvaizduoti duomenys susistemina turimą informaciją, gali būti panaudojami dokumentuojant. Pvz., panaudojus matricinę analizę galima sukurti žmonių ir jų rolių sąryšio matricą. Ji reikalinga tiek analitikams, norint suvokti procesus, tiek realizuojančiai sistemą komandai, atsakingai už saugumą ir naudotojų galimų sistemoje veiksmų konfigūraciją (4.11 pav.).

	CLM_admin	CLM_kurimas	CLM_role
Draudimo_Agentas	X	X	X
Brokeris		X	X

4.11 pav. Žmonių ir priskirtų vaidmenų ryšių atvaizdavimas naudojant statinę analizę

Kitas būdas greitai dokumentacijai ir situacijos matymui yra **ataskaitų** (angl. *Reports*) generavimo galimybė. Siūloma skirtingų ataskaitų, pvz., procesų santrauka (angl. *Process Summary*), procesų aprašymas (angl. *Process Annotation*), procesų apibūdinimas (angl. *Process Details*), procesų procedūros (*Process Procedure*), procesų specifikacija (*Process Specification*), procesų statistika (*Process Statistics*) (4.12 pav.).

Connection Steps For Process

Procedure	Step	Role	Type	Next	Description
1	Poliso paieskos duomenu ivedimas	CLM_role	Task (t)	2	Duomenu ivedimas ieskant poliso
2	Veiksmo irasas1 i DB		Task (t)	3	
3	Rastas polisas pgl paieska?		Decision (e)	4,25	
4	Ziureti polisa?		Decision (e)	5,8	
5	Atidaromas polisas	CLM_role	Task (t)	6	
6	Veiksmo irasas2 i DB		Task (t)	7	

4.12 pav. Ataskaitos „Process Procedure“ fragmentas parodo veiksmų seką

Modelį turint BPEL kalboje, galima gauti **UML** diagramas. BMP akcentuojamas tęstinumas dar padidina modelio vertę. Priėmus sprendimą naudoti visą Business Suite komplektą, ne tik Business Modeler modeliavimui, sukurtame modelyje jau pateikiamas realizacijos įdirbis. Naudojant toliau analitiko paruoštą modelį, mažiau galimybių lieka savotiškai interpretacijai, realizuojančiam personalui galima mažiau gilintis į jau paruoštą modelį ir jo logiką ir susikaupti tik ties savo užduotį atitinkančiom realizacijos detalėm.

Nepriklausomai nuo to, koks darbo organizavimas yra pasirenkamas ir kurie etapai yra išskiriami, teisingų funkcinių, nefunkcinių, kliento ir kitokių tipų valdant reikalavimus susiduriama su daug iššūkių: bendravimo (skirtingi požiūriai, bendravimo stoka, skirtingas supratimas), teisingo informacijos išrinkimo (teisingas poreikių nustatymas, detalių nepraleidimas), valdymo (statuso, pokyčių sekimas), valdymo projekto aspektu (laiku pateikimas komandai, balansavimas tarp vizijų ir galimybių), kokybės (teisingi, pilni, nuoseklūs, vienareikšmiai, suprantami, suskirstyti pagal svarbumą ir stabilumą, vertinami, lengvai keičiami, sekami). Todėl egzistuoja skirtingi sprendimai ir įrankiai.

4.4. Išvados

Aplikacijų reikalavimų valdymas – sudėtinis procesas, kurio užduotys glaudžiai susiję su visa dalyvaujančia komada ir visu aplikacijų kūrimo ciklu. Tvarkingas reikalavimų valdymas, tinkamų metodų, įrankių pasirinkimas, optimalus procesų organizavimas svarbus ir pačiam **analitikui**.

Nors verslo analitiko tikslai ir uždaviniai nesikeičia, reikalavimų valdymas gali būti organizuojamas skirtingais būdais. Funkciniams reikalavimams aplikacijoms aprašyti taikomi Use Case ir UML. BPM metodologijoje reikalavimų analizė sukoncentruota į „vykdomųjų“ modelių kūrimą. Techninėms detalėms konkretizuoti, pridedama techninė dokumentacija, atskirai kuriami ir susiejami su veiksmais GUI eskizai ir aprašai. Su BPEL modeliais, kurie tiesiogiai dalyvauja kūrime, dalis dokumentacijos pakeičiama tiesioginiu modelio redagavimu. Analitikas gali naudoti modelį situacijos analizei, simuliuoti procesus, greitai matyti metrikas, generuoti reikiamas ataskaitas. Kaupiant reikalavimus su Use Case svarbus ryšių matymas, tinkamas informacijos skaidymas ir jos nedubliavimas. Kiekvienu atveju analitikas turi sugebėti suvokti esamą situaciją – turimą funkcionalumą ar savybes, statusą, kas yra daroma ir kas gali būti padaryta bei kiek tai aktualu. Reikalavimų ciklas turi turėti aiškią procesų seką, sistemingą versijavimą. Abi metodikos turi savos specifikos elementų, tačiau abiem atvejais verslo analitikas gali rasti būdą efektyviai pateikti ir valdyti reikalavimus.

Išvados

Atlikus programinės įrangos kūrimo analizę nustatyta, kad sėkmingam produktų kūrimui tapo neišvengiama būtinybe ne tik pažangių technologijų kokybiškas taikymas, bet ir etapų bei procesų efektyvus valdymas, priežiūra, vertinimas.

Išanalizavus projektų valdymo metodus pastebėta, kad norint sėkmingai valdyti projektus svarbu taikyti geras praktikas. Geriausiai pritaikomi korporaciniai projektų valdymo modeliai, tačiau kiekvienai situacijai metodus reikia taikyti individualiai.

Projekto sėkmei būtinas tinkamų reikalavimų atskleidimas, suvokimas, pateikimas komandai. Norint susitvarkyti su vis didėjančia apimtimi, sudėtingumu, gebėti reaguoti į pokyčius reikia suvokti reikalavimų procesą, tobulinti reikalavimų valdymo metodus.

Reikalavimų aspektu, priimant sprendimus svarbu suvokti analizuojamą sritį, kiekvienu momentu žinoti situaciją (turimą produktą, statusą), gebėti vertinti ir racionaliai derinti poreikį ir galimybes. Darbe draudimo srities reikalavimo pavyzdys buvo apibrėžtas Use Case šablonu ir naudojamas kartu su Jira arba IBM Rational produktais. BPM tyrimui buvo pasirinktas IBM Business Modeler. Reikalavimų apibrėžimas grįstas Use Case išskiria reikalavimus, dokumentų ruošimą kaip atskirą modulį. BPM modelis gali būti naudojamas veiklos scenarijams vertinti, modelis tiesiogiai naudojamas realizacijai. Turint BPEL modelį iš jo galima atskleisti informaciją norimu aspektu, galima generacija į UML, rodiklių apskaičiavimas, pagreitėja koregavimo procesas, išvengiama informacijos dubliavimo. Galima greitai paruošti testavimo atvejus, pvz., Smoke ir End To End testus, kurie reikalingi verslo analitikui. Visgi funkcinių reikalavimų rinkimas su Use Case ir procesų valdymo įrankiu gali būti sėkmingas sprendimas dirbant su konkrečiu stambiu klientu (kuriam dažnai reikia išsamesnės dokumentacijos, prisiderinimo prie jo laiko, t.y. atsiliepimai ne visada gaunami laiku ir t.t.).

