

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Patikslintas kritinės grandinės metodas programų
sistemų projektams esant išteklių neapibrėžtumui**

**A specified Critical Chain method for software projects under
resource allocation uncertainty**

Magistro baigiamasis darbas

Atliko: Lina Povilavičiūtė

Darbo vadovas: lekt. Andrius Adamonis

Recenzentė: lekt. dr. Asta Vaitkevičienė

Vilnius – 2016

Santrauka

Projektų įgyvendinimas laiku vis dar išlieka aktualus uždavinys projektų vadyboje. Šis uždavinys tampa sudėtingesnis, atsižvelgiant į neapibrėžtumą, su kuriuo projekto eigoje yra susiduriama. Neapibrėžtumo problemai spręsti pasiūlytas kritinės grandinės metodas palieka papildomų „saugiklių“, kurie turėtų kompensuoti atskirų užduočių vėlavimą dėl nenumatytų problemų. Didelės sėkmės susilaukęs metodas į projekto užduočių tinklo modelį įtraukia išteklių priklausomybes. Nepaisant to, kritinės grandinės metodas nepakankamą dėmesį skiria išteklių paskirstymui užduotims ir nesuteikia konkrečių detalių, kaip spręsti kylančius išteklių priklausomybių konfliktus, sudarant projektų tvarkaraštį pagal šį metodą. Nėra aišku, kaip pasinaudoti buferiais, jei užduotys vėluotų dėl tam tikros kompetencijos stokos ir kaip nepilnai apkrauti nekritiniai išteklių (išteklių buferis) spręstų šią problemą. Šiame darbe yra stengiamasi detaliau išnagrinėti išteklių klausimą kritinės grandinės kontekste taip, kad planuojant projektą būtų galima geriau spręsti išteklių neapibrėžtumo problemą. Darbe apibrėžiamas konkretus kritinės grandinės taikymas programų sistemų projektams, atsižvelgiant į turimą informaciją apie išteklių kompetencijas ir užduočių savybes. Tokiu būdu patikslintas įprastinis kritinės grandinės metodas yra pritaikomas konkrečiam projektui. Žingsnis po žingsnio yra aprašoma specifika, kaip taikyti metodą tvarkaraščio sudarymo etape; kaip spręsti išteklių paskirstymą užduotims ir jų priklausomybių konfliktus, ir kaip išsirinkti geriausią alternatyvą siekiant mažiausios projekto trukmės. Išskirti nauji elementai, kuo reikėtų pasinaudoti, taikant kritinę grandinę projektui planuoti, leidžia lanksčiai modeliuoti projekto užduočių tinklą ir atsižvelgti į galimą išteklių neapibrėžtumą tam, kad numatytais buferiais būtų galima iš tiesų pasinaudoti.

Summary

Delivering projects on time is still an important task in project management. This task becomes more difficult if uncertainty is taken into account. The issue of uncertainty can be tackled by Critical Chain method which establishes buffers in order to compensate late delivery of separate tasks which occurs due to unforeseen problems. This method received much credit for involving resource constraints in project network model. Nevertheless, Critical Chain method lacks concrete details on resource allocation and does not provide the details how to solve resource dependency conflicts during scheduling based on this method. It is not clear how to use the buffers if tasks were late due to insufficient resource competence and how in particular non-critical resources (resource buffer) could solve this problem. In this work we analyse resource issue in more detail in Critical Chain context in such a way that during project planning resource allocation uncertainty could be solved more effectively. In this work, a specified Critical Chain application is defined for software projects, taking initial information about resource competences and task properties into account. A specified Critical Chain method is applied on a concrete project. The method application specifics are described step-by-step, explanations are provided of how to solve resource dependency conflicts and how to choose the best alternative to have the shortest possible project duration. New elements are defined for consideration when applying Critical Chain on project scheduling which makes project network modelling more flexible and respects resource allocation uncertainty so that buffers can really be used.

Turinys

ĮVADAS	5
1. Projekto planavimo sąvokos ir terminologija	9
1.1. Projekto užduočių tinklas	9
1.2. Kritinis kelias ir kritinė grandinė	10
1.3. Projekto užduočių vertinimas	10
1.4. Buferių valdymas	12
2. Neapibrėžtumo problema projektų vadyboje	13
2.1. Neapibrėžtumo rūšys pagal jo prigimtį	14
2.2. Neapibrėžtumas projektų planavime	15
2.3. Skirtingi neapibrėžtumo problemos sprendimo požiūriai	15
2.3.1. Užduočių vertinimas ir neapibrėžtumas	16
2.3.2. Neapibrėžtumo valdymas skirtingais proceso valdymo modeliais	17
2.3.3. Projektų planavimo metodai esant neapibrėžtumui	20
2.3.4. Kritinės grandinės technika ir jos specifika	23
3. Darbo metodika	25
4. Kritinės grandinės metodo tikslinimas	27
4.1. Naudojamų sąvokų interpretavimas ir prielaidos	27
4.2. Tvarkaraščio sudarymo metodas	30
5. Metodo taikymas projekte	32
5.1. Pradiniai duomenys	32
5.2. Duomenų interpretavimas	33
5.3. Užduočių perplanavimas	36
5.4. Išteklių optimizavimas	37
5.5. Buferių pritaikymas	42
REZULTATAI IR IŠVADOS	46
ŠALTINIAI	48

ĮVADAS

Populiariausios projektų valdymo technikos, padedančios planuoti projektus, daro prielaidas, kad tam tikri projektų parametrai yra iš anksto žinomi ir suplanuoti: yra žinomas projekto užduočių tinklas (angl. *project network*)¹ – numatomos užduotys ir jų trukmės, tam tikru tikslumu apibrėžtas biudžetas ir viso projekto trukmė. Tokiomis sąlygomis pagrindinis projektų vadovų uždavinys yra paskirstyti išteklius taip, kad projektas būtų įvykdytas pagal numatytą biudžetą ir laiką. Tokiam uždaviniui spręsti dar 1950 metais sukurta *kritinio kelio* projekto planavimo technika (angl. *critical path method*), kuria vadovaujantis yra sudaromas projekto tvarkaraštis, identifikuojant kritines užduotis, kurios negali vėluoti siekiant baigti projektą laiku [JJ10], bet neatsižvelgiant į žmogiškųjų išteklių apribojimus ar neapibrėžtą užduočių trukmę [Hal12].

Šių dienų projektų specifika (naudojamų technologijų kompleksiskumas, išaugę klientų poreikiai, sutrumpėjęs pageidaujamas projekto laikas ir padidėjęs projekto lankstumo poreikis) dažnai sukuria tokias aplinkybes, kad tikslus projekto užduočių tinklas iš anksto nėra žinomas, o jei ir žinomas, tai apytikris – projekto eigoje užduočių skaičius ir trukmė smarkiai keičiasi. Atitinkamai, projekto eigoje susidūrus su nenumatytais problemomis ar pasikeitus sistemos reikalavimams, iš anksto planuotas išteklių kiekis gali nebeatitikti realaus poreikio ir siekiant išpildyti projekto tikslus prireikia imtis tam tikrų projekto skatinimo priemonių (angl. *crashing*) [Mic08]. Šios priemonės padeda sumažinti projekto vėlavimo ir su tuo susijusių kaštų riziką, sumažinant numatytą laiką skirtą užduotims įvykdyti, ir gali būti nagrinėjamos kaip paprastas tiesinio programavimo uždavinys [Hal12]. Vis dėlto tokie sprendimai reikalauja papildomų finansinių ir žmogiškųjų išteklių, kurie nebuvo numatyti projekto planavimo etape. Be to, Nicolas G. Hall [Hal12] teigimu jie tinka projekto tinklui, kai užduočių trukmės yra žinomos ir tampa komplikuoti, kai užduočių trukmės yra neapibrėžtos.

Neapibrėžtumo tematika projektų valdyme pradėta nagrinėti dar antroje XX a. pusėje PERT technika [Faz59], kuri kaip ir *kritinio kelio* planavimo technika, nagrinėja projekto eigą pagal projektinių užduočių tinklą, bet įgalina planuoti, nežinant tikslų užduočių trukmių, o remiantis statistiniais trukmių dydžiais. Kita technika yra *Monte Carlo* simuliacija, kuri remiasi tikimybinio projekto užduočių trukmės pasiskirstymu. Tačiau projektų vadovai dažnai ne tik nežino, kiek truks viena ar kita individuali užduotis, bet nėra žinomas net ir tikimybinis užduoties trukmės pasiskirstymas [Hal12], dėl to tokios technikos gali būti sunkiai įgyvendinamos. Problemai spręsti

¹ Terminas plačiai naudojamas projektų valdymo literatūros šaltiniuose. Projekto užduočių tinklas yra PERT technikos pagrindas, kur terminas formuluojamas kaip projekto plano grafinė reprezentacija, vaizduojanti projekto uždavinius, jų tarpusavio ryšius, priklausomybes tarp projekto elementų ir jų prioritetus [Faz59].

galima naudoti tam tikras *stabilumo* optimizavimo technikas [BPS04], taip pat *patenkinčiosius* (angl. *satisficing*) modelius [GSF98], kurie remiasi *sprendimų teorija* ir kurių tikslas nėra rasti geriausią variantą, bet nustatyti sprendimą, tenkinantį iškeltus kriterijus tam tikrose ribose, laikant jog optimalaus sprendimo duotoje situacijoje negalima rasti.

Projekto planavimo uždavinį neapibrėžtumo sąlygomis mėgina spręsti pastarąjį dešimtmetį atsiradusios judriosios projektų valdymo metodikos [HDK12], tačiau jų sprendimas dažnai grindžiamas apimties buferio valdymu (apimties buferis slepiasi mažo prioriteto užduotyse, kurios gali būti atmetamos, jeigu nepakanka laiko arba lėšų visai projekto apimčiai įgyvendinti). Tačiau judriosios metodikos neduoda sprendimo, kaip iš anksto, neturint tikslios projekto apimties, nusistatyti projekto laiko buferius. E. Goldratt *apribojimų teorija* ir ja pagrįsta *kritinės grandinės* (angl. *critical chain*) projekto planavimo technika [SG09] leidžia susiskaičiuoti laiko buferį projekte, tačiau vėlgi – reikalauja, kad projekto užduočių tinklas būtų iš anksto žinomas. Technika, kuri pasiremtų abiem požiūriais ir leistų gauti abiejų požiūrių teikiamas naudas, smarkiai palengvintų projektų planavimą neapibrėžtumo sąlygomis.

Neapibrėžtumo problematika ypač aktuali inovatyviems projektams, kur inovatyvumo lygis koreliuoja su neapibrėžtumu. De Fatima S., Sbragia R., Sin Oih Yu [FSS13] įvardija daug faktorių neapibrėžtumui atsirasti: aplinka, klientai, tiekėjai, partneriai, rinkos situacija, organizacijos struktūrizacijos trūkumas, komunikacijos, planavimo ir kitos problemos.

Informacinių sistemų projektuose, kur inovatyvumo lygis yra aukštas, neapibrėžtumas yra kasdienybė. Pagal [MSM13] tik 32% projektų yra baigiami laiku, todėl sprendimai, padedantys išvengti projektų vėlavimo, yra itin aktualūs. Magistro darbo **tyrimo objektas** – programų sistemų projektai, kai realizavimo etapo užduočių kiekis bei jų trukmė nėra tiksliai apibrėžti ir laikoma, jog projekto eigoje užduotys, jų trukmės gali keistis, o kartu – ir išteklių poreikis. Darbe sprendžiamas uždavinys, kaip sumodeliuoti projekto užduočių tinklą ir planą bei turimų išteklių paskirstymą užduotims atlikti, kai žinomi apytiksliai būsimų užduočių trukmės įverčiai ir pradiniai numatomi ištekliai, kad būtų galima efektyviau prognozuoti projekto pabaigos terminą.

Šiame darbe siekiama sukurti metodą, leisiantį efektyviau spręsti projekto tvarkaraščio sudarymo problemą, esant išteklių neapibrėžtumui. Sprendimas remsis Goldratt *kritinės grandinės* technika. Šis metodas literatūroje pristatomas kaip naujesnis, sėkmingesnis ir artimesnis realioms projektams planuoti nei kritinio kelio metodas. Tokia sėkmė susijusi su žmogiškųjų išteklių – kaip dar vieno apribojimo – įtraukimu į projekto užduočių tinklo sudarymo modelį.

Vis dėlto kritinės grandinės metode nepakankamas dėmesys yra skiriamas išteklių priskyrimui projekte. Čia nėra nagrinėjami atvejai, kai užduotims ar projektams reikia daugiau nei vieno to paties tipo išteklių [HR01], [FLY+11]. Taip pat nėra detalios apibrėžiamos, kaip tiksliai spręsti išteklių konfliktus, pridėjus „maitinimo“ buferius projekto tvarkaraščio sudarymo eigoje

[HR01]. Nėra aišku ir tai, kaip pasinaudoti „maitinimo“ ar išteklių buferiais, jei užduotys vėluotų dėl tam tikros kompetencijos stokos. Turint nepilnai apkrautus nekritinius išteklius (Goldratt vadinamą išteklių buferį), bet neturint reikalingų gebėjimų, šie ištekliai neišspręs minėtos problemos.

Darbo tikslas ir uždaviniai

Atsižvelgiant į aktualiausius uždavinius pagal nagrinėjamos tematikos esmę, šio **magistro darbo tikslas** – apibrėžti, kaip taikyti kritinės grandinės metodą programų sistemų projektų tvarkaraščių sudarymui, esant išteklių neapibrėžtumui. Jeigu kritinės grandinės metodas gali būti sprendimas įvairiems projektams, susiduriantiems su bet koku neapibrėžtumu, tai šiuo darbu siekiame susiaurinti tyrimo sritį ir ieškoti sprendimo konkrečiam neapibrėžtumui spręsti būtent programų sistemų projektuose.

Išteklių neapibrėžtumas gali kilti dėl nenumatytų išteklių kiekio pokyčių projekte, neapibrėžto išteklių produktyvumo, neapibrėžto kvalifikacijos poreikio užduotims atlikti, sąlygoto užduočių vertinimo ir kitokio pobūdžio neapibrėžtumo. Šiuo darbu siekiama detaliai atskleisti vieno iš kertinių kritinės grandinės aspektų – išteklių apribojimo – ypatumus ir pateikti specifinį šio metodo taikymą programų sistemų projektų tvarkaraščio sudarymui, tuo patikslinant kritinės grandinės metodą.