Literatūros sąrašas

1. Ambler S. [interaktyvus]. Agile Modelling. [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://www.agilemodeling.com/essays/>
2. ESI International. Eight Things Your Business Analyst Need to Know. A Practical Approach to Recognizing and Improving Competencies. White Paper. JAV, 2006. 20 p.
3. Futrell R., Shafer D., Safer L. Quality Software Project Management. JAV, 2002, 1680 p.
4. Havey M. Essential Business Process Modeling. JAV, 2005. 350 p.
5. IEEE Computer Society [interaktyvus]. History [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: < http://www2.computer.org/portal/web/about/history>
6. IBM Software Group. REQ480Mastering Requirements Management with Use Cases, Instructor Manual. Slides. JAV, 2003. 374 p.
7. IBM. [interaktyvus]. Solutions for Business Process Management, Product Management, Software Development and Systems Engineering [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: < http://www.telelogic.com/Products/>
8. IIBA. A Guide to Business Analysis Body of Knowledge v.1.6. JAV, 2006. 329 p.
9. Infravision. [interaktyvus]. Products > APM [žiūrėta 2009 gegužės 9 d.]. Prieiga per internetą: <http://www.infravision.nl/NL/product_SAAS_APM.htm&lang=1?PHPSESSID=691f8369aed0092f8c8c08f35e7483c2>
10. ISACA. [interaktyvus]. CobiT 4.1 is Available! [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: <http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=7981>
11. IT Governance Institute. CobIT Mapping: Overview of International IT Guidance, 2nd Edition. JAV, 2006. 76p.
12. IT Governance Institute, Office of Governance Commerce. Aligning COBIT® 4.1, ITIL® V3 and ISO/IEC 27002 for Business Benefit. A Management Briefing From ITGI and OGC. JAV, 2008. 131 p.
13. ITIL. The Official Introduction to the ITIL Service Lifecycle. UK, 2007. 252 p.
14. ITIL & ITSM World. [interaktyvus]. The ITIL Triangle. [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: < http://www.itil-itsm-world.com/itsm-kit.htm> < http://www.itil.org.uk/itsm.htm>
15. Itsm.lt. [interaktyvus]. Metodologijų palyginimas [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: <http://www.itsm.lt/index.php>

16. Keen M., Cavell J., Sarah Hill S., Chee Keong Kee, Wendy Neave, Bradley Rumph, Hoang Tran. BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating. Redbooks. JAV, 2004. 390 p.
17. Kroll P. The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. JAV, 2003. 464 p.
18. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition. JAV, 2004. 736 p.
19. Lin Chyun-Chau, Luh Ding-Bang. A vision-oriented approach for innovative product design Advanced Engineering Informatics, Volume 23, Issue 2, April 2009, Pages 191-200.
20. Lombardi. Making the Case for BPM. A Benefits Checklist. JAV, 2008. 13 p.
21. Long J.O. IBM. ITIL® v3 At a Glance. Information Quick Reference. JAV, 2008. 84 p.
22. Microsoft. [interaktyvus]. Microsoft Operations Framework 4.0. [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://technet.microsoft.com/en-us/library/cc506049.aspx >
23. OracleApps Epicenter [interaktyvus]. What is Oracle‘s Application Implementation Methogology (AIM). [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://www.oracleappshub.com/methodology/what-is-oracles-application-implementation-methodology-aim/>
24. PMI. A Guide to Project Management Body of Knowledge. Third Edition. JAV, 2004. 400 p.
25. Pultorak D., Bon J. ir kt. [interaktyvus]. Microsoft Operations Framework (MOF)– A Pocket Guide. žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://books.google.lt/books?id=_-2aMhDAKR8C&dq=mof&printsec=frontcover&source=bl&ots=SqWlnsM5i3&sig=lGlt1tYmzk1RBQTMBtftgrQxkK4&hl=lt&ei=vmAISvqRBYuS_QamkKmLBw&sa=X&oi=book_result&ct=result&resnum=8#PPT2,M1 >
26. Resource Center for Oracle E-Business Suite. [interaktyvus]. Oracle E-business Suite Knowledge Source. [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://ebusinessuite.wordpress.com/category/aim-methodology/>
27. Researchcreations [interaktyvus]. Service Management. [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://www.researchcreations.com/service.asp>
28. Robertson S., Robertson J. Mastering the Requirements Process (2nd Edition). JAV, 2006. 592 p.
29. Rosenberg D., Stephens M. Use Case Driven Object Modeling with UML Theory and Practice. JAV, 2007. 471 p.
30. Silver B. The BPMS ValueProposition. BPMInstitute.org. JAV, 2007. 17 p.
31. Sliger M, Broderick S. The Software Project Manager's Bridge to Agility. JAV, 2008. 384 p.

32. Stappers P.J., van Rijn H, Kistemaker S.C., etc. Designing for other people's strengths and motivations: Three cases using context, visions, and experiential prototypes Advanced Engineering Informatics, Volume 23, Issue 2, April 2009, Pages 174-183 [žiūrėta 2009 gegužės 20 d.]. Prieiga per internetą: < <http://www.sciencedirect.com/science?_ob=ArticleListURL&_method=list&_ArticleListID=926139693&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=8144423&md5=ddbc3510f0cd1bf4fcd79c9c2f713a0f >
33. Steeghs R. [interaktyvus]. RUP for SOA. [žiūrėta 2009 kovo 15 d.]. Prieiga per internetą: <http://ruudsteeghs.blogspot.com/2007/04/rup-for-soa.html>
34. Sureau D. [interaktyvus]. History of Programming Languages and Their Evolution [žiūrėta 2009 kovo 15 d.]. Prieiga per internetą: < http://www.scriptol.com/programming/history.php>
35. Swenson K. [interaktyvus]. BPM is not Software Engineering [žiūrėta 2009 balandžio 13 d.]. Prieiga per internetą: <http://www.bpm.com/bpm-is-not-software-engineering.html>
36. Škatulaitė J. ir kt. Veiklos procesų valdymas. Konferencijos, įvykusios 2008m. balandžio 9d. medžiaga. Vilnius, 2008, p.301-310.
37. Škatulaitė J. Aplikacijų reikalavimų valdymas. Konferencijos, įvykusios 2009m. balandžio 9d. medžiaga.
38. TechnoSolutions. [interaktyvus]. Why manage Requirements? [žiūrėta 2009 kovo 15 d.]. Prieiga per internetą: <http://www.technosolutions.com/topteam_requirements_management.html>
39. The Atlantic Systems Guild. [interaktyvus]. Requirements Resources. Volere Requirements Specification Template. [žiūrėta 2009 kovo 15 d.]. Prieiga per internetą: <http://www.volere.co.uk/template.htm#anchor736934>
40. The ISO 20000 Directory. [interaktyvus]. The ISO 20000 (BS15000 / BS 15000) ITSM Standard. [žiūrėta 2009 balandžio 7 d.]. Prieiga per internetą: < http://www.bs15000.org.uk/>
41. The Programming Languagel Weblog [interaktyvus]. History [žiūrėta 2009 balandžio 10 d.]. Prieiga per internetą: < http://lambda-the-ultimate.org/taxonomy/term/7>
42. Wahli U., et. WebSphere Process Integration V6: Business Process Management Modeling through Monitoring. Redbooks. JAV, 2006. 484 p.
43. Wiegers K.E. Software Requirements, Second Edition. JAV, 2003. 516 p.
44. Wiegers K.E. More About Software Requirements: Thorny Issues and Practical Advice. JAV, 2006. 216 p.
45. Withall S. Software Requirement Patterns. JAV, 2007. 384 p.