Darbo tikslui pasiekti išskirti tokie **uždaviniai**:

1. Išanalizuoti metodus ir technikas, taikomus planuojant projekto vykdymą, atsižvelgiant į neapibrėžtumą projekto eigoje, ir išskirti šių metodų ypatumus ir taikymo aspektus.
2. Nustatyti patikslinto kritinės grandinės metodo taikymo apribojimus ir prielaidas, kylančias iš programų sistemų projektų specifikos: darbo užduočių laiko sąnaudų vertinimo, išteklių komplektavimo ar jų kvalifikacijos sąryšio su užduotimis.
3. Apibrėžti kritine grandine paremto projektų tvarkaraščio sudarymo metodą – veiksmų seką, nusakančią projektų tvarkaraščio sudarymo ir optimizavimo žingsnius, aprašančius, kaip efektyviai suplanuoti turimus išteklius projekto užduotims įgyvendinti, pasinaudojant išteklių ir užduočių specifikos galimybėmis ir atsižvelgiant į jų apribojimus.
4. Atlikti atvejo analizę, pritaikant patikslintą metodą realiam projektui planuoti ir nustatyti jo taikymo efektyvumą.

Darbo rezultatas

Šio **darbo rezultatas** – kritinės grandinės metodo patikslinimas išteklių neapibrėžtumui spręsti programų sistemų projektuose – išdėstytas ketvirtajame skyriuje. Šioje darbo dalyje pateikta konkreti veiksmų seka, reikalinga planuojant projektą nagrinėjamos problematikos kontekste; aprašyta, į kokius duomenis apie išteklius ir užduotis reikia atsižvelgti ir kaip juos panaudoti efektyviam projekto tvarkaraščio sudarymui.

Kitos baigiamojo darbo dalys yra tokios:

Pirmoji dalis paaiškina pagrindines sąvokas ir terminologiją, būtiną atskleisti nagrinėjamos tematikos esmę.

Antrame darbo skyriuje išdėstyta neapibrėžtumo problematika projektuose ir kaip neapibrėžtumas gali būti interpretuojamas iš skirtingų perspektyvų. Aptariami neapibrėžtumo sprendimo būdai ir atskleidžiamos projektų planavimo technikos ir metodai, orientuoti spręsti neapibrėžtumo problemą projektuose.

Trečiame skyriuje aprašoma, kaip buvo apibrėžtas programų sistemų projektų planavimo metodas, kokie reikalavimai iškelti, ir kokiomis prielaidomis ir apribojimais pasižymi kritinės grandinės metodo patikslinimas, kurių ribose jis turi veikti.

Penktame skyriuje pateikta konkretaus projekto atvejo analizė ir specifikuoto planavimo metodo taikymas šiam projektui planuoti. Ten plačiai paaiškinti metodo taikymo ypatumai.

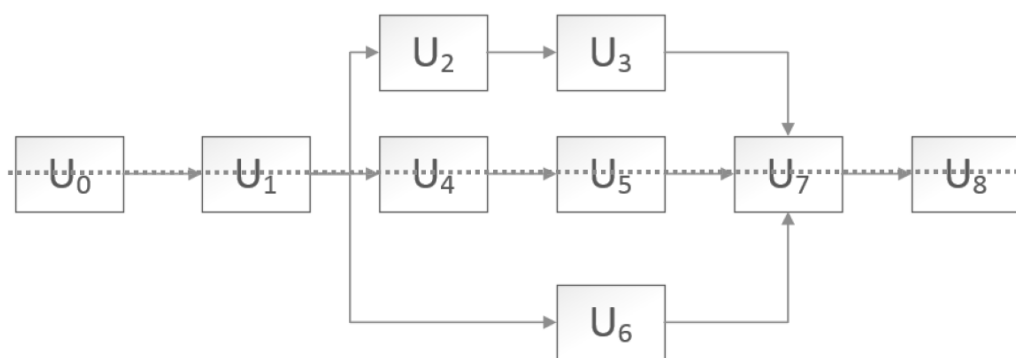
Pabaigoje pateikiamos magistrinio darbo išvados ir aptariami pasiekti rezultatai.

1. Projekto planavimo sąvokos ir terminologija

1.1. Projekto užduočių tinklas

Planuojant projektą, projektas yra suskaidomas vykdymo etapais, kurie savo ruožtu skaidomi į projektines užduotis. Žinant, kokios projekte užduotys numatomos, galima sudarinėti projekto tvarkaraštį, nurodantį, kada ir kokios užduotys yra planuojamos atlikti, kas jas turi atlikti ir kada vėliausiai užduotis gali būti pradėta, nepažeidžiant bendro projekto termino.

Kad būtų paprasčiau įsivaizduoti, kokios užduotys turės būti atliktos ir kad būtų galima aiškiai matyti užduočių tarpusavio sąryšius, projektų planavimo literatūroje dažnai naudojamas *projekto užduočių tinklas*. Ši sąvoka pradėta naudoti PERT technikoje (angl. Program Evaluation and Review Technique). *Projekto užduočių tinklas* yra projekto plano grafinė reprezentacija, vaizduojanti projekto uždavinius, jų tarpusavio ryšius, priklausomybes tarp projekto elementų ir jų prioritetus [Faz59]. Turint tokį užduočių tinklą (1 pav.) galima nesunkiai pamatyti, kokia užduočių seka turi būti būtinai įvykdyta laiku, kad visas projektas būtų įvykdytas laiku. Toks užduočių „kelias“ vadinamas kritiniu keliu ir žymi ilgiausią užduočių seką arba daugiausiai laiko naudojančią seką projekto tinkle [Faz59]. Į kritinį kelią projekto vadovas turi sutelkti daugiausia dėmesio, stebint projekto eigą ir siekiant, kad projektas būtų baigtas laiku. Projekto eigoje keičiantis projekto planui, kritinis kelias taip pat keičiasi.

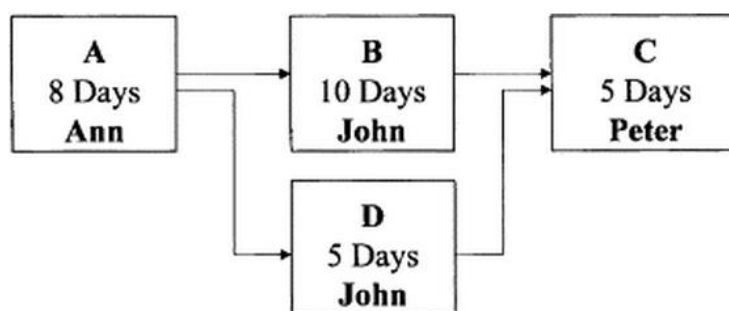


1 pav. Projekto užduočių tinklo pavyzdys. (Taškinė linija žymi galimą kritinį kelią.)

Toliau planuojant projekto tvarkaraštį projektų vadovas turi atsižvelgti į tokius apribojimus, kaip personalo, projekto biudžeto ir laiko.

1.2. Kritinis kelias ir kritinė grandinė

Siekiant papildyti PERT techniką, 1997 pasiūlytas kritinės grandinės metodas. Jo autorius Goldratt pasiūlė į projekto užduočių tinklą žiūrėti ne tik kaip į užduotis ir jų tarpusavio priklausomybes, bet kaip į užduočių tinklą, kurio struktūra brėžiama pagal išteklių priklausomybes. Pagal PERT techniką, pirma yra nubrėžiamas projekto užduočių tinklas ir tik po to atsižvelgiama į išteklių ribojimus, tuo tarpu Goldratt siūlo iš karto atsižvelgti į žmogiškųjų išteklių apribojimus [Ste01]. Tokiu būdu atsirado sąvoka *kritinė grandinė*, kuri iš esmės žymi tą pačią projekto užduočių seką, kuriai suvėlavus, visas projektas vėluos, tačiau kyla iš skirtingų požiūrių ir dėl to gali skirtis nuo kritinio kelio (2 pav.).



2 pav. Kritinio kelio ir kritinės grandinės skirtumai.

Šaltinis: [Ste01]

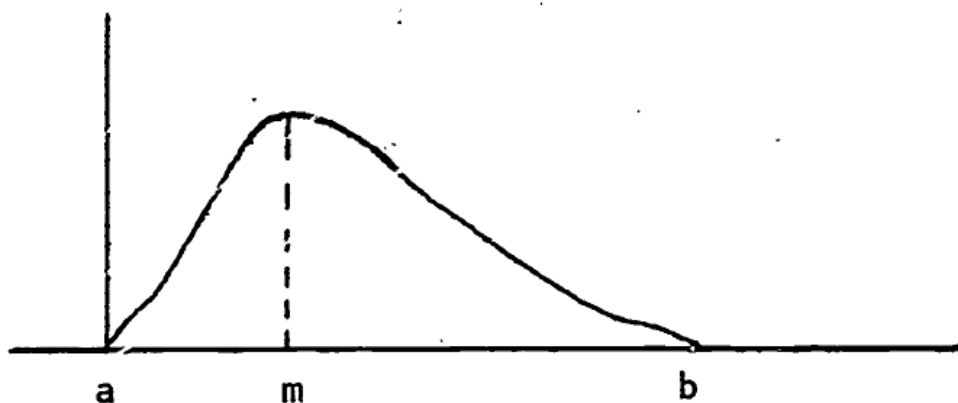
Paveiksle matyti, jog kritinis kelias yra užduočių seka ABC (laiko prasme ilgiausia užduočių seka), o kritinė grandinė yra ABDC arba ADBC. Tas pats žmogus gali atlikti kelias skirtingas užduotis, tačiau vienu metu jis negali daryti daugiau nei vienos užduoties.

1.3. Projekto užduočių vertinimas

Tam, kad būtų galima detaliai planuoti projekto eigą ir sudaryti projekto tvarkaraštį, reikia nustatyti projekto užduočių laiko įverčius. Galima išskirti tris dažnai naudojamas vertinimo technikas [Sha14]: eksperto nuomonė, analogija ir disagregacija. Pirmoji technika remiasi patyrusio specialisto vertinimu, antroji – patirtimi, kai nauja užduotis vertinama tiek, kiek praeityje užtruko atlikti panašią užduotį. Disagregacijos technika naudojama užduotį suskaidant į sub-užduotis ir atskirai vertinant šias, o paskui sudėjus jų vertinimus.

Vertinimui turi įtakos keli faktoriai: kaip daug žinoma apie pačią užduotį – neapibrėžtumo lygis, ar užduotis yra nauja ar jau vykdyta ankstesniuose projektuose ar etapuose. PERT siūlo naudoti 3 skirtingus vertinimus: labiausiai tikėtinas užduoties įvykdymo laikas, optimistinis laikas

ir pesimistinis laikas [Faz59]. Labiausiai tikėtinas laikas dažniausiai remiasi vertintojo patirtimi normaliomis sąlygomis. Optimistinis vertinimas nustato užduoties įvykdymo laiką pagal tai, kaip greičiausiai užduotis bus įvykdyta, esant optimalioms sąlygoms. Pesimistinis vertinimas remiasi laiko trukme, kurios prireiks užduočiai atlikti, jei įvyks patys blogiausi įsivaizduojami įvykiai. Tokį trijų požiūrių vertinimą galima pavaizduoti beta kreive (3 pav.), kur m – labiausiai tikėtina užduoties trukmė, a – optimistinė, o b – pesimistinė.



3 pav. Beta pasiskirstymas trims laiko vertinimams.

Šaltinis: [Faz59]

Kritinės grandinės technika taip pat daro prielaidą, kad užduočių vertinime kiekvienas aukštesnio lygio vadovas prideda papildomą laiko dydį prie įvertintos užduoties [Ste01]. Jeigu tarsime, kad projektų vadovas prideda 10 % papildomo laiko prie specialisto įvertintos užduoties, tuomet esant n lygių organizacinei struktūrai, galutinė įvertinta užduotis sudarys $(1,1)^n$ laiko. Jeigu organizacija turėtų 5 hierarchinius lygius, užduočiai skirtas papildomas laikas neapibrėžtumui padengti sudarytų daugiau kaip 60 % užduoties laiko.

Judriųjų metodikų – tokių kaip Scrum, Ekstremalus programavimas, Lean – užduočių vertinimui naudojama pastangų vertinimas (angl. effort estimation) ir istorijų taškai (angl. story points). Pavyzdžiui, Daniel R. Greening [Gre10] įmonėje naudoja *Enterprise Scrum* metodą, kur vertinant užduotis įtraukiami aukščiausio lygio vadovai. Toks vertinimas vykdomas etapais: pirma užduočių vertinimą atlieka produkto valdybos nariai (produktų vadovai, vyriausiasis architektas, verslo sprendimų direktorius), paskui tas pačias užduotis vertina komandos lyderis, pati komanda ir galiausiai paskiriamos komandos vykdyti tam tikras užduotis. Kiekviename vertinimo žingsnyje vertintojai nežino aukštesnio hierarchinio lygio vertinimų, o pastarieji gauna žemesnio lygio vertinimus ir gali koreguoti savus vertinimus atitinkamai. Tik žemiausiame lygyje – komandose – vertinimas vyksta istorijos taškais.

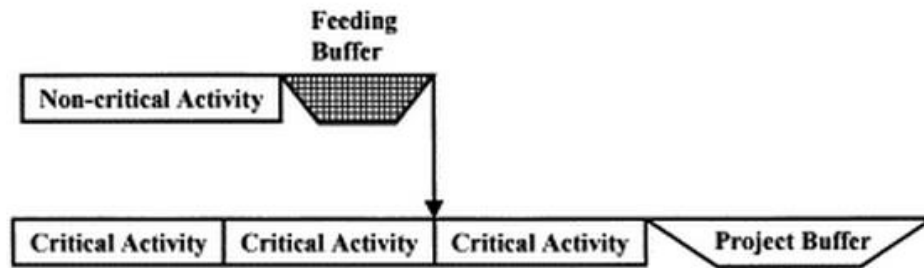
Judriuosiuose metoduose taip pat naudojamos tokios metrikos kaip užduočių atlikimo greitis, kuriuo remiantis planuojamos kitos iteracijos arba prognozuojamas numatytų užduočių iteracijoje įgyvendinamumas. Užduočių atlikimo greitis praėjusiose iteracijose gali būti naudojamas ir minimalių (privalomų) reikalavimų skaičiaus nustatymui [KMI15].