Priedai

VEIKLOS PROCESŲ VALDYMAS

Julija Škatulaitė, Turalas Rasulovas, Jurgis Mikučionis
Vilniaus Gedimino technikos universitetas, juskada@yahoo.com

Anotacija. Nagrinėjama veiklos procesų valdymo (angl. *Business Process Management, BPM*) prasmė, kilmė ir aktualumas mūsų laikais, procesų visumos valdymo galimybės. Aptariami veiklos procesų valdymo komplektai (angl. *Business Process Management Suite, BPMS*), architektūra, naudojami standartai. Apžvelgiami keli BPMS gamintojai, jų siūlomi programiniai sprendimai.

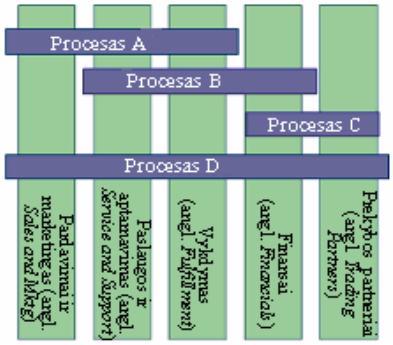
Įvadas

Efektyvus, produktyvus, inovatyvus verslo ar veiklos procesas galimas tik tinkamai naudojant informacines technologijas. XXI amžiuje, norint viską atlikti greitai ir kokybiškai, mažesnėmis sąnaudomis ir didesne apimtimi, žvilgsnis nukreiptas į: klientui aktualių (angl. *on-demand*) poreikių atitikimą; proceso visumos *(*angl. *end-to-end)* valdymą; galimybę atsižvelgti į geriausią elgseną konkrečiu atveju; inovatyvų atsaką pokyčiams; galimybę darant mažiau daryti daugiau. Šios aktualijos verčia tobulinti *BPM*, *BPMS* sprendimus. Pats *BPM*, kaip valdymo disciplina, atsirado prieš 20 metų kaip naujas požiūris į veiklą (kaip planuoti, analizuoti, vertinti ne atskiras dalis, bet procesus visame veiklos kontekste) (Silver 2007). Straipsnyje aptarsime *BPM* metodologiją ir įgyvendinimo akcentus.

Problematika

BPM – plati sąvoka, susijusi ir su veiklos modeliavimu, rodiklių vertinimu, ir su technologija, ir su verslo ir IT bendradarbiavimu, t. y. ir veiklos analizė, ir paskirstymas, ir integracija, ir *KPI* (angl. *Key Performance Indicators*), ir *Workflow* automatizavimas, ir *SOA* (angl. *Service-oriented architecture*), ir t. t.

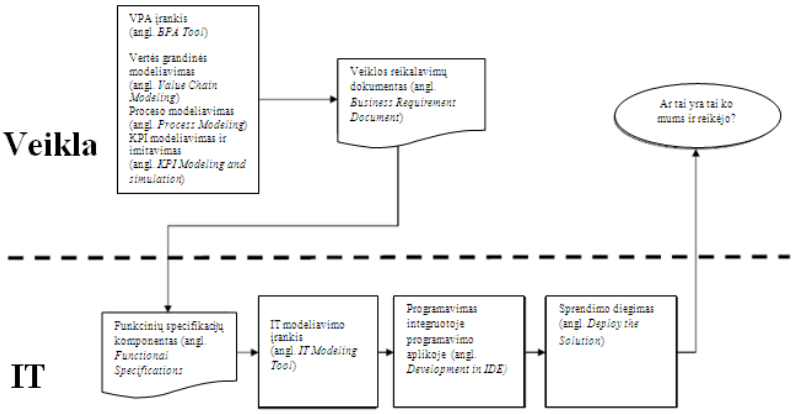
BPM esmė – naujas požiūris į veiklą, kai atskirai analizuojama ne atskiros veiklos, duodančios veiklos rezultatą, bet veikiantys keliose veiklose procesai, kurie leidžia susikoncentruoti ties procesu nuo jo pradžios iki pabaigos (angl. *end-to-end*). Daug privalumų jau žinoma naudojant tokias *enterprise* aplikacijas, kaip pvz., *ERP*, *CRM*, kurios individualių funkcijų ar sistemų lygmenyje automatizuoja veiklą, daro ją efektyvesne, suteikia galimybę duomenų integracijai ir kontrolei. Be to, tai ištakos besikuriant *end-to-end* procesų valdymui, kuriam vykstant dėmesys nukreiptas i galutinį tikslą (angl. *mission-critical*) ir procesai „laužo“ ribas tarp atskirų funkcijų (1 pav.).



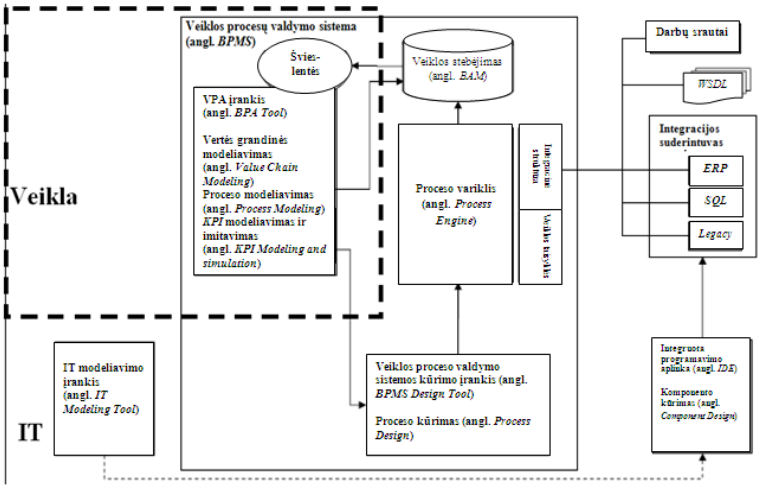
1 pav. Kompanijos ir verslo aplikacijos organizuojamos pagal atskiras veiklos funkcijas, kai dėmesys sutelktas i tikslą, procesai „laužo“ ribas (Silver 2006)

BPM disciplina reikalinga ne tik dėl kitokios veiklos analizės galimybės. Ši naujo požiūrio metodika leidžia, akcentuojant modeliavimą ir analizę, paversti modelį vykdomu, matuojamu ir galimu optimizuoti. Tam tikslui naudojamas veiklos procesų valdymo komplektas – *BPMS*, kuris „laužo“ barjerus tarp atskirų

funkcijų ir padidina efektyvumą (automatizuojant), paslankumą pokyčiams, suderinamumą (logika paremta taisyklėmis) ir aiškų proceso visumos, t. y. viso funkcijų derinio, matomumą. Taip pat *BPMS* visą veiklos procesą padaro vientisą, paremtą veiklos specialistų ir IT specialistų bendradarbiavimu (2 pav., 3 pav.).