1.4. Buferių valdymas

Projektų planavime ir užduočių vertinime dažnai naudojami taip vadinami buferiai (angl. *buffers*) – papildomas laiko kiekis užduočiai įvykdyti, siekiant apsisaugoti nuo netikėtų vėlavimų atliekant užduotį. PERT technika termino *buferis* nenaudoja, tačiau identifikuoja laisvai kintančią užduoties trukmės dalį (angl. *slack time*). „Laisva trukmė“ šioje technikoje reiškia skirtumą tarp didžiausio leistino laiko užduočiai įvykdyti (gali būti išskaičiuojamas iš projekto užduočių tinklo arba formaliai įvertintas, pavyzdžiui, suformuotas vadovybės) ir mažiausio tikėtino užduoties atlikimo laiko. Šis dydis gali būti ir neigiamas, o kritinis kelias gali būti išreikštas kaip seka užduočių, kurios sumoje turi mažiausią teigiamą „laisvą trukmę“ arba didžiausią neigiamą [Faz59]. Pagal PERT „laisva trukmė“ galima manipuliuoti projekto užduočių tvarkaraštį, perkeltiant šį dydį kitai užduočiai ar padalinant „laisvos trukmės“ dalis kelioms užduotims, atsižvelgiant į išteklių apribojimus ar kitas sąlygas.

Tuo tarpu kritinės grandinės technika naudoja buferius – dirbtinai pridedamą laiką prie užduoties atlikimo trukmės – ir išskiria trijų tipų buferius [Gol97]:

1. Projekto buferis. Šis buferis pridedamas prie projekto užduočių tinklo pabaigos tam, kad kritinės grandinės užduotims būtų galima turėti papildomo laiko vėlavimų atveju siekiant, kad bendras projekto terminas liktų nepakitęs.
2. Maitinimo buferis. Buferiai pridėti prie nekritinių užduočių, siekiant apsaugoti kritines užduotis (kritines – esančias kritiniame kelyje) nuo vėlavimų (4 pav.). Tokie buferiai pridedami, kai nekritinės užduotys planuojamos atlikti kaip įmanoma vėliau, siekiant kuo didesnio jų aiškumo, maitinimo buferiais apsaugant kritines užduotis nuo nekritinių užduočių vėlavimų [Ste01].
3. Išteklių buferis. Dalis žmogiškųjų išteklių projekto plane numatomi ne pilno užimtumo, kad, įvykus nenumatytiems nuokrypiams nuo plano, kaip įmanoma greičiau jie galėtų prisidėti prie kritinės grandinės užduočių vykdymo.



4 pav. Maitinimo buferis, skirtas apsaugoti kritines užduotis nuo vėlavimų, jei nekritinė užduotis vėluos.

Šaltinis: [Ste01]

2. Neapibrėžtumo problema projektų vadyboje

Kiekvienas projektas tam tikru metu susiduria su neapibrėžtumu. Tai dažniausiai nutinka planavimo laikotarpyje, kai užsakovui reikia pateikti tam tikrus laiko, apimtį ir kainos įverčius. Dažnai neapibrėžtumas tapatinamas su projekto rizikomis, todėl kyla klausimas, kuo skiriasi šios dvi sąvokos.

Pagal [Cle09] rizika turi kelis atributus:

- reikia žinoti rizikos poveikį (dažniausiai grėsmę);
- rizika gali būti kiekybiškai išreikšta, t. y. jai priskirtas tikimybinis galimybės įvykti įvertis;
- rizika išskiria projekto pažeidžiamas vietas;
- identifikuojama rizika įgalina turėti jos valdymo veiksmų planą.

Pagal šį požiūrį neapibrėžtumas yra visa tai, kas lieka identifikavus rizikas. Jeigu riziką galėtume vadinti „žinoma nežinomybe“, tai neapibrėžtumą – „nežinoma nežinomybe“.

Rengiant projektų planą ir jo metu identifikuojant rizikas dalis pradinio neapibrėžtumo – dar kitaip vadinamo „įgimtu“ neapibrėžtumu (angl. *inherent uncertainty*) – virsta rizikomis, o kita dalis lieka nežinomybe. Likusią nežinomybę [Cle09] vadina „paslėptu“ neapibrėžtumu (angl. *latent uncertainty*). Tokiu atveju rizikų valdymas kaip viena iš projekto valdymo dedamųjų mažina neapibrėžtumą, transformuodama jį į rizikas, tačiau negali valdyti likusio neapibrėžtumo. Dažnai klaidingai manoma, kad projekto plane identifiкуotos rizikos rodo visų nežinomųjų įvertinimą ir nėra skiriama pakankamai priemonių netikėtai atsiradusioms problemoms suvaldyti [Cle09].

2.1. Neapibrėžtumo rūšys pagal jo prigimtį

Neapibrėžtumas projektuose gali atsirasti dėl daugelio priežasčių. Pagal skirtingą jo prigimtį galima kalbėti apie kelių tipų neapibrėžtumą [MSM13]: rinkos neapibrėžtumą, technologinį neapibrėžtumą, aplinkos neapibrėžtumą ir socialinį-žmogaus (angl. socio-human) neapibrėžtumą.

Rinkos neapibrėžtumas susijęs su išorine nežinomybe, koks produktas patiks rinkai ir dėl to sunkiau apibrėžiamu projekto tikslu. Toks neapibrėžtumas svarbus naujų produktų kūrimo projektams ir gali būti mažinamas atliekant detalesnę rinkos analizę.

Technologinis neapibrėžtumas susijęs su projekto vykdyme naudojama technologija: nauja ar išvystyta technologija naudojama, kaip gerai projekto vykdytojas žino naudojamą technologiją.

Aplinkos neapibrėžtumas reiškia organizacijai vidinės ir išorinės aplinkos neapibrėžtumo lygį. Jo poveikis gali pasireikšti netikėtais aplinkos politikos pokyčiais, susijusių organizacijų veiksmams: konkurentų, partnerių, tiekėjų, vartotojų, valstybės valdžios ir kt.

Socialinis-žmogaus neapibrėžtumas kyla iš sudėtingų žmonių ryšių projekte. Ryšiai tarp projekte dalyvaujančių asmenų ne visada yra formaliai apibrėžti, taip pat ypač svarbūs yra skirtumai tarp susijusių žmonių. Kultūriniai, socialiniai, vertybių, asmeninės patirties skirtumai gali turėti įtakos atsirandant netikėtumams ir nenumatytiems atvejams projekte.

Kad ir kokia būtų neapibrėžtumo prigimtis, jis vienaip ar kitaip susijęs su žinių trūkumu, kuris gali būti klasifikuojamas į [Cle09]:

- neapibrėžta informacija;
- neapibrėžtas supratimas;
- neapibrėžtas greitis (tempas);
- neapibrėžtas kompleksiskumas.

Informacijos trūkumas apsunkina sprendimų priėmimą, planavimą ir įvykių numatymą, tačiau turimą informaciją reikia teisingai suprasti, padaryti teisingas išvagas ir išvadas. Tą padarius klaidingai, kyla supratimo trūkumas.

Neapibrėžtas greitis kyla iš nežinojimo, kiek truks tam tikros projekto užduotys ar kaip greitai vyks darbas, kaip greitai reaguos suinteresuotos projekto šalys ir pan. Neapibrėžtas kompleksiskumas atsiranda tuomet, kai nėra iki galo žinoma, kokio sudėtingumo projekto modelis iš tiesų yra.

2.2. Neapibrėžtumas projektų planavime

Žvelgiant iš projekto užduočių tinklo perspektyvos, galima išskirti kelių rūšių neapibrėžtumą:

- užduočių trukmių neapibrėžtumą;
- projekto užduočių tinklo struktūros ir jo evoliucijos neapibrėžtumą;
- terminų neapibrėžtumą;
- išteklių kiekio neapibrėžtumą;
- išteklių kvalifikacijos neapibrėžtumą.

Plačiai kalbama apie užduočių neapibrėžtumą, todėl jis lengvai suprantamas. Projekto užduočių tinklo struktūros neapibrėžtumas reiškia skirtingas užduočių tarpusavio priklausomybes ir kaip jų pasikeitimas pakeičia visą projekto užduočių tinklo struktūrą. Taip gali pasikeisti kritinis kelias ir kritinė grandinė, o dėl to – ir buferių apskaičiavimas.

Išteklių kvalifikacijos neapibrėžtumas reiškia, kad projekto eigoje gali paaiškėti, jog trūksta tam tikros kvalifikacijos išteklių atlikti nenumatytoms užduotims arba numatytoms, bet kitų techninių žinių reikalaujančioms, užduotims. Jeigu šie ištekliai nebuvo numatyti anksčiau, jų trūkumas gali įtakoti projekto įvykdymą laiku.

Plačiaja prasme neapibrėžtumas gali būti skirstomas į dvi pagrindines klases [Cle09]:

1. kintamumas;
2. nenuspėjamumas.

Pirmajai klasei priskiriamas toks neapibrėžtumas, kai galima suformuluoti tam tikrą pasekmių rinkinį tam tikrose ribose. Pavyzdžiui, projekto terminas gali pasikeisti, partnerių darbo atlikimo laikas gali keistis, išteklių kiekis gali sumažėti arba padidėti projekto vykdymo metu. Nors yra žinomi tam tikri pasekmių variantai, tačiau sudėtinga pasakyti, kurie iš jų įvyks ateityje.

Antroji klasė – nenuspėjamas neapibrėžtumas – apima tai, ko negalime numatyti. Sudėtinga apsibrėžti net galimus scenarijus, nes galbūt nėra suprantamas procesas, kuris ir atveda į nenumatytas situacijas, arba nėra aiški įvykių seka, kurios eigoje atsitinka nuspėta pasekmė.

2.3. Skirtingi neapibrėžtumo problemos sprendimo požiūriai

Neapibrėžtumo problema projektuose gali būti sprendžiama įvairiomis priemonėmis ir į sprendimą galima žiūrėti iš skirtingų perspektyvų.

Projektų vėlavimo dėl nenumatytų aplinkybių problemos galima ieškoti visų pirma užduočių vertinimo metoduose, kadangi užduočių įgyvendinimo trukmės tiesiogiai siejasi su viso projekto įgyvendinimo trukme. Tai gana siauras požiūris, neatsižvelgiantis į kitus svarbius projekto

planavimo parametrus, tokius kaip projektui skirtus išteklius. Kad ir kaip tiksliai bus prognozuotos projekto užduotys, turimi žmogiškieji ištekliai dėl tam tikrų savo charakteristikų ar aplinkybių gali laiku neatlikti užduoties. Vis dėlto užduočių vertinimas nėra užmiršta tema ir šioje srityje galima atrasti įdomių įžvalgų susijusių su neapibrėžtumu projektuose, kurias aptariame tolimesniame poskyryje.

Kita problemos sprendimo perspektyva susijusi su skirtingais proceso modeliais, kuriuos taikydami vadovai gali netiesiogiai išvengti neapibrėžtumo padarinių projekto eigoje. Modeliai, pasižymintys tokiais savybėmis, apžvelgiami tolimesniame skyriuje.

Trečioji perspektyva neapibrėžtumo problemai spręsti susijusi su projekto planavimo metodais ir technikomis. Šiam požiūriui skiriame daugiausiai dėmesio ir sukurtas sprendimas bei atlikta atvejo analizė remiasi būtent projekto planavimo efektyvumo tobulinimu.

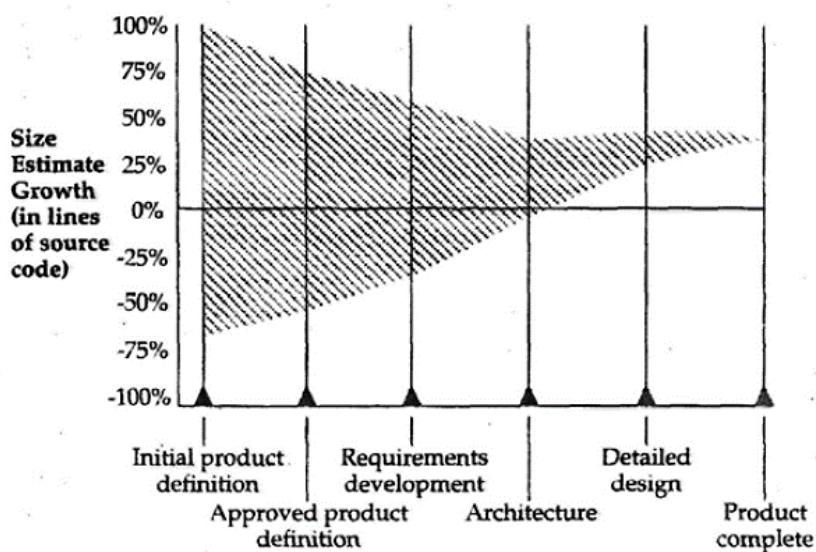
2.3.1. Užduočių vertinimas ir neapibrėžtumas

Projekto pradžioje ir ypač planavimo etape užduočių vertinimas yra viena svarbiausių veiklų, nes jos rezultatai įtakoja, ar projektas bus pradėtas vykdyti ir, jeigu bus pradėtas, apsprendžia tolimesnę jo eigą. Projekto pradžioje vertinama abstrakčiai, tačiau vėliau reikalingi kiek įmanoma tikslesni užduočių vertinimai, kad būtų galima efektyviai skirstyti projekto išteklius bei planuoti galutinį projekto terminą. Projekto vėlavimas, priklausomai nuo sutarties tarp vykdytojo ir užsakovo, gali reikšti netgi baudų išmokėjimą.

Projekto pradžioje turima mažiausiai informacijos, todėl sudėtinga tiksliai prognozuoti užduočių atlikimo trukmės įverčius, dėl ko atsiranda vertinimo neapibrėžtumas. Projekto vykdymui įsibėgėjus projekto užduočių neapibrėžtumas mažėja, todėl atsiranda sąlygos vertinimo neapibrėžtumo mažinimui. Sharfer [Sha14] siūlo metodą neapibrėžtumo mažinimui užduočių vertinime. Šiuo metodu užduočių vertinimas vyksta etapais. Pirmame etape užduotys vertinamos su didele paklaida pasinaudojant eksperto nuomone ir analogijomis iš pereinusių programų sistemos versijų. Antras etapas, vadinamas vertinimo pataisymu, vykdomas po to, kai baigti detalūs sistemos reikalavimai, ir naudojama disagregacijos technika (aprašyta aukščiau). Pirmame etape identifikuojami sistemos savybių reikalavimai, kurie bus pateikiami klientui. Kiekvienam šių reikalavimų suskaičiuojamos darbo valandos, įtraukiant sistemos inžinerijos, programavimo, integracijos, „rankinio“ ir automatinio testavimo veiklas. Antrame etape identifikuojami priklausomybės ryšiai tarp pirmame etape nustatytų reikalavimų, identifikuojami sub-reikalavimai, reikalingas išorinis (kitų produktų) indėlis. Suskaičiuojamos darbo valandos ir nustatomos datos, kada šie reikalavimai bus pateikiami klientui. Reikalavimo vykdymo metu nuolat stebimas defektų skaičius, jų skaičiui augant didėja užduoties atlikimo trukmė. Reikalavimų

realizacijos trukmių pakartotinis vertinimas vyksta tik tuo atveju, kai komanda sutaria, jog reikalavimas buvo klaidingai įvertintas. Tuomet reikalavimo realizavimo trukmė yra pakeičiama, kartu nežymiai pakeičiant kitų reikalavimų įvertinimą.

Pagal [ASB10] ir [Con98] reikia didesnę dėmesį sutelkti į projekto vykdymo eigos stebėjimą ir atitinkamas projekto užduočių įverčių korekcijas, nei į pradinį užduočių įvertinimą. Šis pradinis vertinimas yra smarkiai įtakojamas neapibrėžtumo, o tikslūs užduočių trukmių įverčiai žinomi tik pateikus produktą klientui [ASB10]. Toks reiškinys vadinamas „neapibrėžtumo kūgiu“ (5 pav.). „Kūgis“ suprantamas kaip mažėjantis vertinimo neapibrėžtumas artėjant prie baigiamojo projekto etapo – nuo abstrakčių sprendimų pereinama prie tikslių sprendimų.



5 pav. „Neapibrėžtumo kūgis“.

Šaltinis: [Con98]

Autoriai [ASB10] sukūrė rutininę dalinai automatizuotą vertinimo platformą, kuri turėtų padėti sumažinti programų sistemų kūrimo projektų vertinimų neapibrėžtumą projekto vykdymo eigoje. Sprendimas apjungia unifikuotą kodo skaičiavimo įrankį (UCC) su COCOMO II vertinimo modeliu. Naudojant platformą galima įvertinti projekto ir komandos efektyvumą ir prognozuoti reikalingus išteklius likusiam darbui atlikti. Ši informacija turėtų būti naudojama koreguoti projekto įverčių parametrus – tai galiausiai padės įvertintas ir faktines laiko sąnaudas konverguoti.

2.3.2. Neapibrėžtumo valdymas skirtingais proceso valdymo modeliais

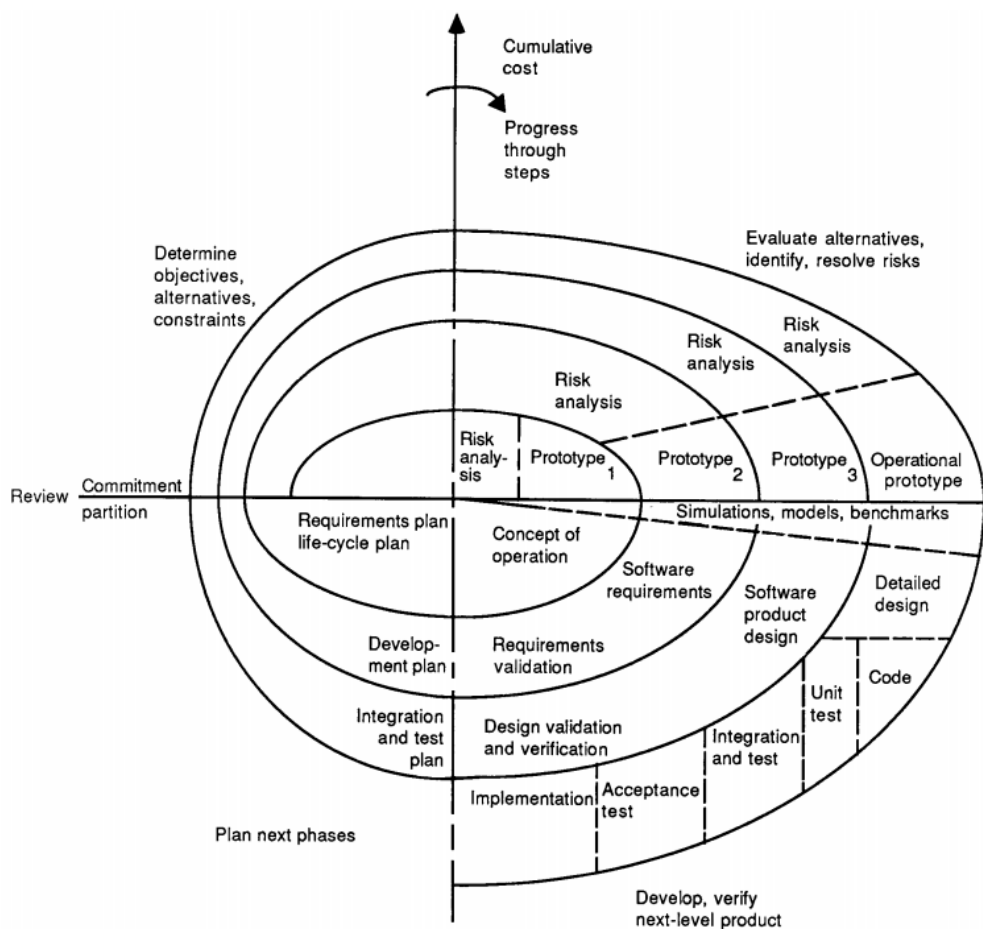
Projektų valdymo modeliai parodo, kokia tvarka projektas bus vykdomas [Boe88]. Tai reiškia, kad skirtingi modeliai nubrėžia skirtingą užduočių etapų seką, kas savo ruožtu įtakoja skirtingą projekto užduočių tinklą tam tikram projektui.

Boehm spiralinis modelis sukurtas siekiant pagerinti programų sistemų kūrimo ir vystymo procesą ir gali iš dalies būti naudojamas kaip neapibrėžtumo valdymo priemonė. Modelio kūrėjas Boehm teigė, kad modelis yra rizikų varomas metodas, o ne dokumentavimo ar programinio kodo varomas proceso modelis [Boe88].

Spiralinis Boehm proceso modelis skirtingų projekto etapų tėkmę vaizduoja spirale (6 pav.). Spiralinės spindulys vaizduoja akumuliuotus projekto vykdymo kaštus, pati spiralė – įvykdymo mastą. Kiekvienas spiralinės ciklas turi tuos pačius žingsnius [Boe88]:

- nustatyti produkto (programų sistemos) dalies tikslus;
- produkto dalies įgyvendinimo alternatyvos;
- alternatyvų ribojimai (kaina, tvarkaraštis ir pan.);
- įvertinti alternatyvas pagal apribojimus.

Pastarasis žingsnis įprastai atskleidžia neapibrėžtumo sritis, kurios yra svarbus projekto rizikų šaltinis. Jeigu neapibrėžtumo sritys identifikuotos, vykdomi tolimesni žingsniai: prototipų kūrimas, simuliacijos, metrikų lyginimas su etalonais ir kt.



6 pav. Boehm spiralinis programų sistemų proceso modelis

Šaltinis: [Boe88]

Boehm aiškiai neatskiria rizikos ir neapibrėžtumo sąvokų, tik užsimena, jog neapibrėžtumas yra rizikų šaltinis. Nustatytos rizikos gali įtakoti darbo produktų kūrimo eigą. Pavyzdžiui, tam tikrame etape techninė specifikacija gali būti rašoma ne iki galo arba visai nerašoma, kol rizikos nėra „išspręstos“.

Kitas proceso modelis, kuris gali būti naudojamas neapibrėžtumui valdyti, yra lygiagretus projekto vykdymas, kitaip vadinamą projektų alternatyvų valdymą. Projektų alternatyvų valdymo (angl. *parallel-trials approach*) taikymas reiškia vienu metu vykstančias projektų užduotis, kai nėra iš anksto žinoma, koku keliu eiti, siekiant gauti norimą rezultatą [LL09]. Toks projektų valdymas buvo taikomas nuo 1940-ųjų metų įvairiuose kariniuose projektuose („Manheteno“, „Atlas“, „Polaris“ ir kt.), kurie pasižymėjo dideliu techniniu neapibrėžtumu [LL09]. Minėtais projektais buvo siekiama gauti sudėtingus ir novatoriškus techninius sprendimus: sukurti atominę bombą, tarpžemyninę raketą ar kt.

Projektų alternatyvų valdymo būdas skirtas išvengti nesėkmingo produkto ir dažnai pastatomas kaip priešprieša „teisingai gauti iš pirmo karto“ būdui, todėl gali būti tinkamas inovatyviems, R&D tipo projektams [LL09]. Alternatyvų būdas dažniausiai kritikuojamas dėl didelių išlaidų (reikia daugiau išteklių, kad vienu metu dirbtų kelios komandos, kurios ne visos tiesiogiai prisidės prie sėkmingo produkto sukūrimo), chaotiškumo ar neprofesionalumo. Tačiau autorius [LL09] atkreipia dėmesį, kad darbo grupių, kurių bandymai neatneša sėkmės, surinkta informacija yra vertinga ir gali būti panaudojama kitiems projektams. Be to, taikant projekto alternatyvų valdymo metodą, sumažėja nesėkmingo projekto rizika.

Judrieji programų sistemų kūrimo metodai yra vieni populiariausių industrijoje [KMI15] ir jų siūlomas lankstumas ir kitoks požiūris į projektų valdymą gali būti viena iš neapibrėžtumo mažinimo priemonių. Šie metodai projekto planavimo etape akcentuoja tris projekto planavimo veiklas: projekto užduočių prioritetizavimą, projekto apimties valdymą ir išteklių valdymą [KMI15].

Lean taikomi principai „daryti sprendimus kaip įmanoma vėliau“ ir „tiekti kaip įmanoma greičiau“ iš principo turėtų padėti mažinti neapibrėžtumo keliamus padarinius. Kai nėra pakankamai informacijos apie tam tikras projekto užduotis, reikėtų atidėti sprendimus kaip įmanoma vėlesniam laikui. Taip sumažinama neteisingų sprendimų rizika.

Darbo skaidymas į iteracijas yra ryškiausia neapibrėžtumo valdymo priemonė. Jeigu projekto apimtį galima mažinti priklausomai nuo to, kaip efektyviai pavyksta įgyvendinti užsibrėžtus tikslus, tai neapibrėžtumas apsiriboja iteracija. Kiekvienoje iteracijoje projektų vadovas ar šiuo atveju produkto šeimininkas turi įsitikinti, kad išteklių ir laiko pakaks numatytiems darbams įvykdyti, net susidūrus su nenumatytais įvykiais.

Vis dėlto programų sistemų kūrimo projektai, paremti judriaisiais metodais, kaip ir kiti projektai, susiduria su neapibrėžtumo valdymo problemomis projekto stebėjimo ir kontrolės kontekste. Pagal [ASB10] projektų progreso stebėjimas yra nepakankamas, mažinant neapibrėžtumą projekto eigoje. Siekiant suprasti bendrą projekto kryptį ir efektyviai planuoti išteklius tolimesniems darbams, būtina periodiškai tikrinti projekto progresą ir mažinti „neapibrėžtumo kūgį“ (5 pav.). Nesiimant priemonių mažinti šį atotrūkį ir nuolat tikslinti projekto įverčius projekto eigoje, rizikuojama prarasti projekto kontrolę ir nespėti įgyvendinti projekto laiku.

2.3.3. Projektų planavimo metodai esant neapibrėžtumui

Autoriai W. Herroelen ir R. Leus [HL02] išskiria šešis projektų planavimo metodus, skirtus projektams pasižyminčiais neapibrėžtumu ir daugiausia tyrinėjančius problemą mašinių planavimo srityje:

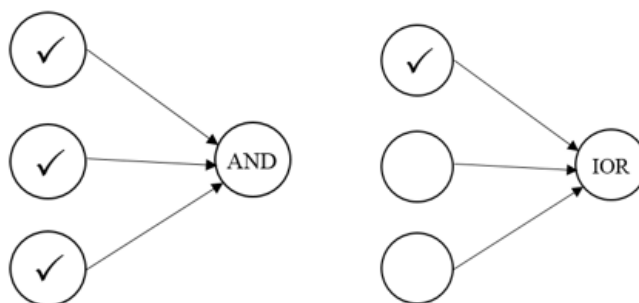
- reaktyvus planavimas;
- stochastinis projektų planavimas;
- stochastinis GERT užduočių tinklo planavimas;
- netikslusis projektų planavimas;
- proaktyvus projektų planavimas;
- atsparumo analizė.

Reaktyvus planavimas (angl. *reactive scheduling*) metodas nesiekia sukurti gero projekto plano atsižvelgiant į neapibrėžtumą, bet pakartotinai peržvelgia planą, kai įvyksta netikėtas įvykis. Visos pastangos sutelkiamos plano reoptimizacijai, kuri gali būti atlikta įvairiomis strategijomis. Tokios strategijos gali būti „dešiniojo poslinkio taisyklė“, kai projekto užduotys tiesiog paslenkamos į priekį arba projekto tvarkaraščio perdarymas taip, kad naujas planas kiek įmanoma mažiau skirtųsi nuo pirminio plano. Sukurta ir sudėtingesnių metodų, tokių kaip autorių Artigues ir Roubellat [AR00] išvystytas polinominis įterpimo algoritmas, kurio esmė yra sugeneruoti dominuojančius įterpimo pjūvius projektų užduočių tinkle, o kiekvienam pjūviui išvedama geriausia galima įterpimų aibė.