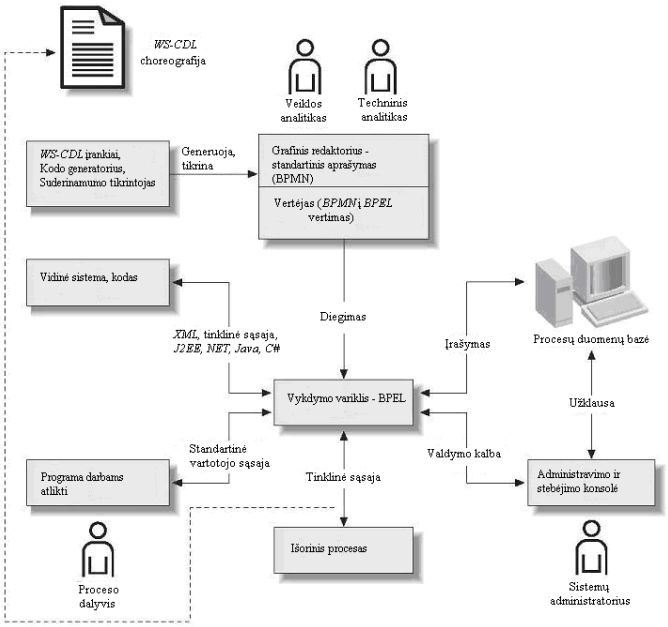
2 pav. Seno požiūrio *BPM* įgyvendinimas (Silver 2006)



3 pav. Naujo požiūrio *BPM* įgyvendinimas – *Clean Handoff* stilius (Silver 2006)

BPM standartai

BPM turi daug žinomų standartų, iš kurių renkamasi geriausios kombinacijos. Dažnai veiklos procesų modeliavimo standartų grupės analizė pradedama veiklos procesų modeliavimo notacija *BPMN* (angl. *Business Process Modeling Notation*), kurios pagrindinis siekis yra suteikti ženklų sistemą, kurią lengvai suprastų visi procese dalyvaujantys žmonės: ir verslo analitikai, kuriantys juodraštinius veiklos procesų modelius, ir techniniai darbuotojai, kurie atsakingi už veiklos procesų vykdymą, reikiamų technologijų diegimą, ir galiausiai tie atstovai, kurie valdys ir prižiūrės šiuos procesus. Veiklos procesų diagramų braižymas yra loginė veiklos procesų valdymo įgyvendinimo pradžia. *BPMN* vaizduoja nuoseklų žingsnių ir veiksmų srautą veiklos procese (4 pav.).



4 pav. *BPM* architektūra (Havey 2005)

BPMN yra susieta su veiklos procesų vykdymo kalba *BPEL4WS* arba tiesiog *BPEL* (angl. *Business Process Execution Language*) ir yra vykdoma *BPEL* variklio. *BPEL*, kaip procesų vykdymo kalba, suformuoja esminį veiklos procesų valdymo sluoksnį. Egzistuoja ir kitos panašaus pobūdžio vykdomos kalbos, pavyzdžiui, *XPDL*, *BPML* ir kt., tačiau šiuo metu labiausiai dėmesys atkreiptas į didžiųjų gamintojų naudojamą *BPEL*, kuri, prireikus, gali būti modifikuojama, priderinant prie kitų kalbų ar perimant kai kurias reikiamas jų savybes.

Tinklinių paslaugų choreografija (angl. *choreography*) paremta *W3C* (angl. *World Wide Web Consortium*) konsorciumo *WS-CDL* (angl. *Web Services Choreography Description Language*) kalba. Choreografija apibūdina procesus iš globalaus požiūrio taško. Choreografijos globalus požiūris yra kontrastas *BPEL* orkestravimui (angl. *orchestration*). *BPEL* procesas yra vieno dalyvio procesas, o choreografija yra dalyvių grupės sąveikos modelis.

WSDL (angl. *Web Services Description Language*) arba tinklinių sąsajų apibūdinimo kalbą pristatė *Microsoft* kartu su *IBM* 2000-ųjų metų spalį. Ji buvo paremta *IBM* tinklu pasiekiamų paslaugų aprašymo kalba ir *Microsoft* paprastų objektų pasiekimo protokolų (*SOAP*) sutartine kalba. *WSDL* apibūdina verslo funkcijas nuo diegimo nepriklausomu būdu, kuris leidžia kitoms taikomosioms programoms jas surasti ir panaudoti. Taigi, *WSDL* yra kalba, pagal kurią nusakomos tinklinės sąsajos ir aprašomas jų naudojimas. Programa, naudojanti tinklines sąsajas, naudoja *WSDL* tam, kad surastų paslaugų vietą ir taip būtų galima prisijungti prie jų.

SOAP pristatė *Microsoft* korporacija 1999-aisiais metais, dalyvaujant *IBM* tam, kad būtų praplėstas pasiūlymo mastas. 2000-ųjų metų kovą abi kompanijos pateikė 1.1 versiją konsorciumui – *W3C*. Šis jungtinis pateikimas labai padidino pasitikėjimą pramonės šakose, nes suteikė galimybių dirbti kartu su dviem šių organizacijų labiausiai paplitusiais kompiuteriniais modeliais, tokiais kaip *J2EE* ir *NET*.

BPEL specifikacija yra laikoma jau esamų tinklinių sąsajų standartų veiklos procesų išplėtimu. Anksčiau tinklinės sąsajos buvo apribotos iki neturinčių būklės sąveikų. *BPEL*, kiti procesai ir choreografijos kalbos rodo, kaip sukurti turinčius būklę, sąveikaujančius veiklos procesus tinklinėms sąsajoms. *BPEL* yra griežta kalba, kuri sukuria ir išplečia web paslaugas sąveikaujantiems procesams.

BPMN ryšys su *BPEL – BPMN* pateikia mechanizmą, kaip iš veiklos lygio notacijos sukurti vykdomą veiklos procesą. *BPMN* diagramos nebūtų tiek naudingos, jei jų nebūtų galima vykdyti. BPEL sąsaja su BPMN leidžia iš BPMN diagramų gauti BPEL kodą, o tai savo ruožtu suteikia galimybę vykdyti *BPMN BPEL* sistemose.

Gamintojai ir *BMPS* sprendimai

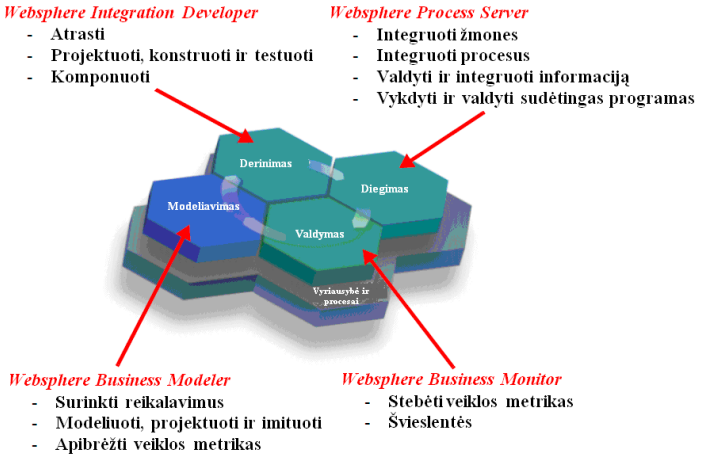
Rinkoje veiklos procesų valdymo sistemos jau siūlomos ne vieno gamintojo. Dauguma programinės įrangos kūrėjų savo *BPMS* sprendimus pateikia sudarytus iš kelių, jau praktiškai standartiniais tapusių komponentų.

Pagrindiniai veiklos procesų valdymo komponentai:

- *BPEL* redaktorius, arba programa, leidžianti grafiškai atvaizduoti įmonėje vykstančius procesus, jų tarpusavio sąveiką ir ryšius;
- veiklos logikos variklis – dažniausiai serveris, kuriame saugomi ir, reikalui esant, pritaikomi tam tikri sprendimai, išreikšti kaip taisyklių rinkiniai. Tokia struktūra leidžia sprendimus vykdyti automatizuotai, t. y. daug greičiau ir pigiau;
- veiklos proceso stebėjimo įrankiai. Vadovai ir sprendimų priėmėjai nuolat stebėdami vykstančius procesus gali kur kas greičiau identifikuoti bei eliminuoti iškilusias problemas;
- žmogiškųjų ir sisteminių resursų integratoriai į sistemą, palaikantys visus resursus sistemoje, taip pat užtikrinantys ryšį tarp atskirų komponentų.

Apžvelkime kelių tiekėjų konkrečius produktus.

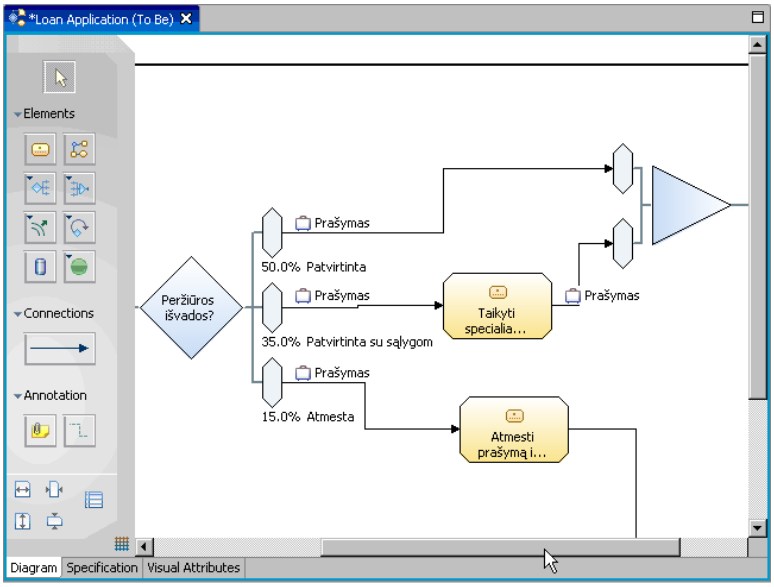
Didelę *BPMS* rinkos dalį užima *IBM* – pirmasis programinės įrangos infrastruktūros tiekėjas, siūlantis nuoseklų ir visą (angl. *end-to-end*) gyvavimo ciklo palaikymą (Silver 2006). *IBM WebSphere Business Suite* – tai 4 atskirų *IBM* produktų rinkinys (5 pav.).



5 pav. *IBM WebSphere Business Suite* komponentai (Wahli, et. al. 2006)

- *WebSphere Business Modeler* – atlieka *BPEL* redaktoriaus vaidmenį, jame simuliuojami procesai;
- *WebSphere Integration Developer* – atlieka integratoriaus vaidmenį;
- *WebSphere Process Server* – veiklos logikos serveris;
- *WebSphere Business Monitor* – veiklos proceso stebėjimo įrankiai.

Šis produktas remiasi *BPEL* ir į paslaugų teikimą orientuota architektūra – *SOA*. Nors *IBM WebSphere Business Suite* parduodamas kaip atskiras produktas, iš tiesų tai sprendimas, kurį sudaro kiti 4 produktai. Palaikomas visas (angl. *end-to-end*) gyvavimo ciklas, komponentai suskirstyti į atskirus etapus. Procesas pradedamas naudojant *IBM Business Modeler* komponentą, kuriame jau analizės metu (modeliuojant veiklos logiką, simuliuojant procesus) kuriamas vykdomasis modelis (6 pav.)



6 pav. *WebSphere Business Modeler* grafinė vartotojo sąsaja, skirta procesams modeliuoti ir veiklai simuliuoti

Microsoft taip pat siūlo produktų sąrašą, tačiau jie įvardino šį rinkinį kaip sprendimą, o ne kaip atskirą produktą.

- *Windows SharePoint* ir *Office SharePoint Server* – sprendimo integratorius.
- *BizTalk Server's BRE* - sprendimo veiklos logikos variklis.
- *BizTalk Server's BAM* kartu su *Microsoft Excel* bei *Office PerformancePoint BAM* informacijai atvaizduoti – veiklos proceso stebėjimo įrankis.
- *Microsoft Visio* – sprendimo redaktorius (Chappell 2007).

Oracle BPEL Process Manager – tai *BPEL* standartu paremtas produktas, kurį rinkai siūlo *Oracle*. Šis produktas, be abejo, yra lengvai suderinamas su kitais Oracle gaminiais, taip pat Oracle teigia, kad jų produktas lengvai suderinamas ir su *Microsoft*, *IBM* bei kitų gamintojų programine įranga (Oracle...2008).

Active Endpoints rinkai siūlo produktą *ActiveBPEL*. Šis produktas įdomus tuo, kad jo kodas – atviras visuomenei, o pats produktas nemokamas. Sudarytas jis iš dviejų dalių: serverio ir redaktoriaus, kuris, beje, sukurtas *Eclipse* bazėje (The ActiveBPEL...2008). *ActiveBPEL* gerokai kuklesnis už savo giminaitį *ActiveVOX*, už kurį naujasis savininkas jau prašo mokėti.

Išvados

Veiklos procesų valdymas – aktuali šių dienų metodologija, lėmusi naują požiūrį į veiklos procesą, „laužant" ribas tarp sprendimų, galimų tik atliekant atskiras funkcijas.

BPMS pagrindinis akcentas – vykdomieji modeliai (t. y. iš karto dvigubas modeliavimas – analitinis ir vykdomasis), sukuriami grafiškai ne programuojant kaip struktūrinė schema ir gali būti diegiami į procesą, kad automatizuotų, integruotų ir stebėtų visą (angl. *end-to-end*) procesą.

Kalbant apie *BPM* svarbūs yra standartai. Naudojami jau žinomi standartai, pavyzdžiui, *UML* (angl. *Unified Modeling Language*) buvo sukurtas visai kitam tikslui, bet naudojamas ir *BPM*. Be to, naujesnių technologijų naudojimas, tokių kaip tinklinių sąsajų (angl. *web services*) naudojimas *BPM* technologijoms, lėmė senesnių standartų tobulinimą ir naujų standartų kūrimą. *BPM* yra keli standartai, vieni yra geresni nei kiti. Geriausiai yra išsirinkti stipriausias jų dalis. Patogi adaptacija yra *BPEL* ir *BPMN*.