Stochastinis (angl. *stochastic project scheduling*), dar kitaip vadinamas *ištekliais ribojamas* (angl. *resource constrained*), projektų planavimas sprendžia, kaip reikėtų sumažinti laukiamą projekto trukmę, kai projekto užduočių trukmės yra neapibrėžtos, remiamasi pabaigos-pradžios užduočių ribojimais, o ištekliai yra atsinaujinantys. Čia projektas vaizduojamas projektų užduočių tinklu, kur pats tinklas gali būti užrašomas kaip $G = (V, E)$, kur aibė $V = \{1, 2, \dots, n\}$ žymi projekto užduotis. Pirmoji ir paskutinė užduotys (1 ir n) yra fiktyvios užduotys, apibrėžti projekto pradžiai

ir pabaigai. Kitų užduočių trukmės apibrėžiamos atsitiktiniu vektoriumi $d = (d_2, d_3, \dots, d_{n-1})$. Aibė E atitinka pabaigos-pradžios ryšius tarp užduočių. Atsinaujinantys ištekliai ($k = 1, 2, \dots, K^p$) yra galimi sveikais skaičiais išreikštais kiekiais a^p_k . Kiekviena užduotis i reikalauja tam tikro a kiekio k tipo išteklių. Tuo būdu yra sugeneruojami projekto užduočių tvarkaraščiai, remiantis tam tikromis planavimo strategijomis.

GERT tinklo planavimas (angl. *stochastic GERT network scheduling*) skirtingai nei *stochastinis planavimas* įtraukia viso projekto užduočių tinklo struktūros vystymosi neapibrėžtumą. Tai reiškia, kad nėra iš anksto žinoma, kokie priklausomybės ryšiai jungs atskiras užduotis. GERT projekto užduočių tinklas turi vieną pradžios ir vieną pabaigos mazgą. Jis leidžia turėti užduočių ciklus ir dėl to tos pačios užduotys gali būti atliekamos kelis kartus. GERT tinklas turi šešių tipų mazgus pagal įėjimo ir išėjimo kraštinių (stygų) tipų kombinaciją. Pavyzdžiui, mazgas turi IR tipo įėjimo kraštinę, jei projekto vykdymo metu mazgas yra aktyvuojamas – įvyksta susietas projekto įvykis – ir visos ankstesnės užduotys (iš kurių išėjo ši užduotis) yra įvykdytos (7 pav.). Mazgas yra ARBA tipo, jei užduotis aktyvuojama, kai bent viena prieš tai sekusi užduotis yra atlikta pirmą kartą.



7 pav. GERT projekto užduočių tinklo mazgų tipai IR ir ARBA

GERT planavimas remiasi prielaida, kad skirtingų užduočių trukmės ir tos pačios užduoties, tik skirtingai atliktos, trukmės yra nepriklausomos. Reikalingi ištekliai yra mašinų tipo. Kadangi GERT tinklų analizė reikalauja didelių skaičiavimo išteklių, jam pagrįsti naudojamos simuliacijos.

Netikslusis (angl. *fuzzy project scheduling*) projektų planavimas remiasi prielaida, kad projekto užduočių trukmių tikimybinis pasiskirstymas nėra žinomas ir vertinant, kiek užtruks atlikti tam tikrą užduotį, yra žinomi tik apytikriai trukmių režiai. Pavyzdžiui, manoma, kad užduotis truks tikrai ilgiau nei dvi dienas, tačiau neturėtų trukti ilgiau nei penkias dienas, todėl ši planavimo technika labiau tinka projektams, susiduriantiems su netikslumu, nei kad neapibrėžtumu. Ši technika naudoja taip vadinamas priklausomybės funkcijas (angl. *membership functions*), reprezentuojančias „tiesos laipsnį“. Naudojamas terminas netiksloji aibė (angl. *fuzzy*

set) matuoja priklausomybės laipsnį aibei. Aibė A aibėje X gali būti apibrėžta priklausomybės funkcija $\mu_A: X \rightarrow \{0,1\}$, kur $\mu_A(x) = 1$, kai $x \in A$ ir $\mu_A(x) = 0$, kai $x \notin A$. Kuo arčiau vieneto yra funkcijos reikšmė, tuo didesnė tikimybė, kad x priklauso aibei A . Jeigu nėra žinoma, ar x priklauso aibei A ar ne, tuomet tariama, jog x patenka į intervalą $[0,1]$.

Netiksliojo projektų planavimo rezultatas bus „miglotas“ projekto tvarkaraštis su netiksliais užduočių pradžios ir pabaigos taškais. Tokia planavimo technika labiau tinkama kaip sprendimo priėmimo pagalbinė priemonė, nei priemonė apibrėžti tikslią projekto užduočių pradžią ir pabaigą [HL02].

Proaktyvus projektų planavimas (angl. *robust (proactive) project scheduling*) leidžia surasti sprendimus, kurie turėtų vienodai gerą rezultatą, nesvarbu, kokius duomenis įvesime [LAN11]. Tokio planavimo tikslas yra sukonstruoti proaktyvų projekto tvarkaraštį, kuris galėtų susidoroti su projekto vykdymo eigoje išskylančiais trikdžiais [VDH08]. Dažnai *proaktyvus projektų planavimas* yra pastatomas kaip priešprieša *reaktyviam projektų planavimui* (aptartas aukščiau), tačiau pagal [VDH04] proaktyvus projektų planavimas efektyvumu lenkia pastarąjį, kai projekto užduočių trukmių nuokrypis nuo numatyto nėra didelis, ir kai projekto vykdymo laiko intervale yra palikta vietos buferiams. Kai šios sąlygos yra priešingos, gerų rezultatų pasiekia hibridinės technikos taikymas – reaktyvaus ir proaktyvaus planavimo kombinacija.

Proaktyvus projektų planavimas susilaukė didelio populiarumo, šiai grupei priskiriamas ir *kritinės grandinės* technika. Šios technikos esminis parametras – buferiai, kurių pagalba stengiamasi išlaikyti bendrą projekto planą tam tikruose režiuose, net jei įvyksta nenumatyti įvykiai. Technika išsiskyrė tuo, kad didelį dėmesį skyrė žmogiškųjų išteklių ir jų tarpusavio priklausomybės problemoms spręsti projektų planavimo etape. Jos autorius Goldratt, naudodamas buferio sąvoką projekto planavime, rėmėsi tam tikromis prielaidomis [Ste01]: a) didelės neigiamos pasekmės individui nespėjusiam atlikti užduoties lyginant su nedideliu atpildu įvykdžius užduotį greičiau nei numatyta; b) užduoties termino sudarymas derybų principu, kai pasiūlomas dydis yra didesnis vien tam, kad būtų apsaugota nuo jo mažinimo derybų metu; c) visų organizacijos lygių vadovai prideda savo rezervus, siekdami apsaugoti nuo netikslų įverčių.

Kitas projektų planavimo metodas – *atsparumo analizė* (angl. *sensitivity analysis*) – nagrinėja klausimus, susijusius su įvairių parametrų keitimu, siekiant išlaikyti optimalų sprendimą. Ši technika naudojama mašinų užduočių planavime, o jos autoriai ieško atsakymų į tokius klausimus, kaip „Kokie yra parametro pokyčių ribos, kad sprendimas liktų optimalus?“, „Kada pirminis užduočių planas išlieka optimalus?“, „Kokia bus nauja optimali kaina, turint parametro pasikeitimą“ ir pan. [HL02].

Autoriai A. Burns ir kiti [BPSW99] tyrinėdami *realaus laiko* sistemų (angl. *real-time systems*) užduočių planavimą, kombinavo *planavimo analizę* (angl. *schedulability analysis*) kartu

su *atsparumo analize*, siekiant nustatyti maksimalų klaidų dažnį, kurį sistema dar gali toleruoti. Tai jie vadina tikimybinės garantijos paieška. Terminas „garantija“ šiuo atveju naudojamas kaip užduočių plano garantija, darant prielaidą, jog visos sistemoje vykdomos užduotys būtų pabaigtos iki nustatyto termino. *Atsparumo analizė* buvo naudojama nustatant galimybės planuoti ribas – kada dar galima planuoti arba minimalus toleruojamas intervalas tarp įvykusių atsitiktinių klaidų.

Aptarti šeši metodai turi savų ypatumų ir tinka ne visiems atvejams. *Stochastinis projektų planavimas* neleidžia sukurti pradinio plano, kuriuo būtų galima remtis derybose su projekto investuotojais. Ši technika projekto planavimą apibrėžia kaip kelių pakopų sprendimo procesą, kuris pasako, kokios užduotys turi būti pradėtos tam tikrame laiko momente [HL02]. GERT planavimas reikalauja didelių skaičiavimo išteklių, taip pat yra laikoma, jog visi užduotis atliekantys ištekliai yra mašinos. Netikslusis planavimas remiasi priklausomybės funkcijomis, kad išvengtų tikimybinio pasiskirstymo naudojimo, tačiau priklausomybės funkcijas gali būti taip pat sudėtinga sugeneruoti kaip ir tikimybinio pasiskirstymo funkcijas [HL02]. *Proaktyvus planavimas* labai suklestėjo mašinų planavimo srityje, ypač vienas jo instrumentų – buferių įterpimas – Goldratt *kritinės grandinės* (angl. *critical chain*) technikos komponentas.

2.3.4. Kritinės grandinės technika ir jos specifika

Kritinės grandinės technika, kilusi iš kritinio kelio metodo, susilaukė didelio populiarumo projektų planavimo srityje.

Ši technika remiasi grafų ir sistemų teorijomis ir skirtingai nei kritinio kelio metodas taip pat remiasi apribojimų teorijos (angl. *Theory of Constraints*) principais [LRS05]:

- sistemos veikimą ribojančių veiksnių identifikavimu;
- turimų išteklių panaudojimu;
- mažiau svarbių išteklių pertekliaus panaudojimu „butelio kaklelio“ išteklių darbo efektyvumui gerinti;
- visos sistemos galingumo didinimu, nepavykus pagerinti jos veikimo naudojant prieš tai minėtus principus.

Nors kritinio kelio metodas kalba apie turimų išteklių svarbą projektų planavime ir kaip prieinamumas keičia projekto planą [Faz59], tačiau skaičiuojant planuojamą projekto trukmę, o kartu ir kritinį kelią, išteklių priklausomybės nėra įtraukiamos. Tuo tarpu kritinės grandinės technika išteklius įtraukia kaip vieną iš pagrindinių projekto užduočių tinklo sudarymo elementų. Taip pat skirtingai nei kritinio kelio metodas, kritinės grandinės technika išskiria išteklius į kritinius ir nekritinius ir didelį dėmesį skiria kritinių išteklių valdymui ir planavimui atsižvelgiant visų pirma į šiuos kritinius išteklius [LRS05].

Kritinės grandinės technika naudoja jau minėtus *patenkinčiosius* modelius, sudarant bazinį projekto planą ir valdant projektus jų vykdymo etape. Pasak Goldratt, toks požiūris yra geriausias šių laikų kompleksiško ir neapibrėžtumo sąlygomis, kas būdinga realiose aplinkose [Gol97].

Svarbus kritinės grandinės technikos elementas yra „paslėptos atsargos“ darbo užduočių laiko įverčiuose eliminavimas ir jų panaudojimas bendrai projekto atsargai, vadinamam globaliam buferiui. Goldratt teigia, jog darbuotojai vertindami užduočių laikus jas įvertina dvigubai tiek, kiek iš tiesų reiktų skirti laiko ją atlikti, kai viskas klojasi sėkmingai [Gol97].

Kita svarbi kritinės grandinės technikos iškelta problema yra išteklių „blaškymasis“ tarp kelių užduočių ar projektų vienu metu (angl. *multi-tasking*). Goldratt pabrėžia, kad šito reikia stengtis išvengti, nes tai mažina darbo efektyvumą ir yra dažna problema organizacijose, kur vykdomi keli projektai vienu metu, o ištekliais yra dalijamasi.

Lyginant kritinio kelio metodą su kritinės grandinės technika, terminų svarbos interpretavimas taip pat skiriasi. Kritinio kelio technika didžiausią dėmesį skiria atskirų projektų užduočių terminams, o kritinės grandinės technika pabrėžia bendro projekto pabaigos termino svarbą [LRS05]. Pastarasis požiūris palieka daugiau lankstumo projekto elementams keisti, kad būtų pasiektas bendras tikslas – visas projektas būtų įvykdytas laiku.

Nors kritinė grandinė sprendžia neapibrėžtumo problemą, būdingą projektams, tačiau metodas nedetalizuoja, kokio tipo neapibrėžtumą mažina. Pagrindinė priemonė šiai projektų problemai spręsti yra buferių taikymas. Šie buferiai turi padengti užduočių (ypač kritinių) vėlavimą ir taip apsaugoti visą projektą nuo vėlavimo. Tokiu atveju tai sprendžia užduočių trukmių, užduočių tinklo struktūros ir užduočių terminų neapibrėžtumo problemą (aptarti antrame skyriuje).

Kritinės grandinės technika ištekliais laiko atsinaujinančius žmogiškuosius išteklius ir nekalba apie neatsinaujinančius. Pasak [HR01] šis metodas nepakankamą dėmesį skiria užduotims, kurios reikalauja kelių tokio paties tipo atsinaujinančių išteklių, o ne tik vieno.

Kritinė grandinė nesiima spręsti išteklių neapibrėžtumo poveikio projekto eigai, nors ištekliai yra vienas svarbiausių šios technikos ypatumų. Išteklių įtraukimas į projekto užduočių tinklą yra svarbus indėlis projektų planavime ir jų svarba yra neabejotina. Tačiau kad ir kaip sėkmingai suplanuosime projekto užduočių tinklą ir paliksime vietos buferiams, neužtikrinsime išteklių neapibrėžtumo problemos sprendimo. Tai susiję su skirtingomis išteklių kvalifikacijomis ir skirtingų kvalifikacijų poreikiu skirtingoms užduotims. Jeigu nebus prieinami suplanuoti tam tikros kvalifikacijos ištekliai, numatyti buferiai gali nepadengti užduočių vėlavimo, kol bus laukiama reikiamų išteklių.

3. Darbo metodika

Šiame skyriuje aprašoma, kaip buvo patikslintas kritinės grandinės metodas programų sistemų projektų tvarkaraščių sudarymui, kokių duomenų ir kokios analizės reikėjo siekiant darbo tikslo.

Darbo eigą galima išskirti į keletą etapų:

1. Dalykinės srities analizė
2. Patikslinto kritinės grandinės metodo kūrimas
3. Specifikuoto metodo taikymas
4. Išvadų ir rezultatų apibendrinimas

Toliau detaliai aprašoma kiekvieno etapo eiga ir kaip buvo prieita prie darbe iškelto tikslo.