Visam procesui naudojami *BPMS* programiniai sprendimai turi valdymo organizavimo skirtumų priklausomai nuo gamintojo.

Literatūra

The ActiveBPEL Community Edition Engine [interaktyvus]. 2008 [žiūrėta 2008 m. balandžio 4 d.]. Prieiga per internetą: <http://www.activevos.com/community-open-source.php>.

Chappell D. 2007. *Microsoft and BPM: A Technology Overview*. Microsoft Corporation. USA.

Havey M. 2005. *Essential Business Process Modeling.* O'Reilly. USA.

Oracle Business Process Management [interaktyvus]. 2008 [žiūrėta 2008 m. balandžio 3 d.]. Prieiga per internetą: <http://www.oracle.com/technologies/bpm/index.html>.

Silver B. 2007. *New Directions in BPM Tools and Technology*. Bruce Silver Associates. USA.

Silver B. 2005. *Standards and the Process Lifecycle.* [interaktyvus]. Bruce Silver Associates [žiūrėta 2008 m. balandžio 5 d.]. Prieiga per internetą: <http://www.bpminstitute.org/articles/article/article/standards-and-the-process-lifecycle>.

Silver B. 2007. *The BPMS ValueProposition.* BPMInstitute.org. USA.

Silver B. 2006. *Understanding and Evaluating BPM Suites.* BPMInstitute.org. USA.

Wahli U., et. al. 2006. *WebSphere Process Integration V6: Business Process Management Modeling through Monitoring*. IBM Press. USA.

BUSINESS PROCESS MANAGEMENT

J.Škatulaitė, T.Rasulovas, J.Mikučionis

Summary

The study deals with Business Process Management (BPM) as a new management discipline which enables management of end-to-end processes that extends across organizational and system boundaries. We have discussed its relevance to today's business process. We have also discussed Business Process Management Suite (BPMS) and used standarts. In addition to this, we have overviewed some BPMS vendors and products.

„MOKSLAS – LIETUVOS ATEITIS“. 2009, Nr. 2 (2)
Informatika
Mokslinių straipsnių rinkinys ISSN 2029-2341 (print) / ISSN 2029-2252 (online)

APLIKACIJŲ REIKALAVIMŲ VALDYMAS

Julija Škatulaitė
Magistrantė,
Vilniaus Gedimino technikos universitetas,
juskada@yahoo.com

Anotacija. Nagrinėjamas informacinių technologijų projektų valdymas ir programinės įrangos reikalavimų valdymas. Aptariama reikalavimų svarba sėkmingam aplikacijų, atitinkančių galutinio naudotojo poreikius, kūrimui. Apžvelgiamas verslo analitiko vaidmuo ir verslo analizės procesas bei jo iššūkiai. Analizuojamas aplikacijų reikalavimų dokumentavimas naudojantis *Use Case* metodika ir veiklos procesų valdymo metodologija.

Reikšminiai žodžiai: reikalavimų analizė, verslo analitikas, reikalavimų dokumentavimas, *Use Case*, BPM, IT projektų valdymas.

Įvadas

Sėkmingam informacinių technologijų (IT) projektų įgyvendinimui didelę įtaką daro projektų valdymas. Tai ypač aktualu ilgalaikiams dideliems projektams. Su projektų valdymo disciplina (apimančia laiko, resursų, biudžeto, apimties valdymą) glaudžiai susijęs reikalavimų valdymas. Projektų nesėkmės priežastimi dažnai įvardijami prasti reikalavimai. Verslo analitikas identifikuoja verslo tikslus, kliento poreikį, planuoja funkcionalumą, funkcinius ir nefunkcinius reikalavimus aplikacijoms, jų svarbą aplikacijai, taigi kartu su projekto vadovu dalyvauja nustatant projekto apimtį. Projektui ir reikalavimų valdymo procesui svarbi dokumentavo metodika. Straipsnyje palyginsime reikalavimų valdymą su *Use Case* šablonu ir modeliavimu remiantis veiklos procesų valdymo (angl. *business process management,* BPM) metodologija.

Projektų ir reikalavimų valdymas

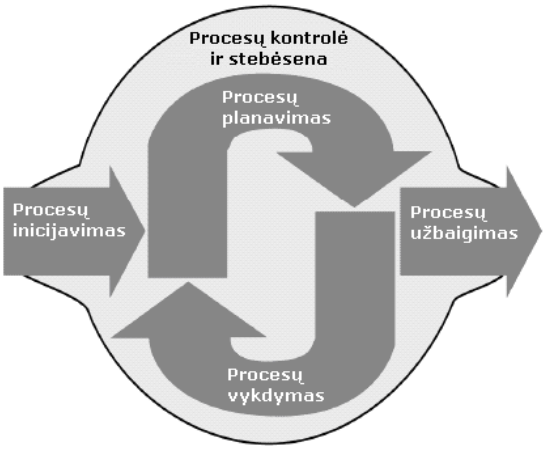
Kuriant programinę įrangą galima išskirti pagrindinius keturis etapus: reikalavimų analizė, realizacija, produkto testavimas ir kokybės užtikrinimas, rezultatų įvertinimas ir tobulinimas, kurių pagrindinis tikslas - sukurti galutinio naudotojo poreikius atitinkantį sprendimą, jį pristatatyti laiku, reikiamos kokybės ir nepažeisti biudžeto (1 pav.).



1 pav. Programinės įrangos kūrimas (*TechnoSolutions 2009*)

Galimi skirtingi organizavimai, taigi, projekto įgyvendinimo žinomas ne vienas modelis. Pvz., klasikiniu yra laikomas krioklio (angl. *waterfall*) modelis, kuriame visi etapai vyksta paeiliui. Praktikoje yra taikomas šiuolaikinei dinaminei visuomenei ypač tinkantis iteracinis modelis, kai darbas vykdomas fazėmis, tuo įgalinant bendradarbiavimą su klientu, lengvesnį apimties (angl. *scope*) ir pokyčių (angl. *change*) valdymą.

Projekto valdymo disciplina ypač aktuali dideliems projektams. Projekto valdymo procesas, susietas su planuoti-realizuoti-tikrinti-naudoti ciklu, pavaizduotas 2 pav. (PMI 2004).