1. Dalykinės srities analizės etapas. Šioje dalyje buvo atliekama mokslinių ir ne mokslinių darbų analizė, kurios metu rinkti duomenys apie egzistuojančius projektų planavimo metodus neapibrėžtumo kontekste. Buvo išanalizuoti sprendimai iš skirtingų perspektyvų. Šio etapo metu buvo nuspręsta orientuotis į projektų planavimo metodus ir priemones. Iš tokių planavimo metodų buvo atrinkta kritinės grandinės technika, kurios principais ir remiamasi šiame darbe. Kritinės grandinės technika yra kilusi iš kritinio kelio metodo ir kaip jau minėta antrame skyriuje yra laikoma geresnė, modernesnė ir labiau tinkama realiems projektams planuoti nei jos „pirmtakas“.

Galime išskirti šiuos kritinės grandinės principus, kuriuos imame kaip bazę siekiant darbe iškelto tikslo:

- „Atgalinis planavimas“. Jis leidžia atsižvelgti į „neapibrėžtumo kūgį“ ir vadovautis tuo, kad kuo vėliau pradedama užduotis, tuo daugiau informacijos turima – neapibrėžtumas mažėja.
- Pagrindiniais projekto užduočių tinklo sudarymo principais:
 - užduočių tarpusavio ryšių identifikavimas;
 - užduočių „paslėpto“ buferio eliminavimas;
 - išteklių priskyrimas užduotims taip, kad kiekvienas jų turėtų tik 1 užduotį vienu metu;
 - kritinės grandinės nustatymas.
- Goldratt numatytais projekto ir „maitinimo“ buferių tipais ir jų apskaičiavimu.

2. Patikslinto kritinės grandinės metodo kūrimas. Šiame etape kuriamas projektų planavimo metodas. Metodas kurtas tokia eiga:

1. Naudojamų sąvokų išgryninimas ir interpretavimas
2. Kuriamo metodo taikymo prielaidų ir apribojimų nustatymas
3. Metodo specifikos ir konkrečių taikymo žingsnių aprašymas

Nagrinėjant kritinės grandinės taikymo ypatumus ir atsižvelgiant į šiuolaikinių programų sistemų projektų specifiką, buvo išgryninti aspektai, į kuriuos kritinės grandinės metode nėra atsižvelgiama ir kurie nėra detalai nagrinėjami projektų planavimo kontekste. Vienas pagrindinių tokių aspektų yra išteklių apribojimas ir galimybės valdyti išteklių komplektavimo neapibrėžtumą. Šių ypatumų pastebėjimas leido papildyti įprastinę kritinės grandinės techniką nauju požiūriu ir spręsti išteklių neapibrėžtumo problemą.

4. Specifikuoto metodo taikymas. Šio etapo metu patikslintas metodas buvo pritaikytas realiam pasibaigusiam projektui. Metodui patikrinti buvo atlikta **atvejo analizė**. Išnagrinėti realaus programų sistemų projekto planavimo ir vykdymo eigos duomenys, kurių analizė leido išskirti ir susisteminti projekto užduočių tinklo sudedamąsias dalis, reikalingas pritaikyti metodą.

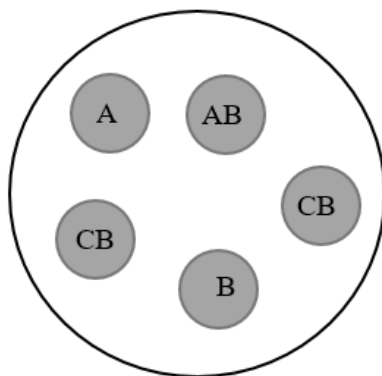
Taip pat buvo apskaičiuota, kaip papildyto metodo taikymas leido prognozuoti projekto pradžios datą ir bendrą vykdymo laiką. Šie dydžiai buvo palyginti su faktiniais projekto planavimo ir įgyvendinimo duomenimis. Tai leido nustatyti metodo efektyvumą pagal jo kūrimo etape iškeltus kriterijus.

5. Išvadų ir rezultatų apibendrinimas. Darbo gale aptariami metodo rezultatai ir įvertinama, ką pavyko pasiekti nauju kritinės grandinės metodo taikymu. Taip pat pateikti pasiūlymai, kaip darbe atskleidžiamus projekto planavimo veiksnius galima panaudoti tolesniems moksliniams tyrimams atlikti ir kas galėtų būti nagrinėjama akademikų darbuose ateityje.

4. Kritinės grandinės metodo tikslinimas

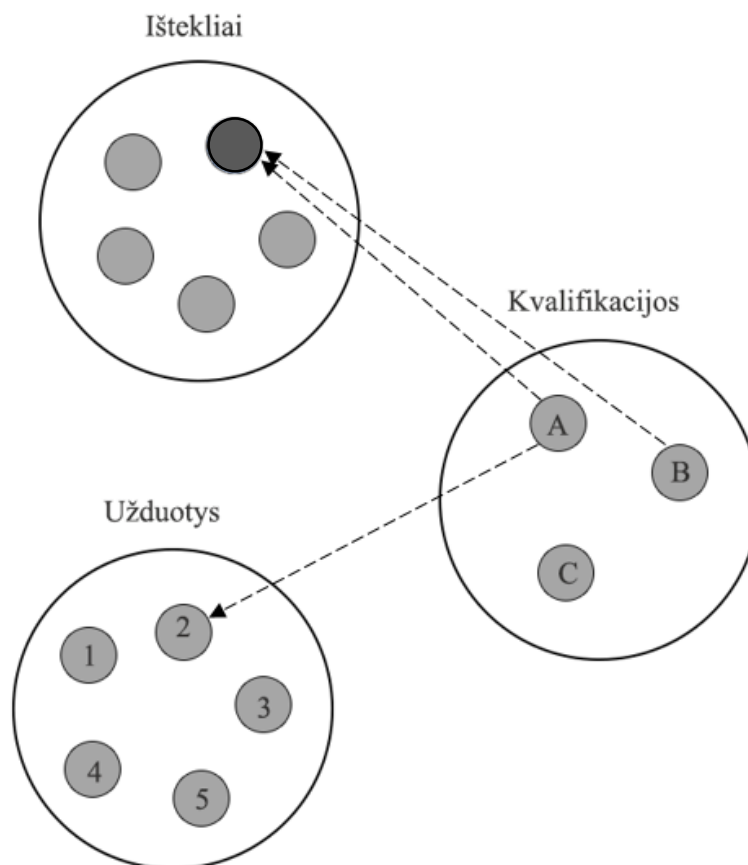
4.1. Naudojamų sąvokų interpretavimas ir prielaidos

Tyrimė minimas „išteklių“ terminas naudojamas tiek žmogiškųjų, tiek dirbtinių išteklių sąvokai perteikti. Darbe laikoma, kad žmogiškuosius išteklius galima panaudoti vienai užduočiai vienu metu vykdyti. Dirbtinius išteklius galima panaudoti dviem būdais, priklausomai nuo jų tipo, todėl tie patys dirbtiniai ištekliai gali būti panaudoti ir vienu metu kelioms užduotims vykdyti, ir tik vienai užduočiai vykdyti. Ištekliais laikomi tam tikrą funkcijų rinkinį atliekantys subjektai ar objektai. Aštuntame paveiksle galima pavaizduoti išteklių rinkinį (angl. *resource pool*) su penkiais ištekliais ir jų gebėjimu atlikti funkcijas A, B, C. Tai reiškia, jog tas pats žmogiškasis išteklius gali turėti kompetencijos atlikti keletą skirtingo tipo užduočių. Pavyzdžiui, programuotojas gali testuoti, o testuotojas – programuoti arba programuotojas gali gebėti atlikti serverio administravimo darbus ir t.t.



8 pav. Išteklių rinkinys

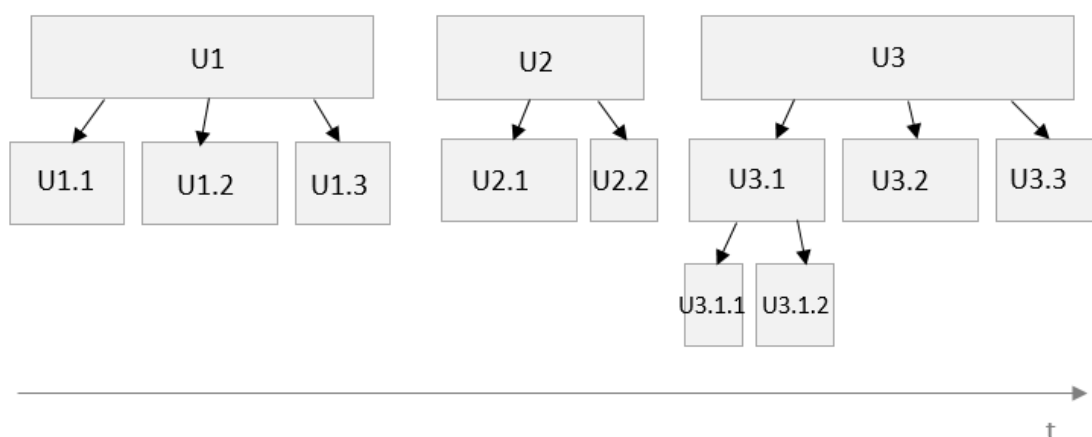
Toks išteklių požiūris yra svarbus, siekiant numatyti visus galimus variantus užduočiai priskirti išteklius. Galimi išteklių, kvalifikacijų ir užduočių rinkiniai pavaizduoti devintame paveiksle. Čia parodyta, kad antrai užduočiai atlikti reikalinga kvalifikacija A, o ją kartu su kvalifikacija B turi pažymėtas išteklius.



9 pav. Išteklių, kvalifikacijos ir užduočių sąryšis

Toliau plačiau paaiškiname kitą tyrime naudojamą sąvoką – **užduotis**. Užduotimi laikomas projekto darbo vienetas, kuris yra projekto apimties dalis. Užduočiai atlikti reikalinga tam tikra viena kvalifikacija, pavyzdžiui, programuotojo kvalifikacija. Jeigu sistemos reikalavimui įgyvendinti reikia atlikti analizės, programavimo ir testavimo darbus, tai darbai bus skirstomi į mažiausiai tris užduotis. Užduotis galima grupuoti ir skaldyti, priklausomai nuo poreikio. Šiame darbe užduotis gali būti stambi ir ją gali sudaryti keletas užduočių, tačiau bendra grupė vadinama užduotimi ir kitaip neišskiriama (10 pav.). Tyrime užduotys skirstomos į tipus pagal tai, kiek išteklių užduotį gali vykdyti vienu metu. Yra atskiriamos užduotys, kurias vienu metu gali vykdyti tik vienas išteklius dėl užduoties specifikos; yra tos užduotys, kurias gali vykdyti du, trys ar daugiau išteklių vienu metu. Pavyzdžiui, sistemos versijos (angl. *patch*) užkėlimą ant testavimo aplinkos atlieka vienas žmogus, nes keliems tiesiog nebūtų prasmės tą daryti, o vieno stambesnio reikalavimo realizavimą gali vykdyti keli programuotojai vienu metu.

Kita užduočių savybė, į kurią darbe atsižvelgiama yra tai, kad vienos užduotys gali būti atliekamos vienu metu su kitomis projekto užduotimis, o kitos privalo eiti nuosekliai. Pavyzdžiui, funkcijos x programavimas negali būti atliekamas vienu metu su funkcijos x testavimu – šias užduotis reikia vykdyti nuosekliai.



10 pav. Projekto užduotys ir jų grupavimas

Dar kita užduočių savybė – galimybė jas skaldyti. Tai reiškia, kad kartais užduotis galima dalinti į kelias dalis ir atlikti ne visą iš karto. Ši užduočių savybė gali būti naudinga planuojant projektą, kai siekiama efektyviau paskirstyti užduotis laike ir išteklius užduotims atlikti. Gali būti, kad naudingiau užduotį atlikti etapais, nei nuosekliai.

Tyrime daromos tam tikros prielaidos dėl darbo užduočių laiko perskaičiavimo, kai prie užduoties prisijungia daugiau nei vienas išteklius. Naujai užduoties trukmei apskaičiuoti naudojama paprasta algebrinė lygtis, pagal kurią galima apskaičiuoti, per kiek laiko tą patį darbą atliks keli ištekliai, nei vienas. Jeigu paskaičiuota, kad tam tikrą užduotį galima atlikti per T laiko dirbant vienam, tai keli ištekliai šią užduotį atliks per:

$$t = \frac{T}{r},$$

kur T – laikas, skirtas užduočiai atlikti, kai dirba vienas išteklius, r – išteklių skaičius.

Ši formulė nebūtinai bus tinkama kitokio pobūdžio nei informacinių sistemų vystymo projektams, tačiau metodas apsiriboja būtent tokio tipo projektų planavimu. Taip pat galima pastebėti, jog ši formulė bus tinkama tokiu atveju, kai visi ištekliai, dirbantys prie tos pačios užduoties, yra vienodo gebėjimų lygio ir netrukdo vienas kitam atlikti užduoties. Pavyzdžiui, vyresnysis programuotojas nemoko jaunesnio programuotojo įgyvendinti sistemos reikalavimo, o abu yra to paties lygio programuotojai, tiek pat žinantys apie kuriamą sistemą.

Darbe naudojama **buferio** sąvoka reiškia papildomą laiko trukmę, pridedamą prie užduoties ar užduočių sekos, kad apsisaugoti nuo netikėtumų. Jų tipai aprašyti ankstesnėse darbo dalyse ir šiame darbe nėra detalios aprašomi.

4.2. Tvarkaraščio sudarymo metodas

Šiame skyriuje pateikiamas projekto tvarkaraščio sudarymo metodas. Išskiriame šiuos projekto užduočių tinklo sudarymo žingsnius, kurių visuma atskleidžia patikslintą kritinės grandinės metodą:

1. Pasiruošimo etapas. Surenkami svarbiausi projekto duomenys:
 - a. Projekto užduotys. Nesvarbu, kiek detalai jos yra išskaidytos, kadangi vėliau esant reikalui bus galima jas detalizuoti. Pradžioje labai detalių užduočių projektų vadovas gali ir nežinoti, o jeigu ir žino, tokiu atveju gali būti sudėtingiau taikyti šį metodą.
 - b. Identifikuojami projekto užduočių tarpusavio ryšiai ir nustatoma, kada užduotis turi būti pradėta, kada privalo būti baigta, kokias kitas užduotis įtakoja ir t.t.
 - c. Užduočių grupavimas. Turimos užduotys yra suskirstomos į grupes pagal jų pobūdį:
 - i. Užduotys, kurias gali vykdyti daugiau nei vienas išteklius vienu metu.
 - ii. Užduotys, kurias geba atlikti daugiau nei vienas turimas išteklius.
 - iii. Užduotys, kurios gali būti atliekamos ne iš karto, t.y. jos gali būti pertraukiamos ir atliekamos dalimis.
 - d. Užduotims įvykdyti skirtas laiko įvertis, gautas iš tų gebančių atlikti specialistų – vyresniųjų programuotojų, projektuotojų, testuotojų, projekto vadovo ar kt.
 - e. Ištekliai. Įvertinami projektui skirti ištekliai. Metodas pritaikytas projektams, kurie turi baigtinį išteklių skaičių ir kuris nesikeičia viso projekto eigoje. Turimi ištekliai yra įvertinami pagal tai, kurias projekto užduotis geba atlikti.
 - f. Preliminariai ištekliai gali būti priskirti tam tikroms užduotims. Tai nebus galutinis išteklių pasiskirstymas, tačiau tokia informacija turi būti užfiksuota, jeigu yra, nes gali būti kaip atspirties taškas optimizuojant išteklius.
2. Pagal turimus pradinis duomenis sudėliojamas preliminarus projekto užduočių tinklas.
3. Turimi užduočių laiko įverčiai yra perskaičiuojami vienam ištekliui, jeigu buvo pateikti kitaip.
4. Eliminuojamas užduočių „paslėptas“ buferis ir numatytos užduočių trukmės yra sumažinamos 30 %. Šis parametras gali būti keičiamas, priklausomai nuo turimų žinių apie užduočių vertinimo kultūrą ar kitokias turimas žinias, kiek vidutiniškai specialistai yra linkę pridėti daugiau laiko užduotims įvykdyti. Laikoma, kad likusi užduočių trukmė yra lygi „optimistiškam“ užduočių laiko įverčiui, kai viskas klojasi pagal planą.

5. Pagal sumažintas užduočių trukmes visos užduotys antrojo žingsnio metu sukonstruotame projekto tvarkaraštyje yra pastumiamos link projekto pabaigos – laikomasi „atgalinio“ planavimo principo.
6. Atsižvelgiant į pasiruošimo etape nustatytas užduočių tarpusavio priklausomybes, užduočių įvykdymo seka yra pakeičiama taip, kad užduotys būtų suplanuotos „kaip įmanoma vėliau“ (angl. „as-late-as-possible“) principu.
7. Peržiūrimas projekto tinklas su preliminariai užduotims priskirtais ištekliais. Jeigu konkretus išteklius priskirtas tam tikrai užduočiai, kol kas tai nėra keičiama, nes daroma prielaida, kad preliminariai priskirti ištekliai ne be reikalo, bet remiantis patirtimi – galbūt šis išteklius labiausiai tinkamas atlikti konkrečią užduotį, nors geba atlikti ir kitas. Likusioms užduotims, priskiriami jas gebantys atlikti ištekliai taip, kad vienu metu tas pats išteklius turėtų tik vieną užduotį. Jeigu preliminariai priskirti ištekliai turi daugiau nei dvi užduotis tame pačiame laiko momente, šias užduotis reikia atitinkamai perstumti.
8. Ieškoma laiko tarpų, kai vienas ar daugiau išteklių neturi paskirtos užduoties, t.y. nėra suplanuotas pilnas išteklių panaudojimas. Jei tokių tarpų yra, nagrinėjamos alternatyvos, kaip pasiekti didesnę išteklių paskirstymo efektyvumą, kartu kaip įmanoma mažinant projekto trukmę:
 - a. Galbūt laisvas išteklius geba atlikti kitą užduotį ir tuo metu gali prisijungti prie kitos tuo pačiu metu numatytos vykdyti užduoties. Taip ne tik bus pasiektas didesnis jo užimtumas, bet ir mažės tos užduoties įvykdymo laikas, kai prie jos dirbs daugiau išteklių.
 - b. Nepilnai užimtas išteklius gali „susikeisti vietomis“ su kitu ištekliu, jei abu geba atlikti tą pačią užduotį ir toks apsikeitimas leistų efektyviau panaudoti antrąjį išteklių.
9. Iš visų galimų alternatyvų yra pasirenkama ta, kuri leidžia turėti mažiausią projekto trukmę.
10. Po išteklių perskirstymo kai kurių užduočių trukmė galėjo sutrumpėti, tad visos užduotys atitinkamai pastumiamos link projekto pabaigos per atsiradusius „laisvo“ laiko tarpus.
11. Kai ištekliai ir užduočių pasiskirstymo struktūra galutinai sudėlioti, identifikuojama kritinė grandinė.
12. Įterpiamas projekto buferis gale projekto, lygus pusei kritinės grandinės ilgio.
13. Identifikuojamos „maitinimo“ užduotys – joms vėluojant vėluotų kritinės grandinės užduotys.
14. Įterpiamas „maitinimo buferis“, lygus pusei „maitinimo“ grandinės ilgiui, „maitinimo“ ir kritinės grandinės užduočių sandūroje.

15. Pridėjus šiuos buferius, skirtus susidoroti su netikėtumais projekto eigoje, visos užduotys atitinkamai pastumiamos atgal per buferių ilgius ir gaunama projekto pradžios data.

Pateiktas metodas vadovaujasi kritinės grandinės principais, tačiau taip pat turi keletą savitų šios technikos taikymo ypatybių, kurios yra unikalios lyginant su įprastiniu kritinės grandinės taikymu. Ryškiausias išskirtinumas yra daug didesnis dėmesys išteklių paskirstymui projekto užduotims. Kritinės grandinės technika nuo ankstesnių technikų išsiskiria tuo, kad įtraukia išteklius kaip dar vieną apribojimą projekto užduočių tinkle, tačiau konkretus šios technikos taikymas nėra aprašomas. Tuo tarpu šiame darbe yra pateikti būdai ir alternatyvos, kaip konkrečiai ribotus išteklius galima skirstyti užduotims projekto planavimo etape. Tai leidžia spręsti išteklių kvalifikacijos ir jų kiekio neapibrėžtumą, nes projektų vadovas gali įvertinti, kiek ir kokios kvalifikacijos išteklių reikia numatyti projekto planavimo metu, kad netikėtai pasikeitus ištekliams ar užduočių specifikai, būtų galima vis dar sėkmingai įvykdyti projektą.

Išteklių klausimas yra neatsiejamas nuo skirtingų užduočių savybių panaudojimo, ką taip pat išskiriame kaip svarbų planavimo technikos taikymo ypatumą. Todėl skiriamas taip pat didelis dėmesys pradinei turimai informacijai apie užduotis. Siūlome pasinaudoti šia informacija kaip svarbiu įrankiu planuojant projektą ir užduotis suskirstyti į tipus pagal tam tikrus jų požymius. Tai leidžia susisteminti turimą informaciją ir matyti visas įmanomas alternatyvas, kaip užduotis paskirstyti laike ir taip pat joms priskirti išteklius. Kadangi nėra įmanoma nuspėti visų faktorių, kurie lems projekto eigą, pastangos optimizuoti užduotis ir išteklius leidžia sumažinti projekto trukmę ir išlaikyti vietos buferiams. Tokiu būdu pasiekama verslui priimtina projekto trukmė su „saugikliais“, leidžiančiais įveikti netikėtumus ir sumažinti neapibrėžtumo poveikį projekto rezultatams.

5. Metodo taikymas projekte

Šiame skyriuje aprašoma, kaip patikslintas kritinės grandinės metodas buvo taikomas konkrečiam projektui. Atskleidžiami metodo ypatumai ir taikymo aspektai. Taip pat buvo apskaičiuoti pagrindiniai projekto tvarkaraščio dydžiai – numatoma projekto trukmė, projekto pradžios data – ir palyginti su faktinėmis vertėmis. Tyrimo rezultatai ir išvados pateikiami darbo gale.

5.1. Pradiniai duomenys

Metodo taikymui naudojami pradiniai duomenys – programų sistemų projekto grubus planas ir užduočių įvykdymo įverčiai (angl. *rough estimate*). Taip pat yra žinoma, kada projektas prasidės

ir kada turi būti pabaigtas. Projekto vykdymui yra numatyti tam tikri žmogiškieji ištekliai ir jie yra žinomi prieš taikant metodą.

Taigi turimas sąrašas stambių užduočių, kurias reikės atlikti, kad projektas būtų sėkmingai įgyvendintas. Egzistuoja užduočių tarpusavio priklausomybė ir kai kurios užduotys negali būti pradėtos, kol nebus įvykdytos ankstesnės; kai kurios užduotys gali būti vykdomos vienu metu. Susistemintas užduočių sąrašas su numatyta jų atlikimo trukme pateiktas pirmoje lentelėje. Užduočių pavadinimai pakeisti į identifikacinius numerius, kad būtų išlaikytas konfidencialumas ir būtų paprasčiau sisteminti metodo žingsnius. Šioje lentelėje taip pat matomas žmogiškųjų išteklių skaičius vertinant užduoties trukmę. Tai reiškia, kad pavyzdžiui užduotis U1 turėtų būti įvykdyta per dvi savaites, jeigu prie jos dirbs du žmonės, U2 bus įvykdyta per dvi savaites, jei prie jos dirbs vienas žmogus ir t.t.

1 lentelė. Tiriamo projekto užduotys, numatyta data, jų trukmė savaitėmis ir išteklių skaičius trukmės įverčiui

Užduotis	Data (savaitės nr.)	Trukmė (savaitėmis)	Numatyta išteklių (žm.sk.)
U1	4-5	2	2
U2	4-5	2	1
U3	5-7	2	1
U4	6-8	3	2-3
U5	9	1	2
U6	8-12	5	1
U7	13-14	2	1
U8	8-10	3	2
U9	11-14	4	2
U10	15-19	5	n/a

Numatyta projektą baigti 19-ąją savaitę. Pateiktame sąrašė nėra formalių duomenų apie 1-3 savaitę vykdomas užduotis, bet yra laikoma, kad pirmą mėnesį vyko projekto paruošiamieji darbai: planavimas, darbų derinimas su klientu ir kt. Pagal faktinius duomenis, analizuojamas projektas vėlavo keturias savaites.

Projektui priskirti žmogiškieji ištekliai iš vykdytojo pusės: trys programuotojai, vienas projektų vadovas. Iš užsakovo pusės priskirti ištekliai: sistemų analitikai ir testuotojų komanda.

5.2. Duomenų interpretavimas

Visų pirma suklasifikuojamos projekto užduotys. Turimos žinios apie užduočių specifiką, tik jos čia specialiai nėra įvardijamos (dėl konfidencialumo išsaugojimo). Pagal šias turimas žinias užduotys priskiriamos dviem grupėms pagal tai, ar jos gali būti vykdomos vieno ištekliu, ar

kelių. Pagal šį kriterijų užduotys U2 ir U3 gali būti atliktos tik vieno ištekliaus, o likusios gali būti atliekamos visų trijų turimų programuotojų.

Toliau užduotys priskiriamos grupėms pagal tai, ar jos gali būti vykdomos kartu su kitomis užduotimis, ar nuosekliai. Iš to išplaukia ir užduočių tarpusavio ryšių identifikavimas. Pagal šį kriterijų visos užduotys gali būti atliekamos kartu su kitomis, tačiau su tam tikrais apribojimais. Užduotys U1, U2 gali būti atliekamos vienu metu nepriklausomai nuo to, kuri bus pradėta anksčiau. Užduotys U3-U6 gali būti atliekamos vienu metu, tačiau U4-U6 gali būti pradėtos tik po to, kai U1 ir U2 bus abi baigtos, o U3 gali būti pradėta įpusėjus U1, U2 užduotims ir turi būti baigta iki U8 pradžios. Yra žinoma, kad svarbu U6 baigti įpusėjus U9. U7 gali būti pradėta po to, kai prieš ją einančios užduotys atliktos; U8 gali būti pradėta praėjus savaitei po U4 vykdymo pradžios ir negali baigtis anksčiau nei U4-U6 užduotys pabaigtos. Užduotis U9 gali būti pradėta tik atlikus U8 ir gali būti vykdoma kartu su U10. Svarbu pažymėti, kad šie užduočių ryšiai identifiukuoti, neatsižvelgiant į turimus išteklius, užduotims atlikti.

Antroje lentelėje pavaizduotos užduotys išdėstytos laiko skalėje taip, kaip buvo numatyta projekto vadovo. Kadangi 1-3 savaitėmis vyko paruošiamieji darbai, paprastumo ir aiškumo dėlei čia laiko skalė normalizuota ir pavaizduota nuo pirmos savaitės, pastumiant užduotis per tris savaites atgal. Taigi galima tarti, kad projektas numatytas atlikti per 16 savaičių.

Užduočių tarpusavio ryšių identifikavimas buvo reikalingas daugelyje metodo taikymo žingsnių. Šalia šių duomenų, buvo atlikta ir projekto išteklių analizė.

2 lentelė. Suplanuota projekto užduočių trukmė ir jų paskirstymas laiko skalėje

Užduotis	Savaitė															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1	■	■														
U2	■	■														
U3			■	■												
U4			■	■	■											
U5						■										
U6					■	■	■	■	■							
U7										■	■					
U8					■	■	■									
U9								■	■	■	■					
U10												■	■	■	■	■

Ištekliai suklasifikuoti pagal tai, kokias užduotis geba atlikti. Čia svarbu išskirti gebėjimą ir „galėjimą“ atlikti. Nebūtinai išteklius, gebantis atlikti U1 ir U2 galės jas atlikti. Šios užduotys gali būti atliekamos vienu metu ir tas pats išteklius negalės jų abiejų atlikti, jeigu jos numatytos vykdyti

tuo pačiu laiku. Taigi šioje pastraipoje suskirstėme išteklius į grupes pagal tai, ką jie „teoriškai“ geba atlikti pagal savo kompetenciją, nepriklausomai nuo to, ką norėtų atlikti ir ar turėtų laiko atlikti. Paašškėjo, kad pirmas išteklius (pavadintas R1) geba atlikti visas užduotis, išskyrus U10. Ištekliai R2, R3 geba atlikti visas užduotis, išskyrus U2, U3, U10. Pastaroji užduotis (U10) gali būti atlikta tik kliento numatytų išteklių, todėl tam tikra prasme projekto vykdytojas neturi kontrolės šios užduoties vykdymui. Dėl šios priežasties būtų galima išimti šią užduotį iš analizuojamų duomenų, tačiau ji dar gali būti svarbi tolesniam metodo taikymui, todėl kol kas ji matoma projekto duomenyse. Užduotys su jas galinčiais atlikti ištekliais pavaizduotos trečioje lentelėje. Išteklius R4 atitinka kliento išteklius užduočiai U10 atlikti.