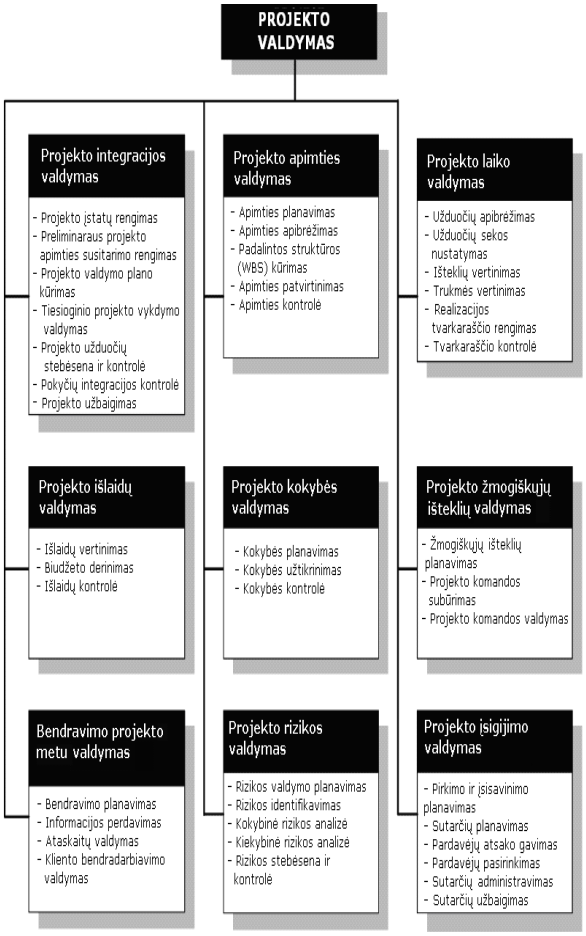


2 pav. Projekto valdymo procesas

Projektai yra koordinuojami projekto vadovo, kurio pirminės atsakomybės yra:

- planuoja ir organizuoja darbą;
- vadovauja kasdienėms projekto užduotims (angl. *day-to-day activities of a project*);
- pristato projekto sprendimą (angl. *project deliverables*) klientui;
- identifikuoja galimus klientus.

Projektų valdymo uždaviniai apima projekto koordinavimą įvairiais aspektais (3 pav.) (PMI 2004).



3 pav. Projektų valdymo sferos ir procesai

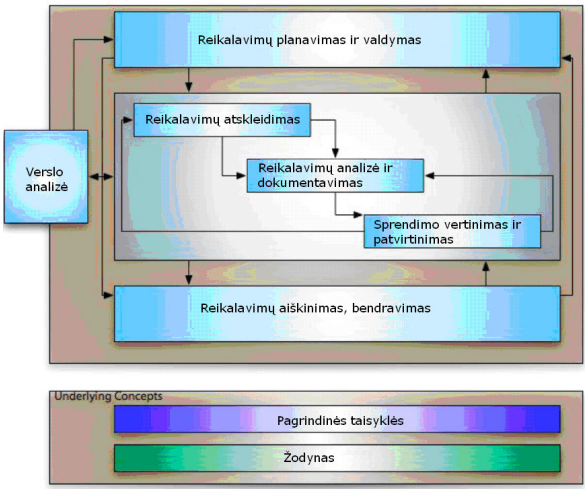
Valdsant projekto apimtį ir užtikrinant kokybę (ypač, jeigu testavimas yra organizuojamas remiantis dokumentacija ir už kokybę aspektu, kiek tai atitinka verslo ir naudotojo poreikį, atsako ne testuotojų padalinys) projekto valdymas kertasi su reikalavimų valdymu. Reikalavimų iššsaiškinimas ir tikslo suvokimas – pati pirminė bet kurio projekto stadija, kur dirba verslo analitikas. Verslo analitiko pirminės atsakomybės yra:

- efektyvus reikalavimų rinkimas, atskleidimas;
- aiškus, tikslus reikalavimų išreiškimas (dokumentavimas);
- reikalavimų koordinavimas tarp komandos narių ir kliento.

Viso projekto esmė – pristatyti laiku ir tinkamo biudžeto bei kokybišką sprendimą, atitinkantį verslo ir naudotojų poreikius. Būtent verslo ir naudotojų poreikiai yra nustatomi reikalavimais aplikacijai. Be to, laiku tinkamai suprasti ir organizuotai pateikiami bei

dokumentuojami reikalavimai įgalina kuriančiai komandai efektyvų darbą.

Reikalavimų valdymo procesas apima veiklos sferos suvokimą, vartojamą specifinį žodyną, verslo analizę (angl. *enterprise analysis*), reikalavimų atskleidimą (angl. *requirements elicitation*) ir analizę, dokumentavimą, sprendimo įvertinimą ir patvirtinimą (angl. *solution assessment and validation*), reikalavimų išaiškinimą komandos nariams - tiek kūrėjams, tiek naudotojams (4 pav.) (IIBA 2006).



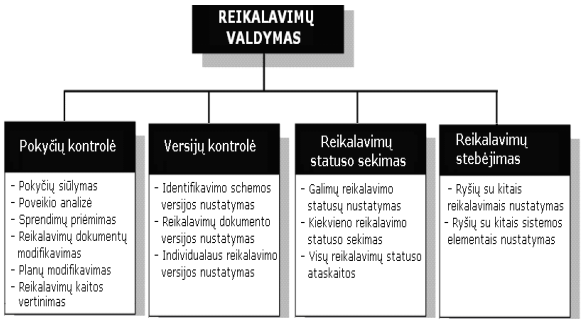
4 pav. Reikalavimų analizės proceso akcentai

Reikalavimų tipai:

- verslo (angl. *business*) – apibrėžia tikslus, kaip sistema paveiks veiklos sritį, verslą;
- naudotojų (angl. *user*) – apibūdina naudotojo poreikius ir problemas;
- funkciniai (angl. *functional*) – apibūdina sistemos elgseną, programinės įrangos darbą;
- kokybės (angl. *quality of service*) – taikomi visai sistemai ir yra „galimybės“, pvz., prieinamumas (angl. *accessibility*), greitaveika (angl. *performance*), keičiamumas (angl. *scalability*) ir kt.

Reikalavimų valdymo užduotys (5 pav.) (Wiegers 2003):

- pokyčių kontrolė reikalavimams;
- projekto planų pagal esamus reikalavimus palaikymas;
- versijų kontroliavimas (ir individualių reikalavimų, ir reikalavimų dokumentų);
- reikalavimų statuso projekte sekimas;
- loginių ryšių tarp individualių reikalavimų ir kitų veikiančio projekto produktų valdymas.



5 pav. Reikalavimų valdymo uždaviniai

Patogiam darbui su dokumentais būna naudojami ir įvairūs programiniai sprendimai, pvz., *Telelogic DOORS, Jira* ar kt., leidžiantys naudotojui lengvai ir greitai filtruoti, kurti ataskaitas reikiamais aspektais (atributais), rasti ryšius tarp duomenų ar dokumentų.

Reikalavimų valdymas su *Use Cases*

Use Case – naudojimo atvejis. Kitaip tariant, tai sąveikos tarp sistemos ir vieno ar daugiau veikėjų (angl. *actor*) aprašymas. Šis specifikavimo būdas yra grįstas naudojimu (angl. *usage-centric*), todėl dažnai palengvina bedradarbiavimą su verslo naudotojais, kuriems toks formatas lengviau suvokiamas. Karl‘o Wiegers‘o pateiktas *Use Case* šablonas pavaizduotas 6 paveiksle.

Use Case ID:	
Use Case Name:	
Created By:	Last Updated By:
Date Created:	Date Last Updated:

Actors:
Description:
Trigger:
Preconditions:
Postconditions:
Normal Flow:
Alternative Flows:
Exceptions:
Includes:
Priority:
Frequency of Use:
Business Rules:
Special Requirements:
Assumptions:
Notes and Issues:

6 pav. *Use Case* dokumentavimo šablonas (Wiegers 2006)

Naudojantis šablonu aprašomas *Use Case ID*, pavadinimas, sukūrimo ir paskutinio atnaujinimo data ir autorius. Ši informacija leidžia matyti reikalavimų kiekį, sekti jų atsiradimą ir keitimą. Šablone aprašomas

funkcionalumas, veikėjai, trigeriai, pradinės sąlygos, normali eiga, alternatyvūs atvejai, specialieji reikalavimai, verslo taisyklės, pastabos, išimtys, prielaidos, jei yra. Prioritetas parodo svarbą. Pasirinkus šį reikalavimų aprašymo metodą, analogiškai dokumentuojami ir papildomas funkcionalumas - *Extended Use Case*. Aprašant pagrindinę eigą ir alternatyvius atvejus juos patogu numeruoti ir įterpti stulpelį veikėjui, taip kiekvienam žingsniui matomas vykdytojas (asmuo ar sistema). Šablonas patogus bendraujant su verslo klientais, kuriems jis lengvai suprantamas, tačiau iš įgyvendinimo perspektyvos toks dokumentavimo būdas nėra pakankamai detalus. Dažnai kartu yra pateikiamos įvairios *UML* diagramos, pvz., sekos (angl. *sequence*), komponentų (angl. *component*) ir kt. funkcionalumui detalizuoti. Prie *Use Case* patogu turėti grafinės naudotojo sąsajos (angl. *graphical user interface*, GUI), naudojamos veiksmui įvykdyti, aprašymą. GUI pavyzdinis paveikslas ir elementų lentelė - lengvai suvokiamas pateikimo būdas verslo naudotojui ir vienareikšmis nurodymas įvykdymui. Vertinant reikiamą laiką ir analizuojant situaciją šis modelis nėra pakankamai informatyvus verslo analitikui ir projekto vadovui, kurie turi patys vertinti remdamiesi savo metodika ir patirtimi.

Reikalavimų valdymas su BPM metodologija

Jau keletą pastarųjų metų BPM įvaldymas ir tobulinimas kompanijoms yra prioritetas. BPM akcentuojamas privalumas tai, kad metodologija įgalina apdoroti (angl. *process*) daugiau ir aukštesne kokybe tuo pat metu įdedant mažiau pastangų. Pagrindinė nauda: našumas (angl. *efficiency*), efektyvumas, judrumas (angl. *agility*) (1 lentelė).

1 lentelė. Produktyvumo gerinimas įvairiose projekto fazėse (Lombardi 2008)

Tipinės BPM projekto fazės	% projekte	Produktyvumo pagerinimas
Funkciniai reikalavimai ir funkcinis dizainas	25%	50%
Realizacija (angl. *implementation*)	50%	20-25%
Kokybės užtikrinimas / testavimas	25%	30%
Verslo atskleidimas	N/A	N/A

Veiklos procesų valdymo komplektai (angl. *business process management suite*, BPMS) pristato verslo ir IT bendradarbiavimo paradigmą, modeliavimo įrankiai leidžia įvertinti esamą situaciją (angl. *as-is process*) ir nustatyti patobulintą (angl. *to-be process*) bei jas įvertinti be techninių realizacijos (angl. *implementation*) detalių. (Silver 2007).

Naudojant BPMS pirmasis etapas yra skirtas reikalavimų nustatymui ir apibrėžimui, naudojantis modeliavimo įrankiu. Šiame etape modeliuojama įvykių seka, svarbus reikalavimas, kad kiekvienas procesas turėtų trukmę, aiškią įvestį (angl. *input*) ir rezultatą (angl. *output*). (Wahli et. al. 2006). Kiekvienam procesui gali būti priskiriamas resursas, rolė, darbo laikas, tvarkaraštis, išlaidos. Modelis lengvai koreguojamas, galimas sekos ar proceso daugkartinis panaudojimas. Sukurtas modelis yra "vykdomasis" (angl. *executable*), t.y. galima „simuliuoti" (angl. *simulate*) veiklą ir matyti rezultatą. Ši savybė patogi sprendimo vertinimui ir verslo analitikui ieškant gerinimo galimybių, *to-be* modelių. Be to, lengvai generuojamos ataskaitos reikiamu aspektu. BPM'e naudojama BPEL kalba lengvai generuojama į UML. Nors pakeitimai greitesni, mažesnis dėmesys pokyčių sekimui ir kaitos registravimui, kas yra reikalinga reikalavimų valdymui. Naudojantis BPMS pirmame etape sukurtas modelis toliau naudojamas realizacijai, kitaip nei use case, kur dokumentaviams ir realizacija yra atskiri procesai.

Išvados

1. Laiku suprasti, teisingi, patogiai pateikiami ir lengvai valdomi reikalavimai vaidina svarbų vaidmenį projekto sėkmei, leisdami vykdančiai komandai tikslingai ir našiai dirbti bei rezultate pateikiant sprendimą, atitinkantį galutinio naudotojo poreikį.

2. Reikalavimų valdymas yra sudėtingas kelių etapų procesas, kurį galima organizuoti įvairiai. Funkciniams reikalavimams aplikacijoms aprašyti taikomi *Use Cases* ir UML. BPM metodologijoje reikalavimų analizė sukoncentruota į „vykdomųjų" modelių kūrimą. Abiem atvejais verslo analitikas gali rasti būdą efektyviai pateikti ir valdyti reikalavimus.

3. Kuriant dokumentą sprendimo funkcionalumui apibrėžti, rezultato aiškumui nustatyti šalia funkcinės eigos aprašymo (*Use Case* arba BPMN modeliu) patogu pateikti susietą GUI pavyzdį ir elementų lentelę su kiekvieno elemento aprašu.

4. Iš kompanijos perspektyvos: procesai organizuojami bei sprendimai, kokius įrankius naudoti, priimami vertinant išlaidų ir naudos santykį.

Literatūra

IIBA. 2006. *A Guide to Business Analysis Body of Knowledge v.1.6.*JAV.

Lombardi. 2008. *Making the Case for BPM. A Benefits Checklist.* JAV.

PMI. 2004. *A Guide to Project Management Body of Knowledge. Third Edition.* JAV.

Silver B.2007. *The BPMS ValueProposition. BPMInstitute.org*. JAV.

TechnoSolutions. 2009. *Why manage Requirements?* [interaktyvus]. [žiūrėta 2009 kovo 15 d.]. Prieiga per internetą: <http://www.technosolutions.com/topteam_requirements_management.html>

Wahli U., et. 2006. *WebSphere Process Integration V6: Business Process Management Modeling through Monitoring*. JAV.

Wiegers K.E.. 2003. *Software Requirements, Second Edition*. JAV.

Wiegers K.E. 2006. *More About Software Requirements: Thorny Issues and Practical Advice*. JAV.

APPLICATION REQUIREMENTS MASTERING

Julija Skatulaite

Summary

The study deals with requirements definition and management which is the key process for successful projects. We have reviewed a role and impact of requirements to a project management and project plan. We have also discussed a requirements process, its activities and business analysts' challenges. In addition to this, we have overviewed requirements documentation with Use Cases and UML and compared it with requirements definition using BPM methodology.