3 lentelė. Projekto užduotys ir ištekliai, gebantys jas atlikti

Užduotis	Ištekliai
U1	R1, R2, R3
U2	R1
U3	R1
U4	R1, R2, R3
U5	R1, R2, R3
U6	R1, R2, R3
U7	R1, R2, R3
U8	R1, R2, R3
U9	R1, R2, R3
U10	R4

Yra laikoma, kad projektų vadovas ir kiti projekto planavime dalyvaujantys asmenys pateikdami užduočių įvykdymo numatomus laikus, prideda „paslėptą“ buferį. Tai reiškia, jog jei viskas klostytųsi pagal planą, užduotį būtų galima atlikti per trumpesnę laiką, nei deklaruota. Kritinės grandinės autorius Goldratt siūlo šią problemą spręsti keičiant organizacijos kultūrą ir joje dirbančio individo elgseną, t.y. skatinant darbuotojus vertinti užduočių laikus be paslėpto buferio; akcentuojant ankstesnę užduoties atlikimą, o ne vėlavimo neigiamus padarinius. Mes siūlome sumažinti užduočių laiko įverčius statistiniu dydžiu - kokią dalį vidutiniškai žmonės prideda prie numatytos įgyvendinti užduoties, iš karto darant prielaidą, kad darbuotojai prideda „paslėptą“ buferį. Šis dydis gali būti keičiamas, priklausomai nuo organizacijos kultūros, jos darbuotojų susiformavusios elgsenos ypatumų, o šiuo atveju yra pasirinktas 30%. Taigi turint preliminarų užduočių vykdymo trukmių sąrašą, šie įverčiai sumažinti trečdaliu, tokiu būdu laikant, kad paslėpti buferiai buvo eliminuoti.

5.3. Užduočių perplanavimas

Kadangi siekiama eliminuoti „paslėptus“ buferius, tai pagal aukščiau išvardintas prielaidas, antroje lentelėje pavaizduotos užduočių trukmės nuo šiol sumažinamos 30%. Visos užduotys yra paslenkamos arčiau numatyto projekto pabaigos. Toks požiūris vadinamas „atgaliniu“ planavimu (angl. *backwards planning*) ir yra būdingas kritinės grandinės technikos taikymui. Toks požiūris yra svarbus, nes siekiama ne ištesti viso projekto trukmę, bet priešingai – kaip įmanoma greičiau atlikti projektą laiku. Taigi pastūmus užduotis toliau, projekto pradžia atidedama vėlesniam laikui. Šis postūmis yra tik pradinis žingsnis ir yra laikinas. Taikant tolesnius žingsnius ir optimizuojant išteklius, jis keičiasi.

Atėmus „paslėptus“ buferius ir laikantis „atgalinio“ planavimo principų, atnaujintas užduočių išsidėstymas yra pateiktas ketvirtoje lentelėje. Čia užduotis U7 pastumta į projekto plano galą, nes ji neįtakoja tolesnių užduočių, o U1 ir U2 paslinktos į dešinę dėl to, kad U3 gali būti pradėta įpusėjus U1, U2 ir yra siekiama projekto trukmę išlaikyti kiek įmanoma mažesnę. Visa tai sudėjus, šiame etape projekto užduočių tinklo užduočių trukmės nėra tik sutrumpintos trečdaliu, bet ir pagal tarpusavio priklausomybes jos paslinktos prie projekto pabaigos ir todėl tokio plano struktūra šiek tiek skiriasi, nei buvo numatyta duomenų analizės pradžioje. Iš čia išryškėja kritinis kelias, tačiau nebūtinai kritinė grandinė. Lentelėje šviesiai pilka spalva pažymėtas pradinis numatytas užduočių pasiskirstymas, buvęs prieš projekto užduočių tinklo koregavimą.

4 lentelė. Užduočių trukmės ir jų pasiskirstymas laike be „paslėpto“ buferio

Užduotis	Savaitė															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1					■	■	■									
U2					■	■	■									
U3						■	■	■								
U4							■	■	■							
U5									■							
U6										■	■	■	■			
U7																■
U8										■	■	■				
U9												■	■	■		
U10																■

Skaičiuojant šio etapo kritinį kelią projektas truktų apie 13 savaičių. Toliau reikia paskirstyti užduotis ištekliams. Pagal analizuojamo projekto pirminį planą užduotį U1 paskirta atlikti ištekliams R1 ir R2, U2, U3 – ištekliui R1. Toliau nebuvo tiksliai numatyta, kas atliks kurią užduotį. Tačiau aukščiau yra identifikuota informacija apie tai, kokias užduotis, kokie ištekliai

geba atlikti. Naudojantis šia informacija galima optimizuoti išteklių paskirstymą užduotims taip, kad per kuo trumpesnę laiką projektą būtų įmanoma įvykdyti.

5.4. Išteklių optimizavimas

Svarbu pažymėti, kad nagrinėjamas programų sistemų projekto atvejis, kai visi priskirti išteklių dirba prie vieno projekto iki jo pabaigos. Tai reiškia, kad nebus nagrinėjami atvejai, kai pavyzdžiui, analitikas atlikęs analizės darbus yra perskiriamas kitam projektui šiam nepasibaigus arba testuotojas yra priskiriamas projektui tik testavimo etapo metu ir t.t.

Jeigu pirmoms trims užduotims priskiriami išteklių pagal jų gebėjimą šias užduotis atlikti, išteklių bent jau šioms užduotims bus optimizuoti. Tokiu atveju užduotį U1 vykdys išteklių R2 ir R3, o užduotis U2, U3 vykdys R1. Likusioms užduotims priskirti išteklius sudėtingiau, kadangi visi išteklių geba atlikti skirtingas užduotis.

Paprastai projekto planavimo technikoje vienai užduočiai priskiriamas vienas išteklius jai atlikti. Tačiau reikia išanalizuoti galimybę užduotį priskirti keliems ištekliams, kai ši užduotis gali būti vykdoma kelių išteklių.

Kad būtų paprasčiau perskaičiuoti užduočių trukmę, jas atliekant skirtingam išteklių skaičiui, pirmoje lentelėje matomus duomenis galima sunormalizuoti ir kiekvienai užduočiai paskaičiuoti, per kiek ji bus atlikta ją vykdant vienam ištekliui (5 lent.). Kadangi užduočiai U4 preliminariai įvertinta 2-3 savaitėmis, tai imamas šio įverčio vidurkis 2,5 ir dėl to trukmė vienam ištekliui apskaičiuota 7,5 savaitės.

5 lentelė. Užduočių trukmės vienam ištekliui

Užduotis	Trukmė, kai dirba 1 išteklius (savaitėmis)
U1	4
U2	2
U3	2
U4	7,5
U5	2
U6	5
U7	2
U8	6
U9	8
U10	5

Nagrinėjamu atveju yra trys galimybės užduočiai U4 priskirti išteklius:

1. Priskirti užduotį vykdyti vienam iš išteklių, pavyzdžiui R1, kadangi tuo metu jis turėtų būti baigęs užduotį U3.

2. Priskirti užduotį atlikti dviem ištekliams vienu metu.
3. Priskirti užduotį atlikti trimis ištekliams vienu metu.

Renkantis pirmą variantą, iškyla dar keletas galimybių – kurį išteklių priskirti užduočiai – R1, R2 ar R3. Išteklius R1 priskyrimas gali nebūti geriausias sprendimas, nes jeigu jis vėluos atlikti užduotį U3 arba U2, tuomet jis vėluos pradėti U4.

Iš esmės norint optimaliai paskirstyti išteklius, reikia stengtis sutrumpinti kritinę grandinę, kiek tai yra įmanoma. Tokiu būdu visa projekto trukmė bus mažiausia.

Siekiant efektyviai panaudoti išteklius ir atsizvelgiant į jų gebėjimus atlikti skirtingas užduotis, toks jų paskirstymas pavaizduotas 6-oje lentelėje. Iš čia matyti, kad ištekliai dar nėra optimizuoti: išteklius R1 devintą savaitę turi dvi užduotis, kas nėra korektiška, dėl to užduotį U5 reikėtų paslinkti atgal, t.y. paankstinti. Tačiau dėl užduočių tarpusavio priklausomybių viso projekto trukmė išsitęstų. Taigi kyla natūralus klausimas, kaip perskirstyti išteklius, kad to būtų išvengta. Ta pati problema matoma 5-6 savaites, kai išteklius R1 turi dvi užduotis vienu metu. Kadangi užduotis U3 turi būti baigta iki U8 pradžios, šią užduotį būtų galima pastumti į dešinę laiko skalės atžvilgiu ir taip išvengti projekto trukmės padidėjimo. Kita minėtoje lentelėje matoma problema arba neišnaudota galimybė yra 12-13 savaitės, kai išteklius R1 neturi užduoties, taigi jo darbas nėra maksimaliai išnaudotas. Tas pats matyti ir 7-8 savaitėmis, o 6-os savaitės pusė neišnaudota R2, R3 išteklių atžvilgiu.

6 lentelė. Preliminarus išteklių priskyrimas užduotims

Užduotis	Savaitė															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1					R2, R3											
U2				R1												
U3						R1										
U4							R2, R3									
U5									R1, R2							
U6									R1							
U7																R2
U8									R2, R3							
U9											R2, R3					
U10																R4

Siekiant išspręsti minėtas problemas, ištekliai vėl perskirstomi užduotims. Kadangi užduotis U3 gali būti pradėta ir vėliau, ji pastumiamą į dešinę taip, kad išteklius R1 turėtų tik vieną užduotį vienu metu. Kol kas šią užduotį galima slinkti iki 9-osios savaitės pradžios, kai R1 jau turi kitą užduotį, tačiau toks poslinkis nebūtinas. Paslinkus užduotį aptartu atstumu, atsiranda nedidelis

laiko tarpas, esantis 6-7 savaitėms, dėl to užduotis U1 ir U2 galima kartu pastumti į dešinę iki U3 pradžios (7 lent.).

7 lentelė. Tarpinis užduočių postūmis

Užduotis	Savaitė															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1						R2, R3										
U2						R1										
U3							R1									
U4							R2, R3									
U5								R1, R2								
U6									R1							
U7																R2
U8									R2, R3							
U9											R2, R3					
U10																R4

Toliau nagrinėjamos alternatyvos, kaip būtų galima perskirstyti išteklius užduotims U5, U6, U8 ir U9, kad būtų išvengta projekto trukmės padidėjimo dėl U5 pastūmimo atgal. Kol kas šių užduočių projekcija laiko ašyje siekia 4,64 savaites. Yra keletas alternatyvų perskirstyti išteklius šioms užduotims:

1. Užduočiai U5 pridėti dar vieną išteklį ir taip sumažinti jos trukmę trečdaliu, viso $2 \cdot 0,66/3 = 0,44$ savaitės. Tokiu atveju šią užduotį reikėtų pastumti atgal mažesniu atstumu ir reikėtų užtikrinti, kad likusieji ištekliai neturėtų tuo metu kitų užduočių. Galiausiai visa projekto trukmė pailgėtų per naująjį U5 dydį, kai prie jos dirba trys žmonės.
2. Užduočiai U6 paskirti vienu ištekliau daugiau (iš viso du ištekliai) ir taip sumažinti šios užduoties trukmę pusiau – naujoji trukmė būtų lygi 1,65 savaitėms ($5 \cdot 0,66/2$). Tačiau tuo metu R2, R3 jau yra užimti, dėl to prieš tai einančios užduotys turi būti pastumtos atgal. Labiausiai sumažinti šį postūmį atgal būtų galima užduotims U8 ir U9 priskyrus visus išteklius ir taip sumažinant jų trukmę trečdaliu. Vis dėlto tokiu atveju bendra užduočių U5, U6, U8 ir U9 trukmė būtų didesnė nei pirmosios alternatyvos ir siektų:

$$0,66 + 1,65 + 1,32 + 1,76 = 5,39 \text{ (sav.)}$$

3. Galima būtų visoms užduotims U5-U9 priskirti visus galimus išteklius. Tuomet atėmus „paslėptą“ buferį ir atsižvelgiant į ištekliau skaičių, U5 būtų atlikta per $2 \cdot 0,66/3$ savaitės, U6 – per $5 \cdot 0,66/3$, U7 – per $2 \cdot 0,66/3$, U8 – per $6 \cdot 0,66/3$, U9 – per $8 \cdot 0,66/3$ savaitės (8 lent.). Kadangi visos šios užduotys turėtų eiti nuosekliai, bendras laikas joms atlikti siektų

5,06 savaitės ir dėl to projektas išsitęstų per 5,06 – 4,64 (sav.), t.y. 0,42 savaites. Tai yra mažesnis postūmis nei pirmoji alternatyva.

4. Kita alternatyva yra skelti užduotį U6 į dvi dalis, kadangi ši užduotis gali būti atliekama dalimis. Pirmoji dalis sieks tiek, kiek laisvo laiko turi R1 8-ąją savaitę, t.y. 0,5 savaitės. Užduotis U6 be „paslėpto“ buferio numatyta atlikti viso per 3,3 savaites dirbant vienam ištekliui, taigi antroji šios užduoties dalis būtų lygi 2,8 savaitėmis. Pirmąją U6 užduoties dalį jis atliktų 8-ą savaitę, o antrąją dalį 10-12 savaitėmis. Jeigu užduočiai U5 priskiriamas tik R1, tuomet jos trukmė padidėtų iki 1,32. U6 turi būti baigta iki U9 pusės, t.y. pusės 2,64 dydžio (8 lent.) – 1,32 savaitės. Kadangi antroji U6 dalis užimtų 2,8 savaites, tai 9-10 savaitėmis ši užduotis užimtų dar 2,8 - 1,32 (sav.) laiko, t.y. 1,48 sav. Reiškia tuo pačiu laikotarpiu užduočiai U5 lieka (2 - 1,48), t.y. 0,52 (sav.) laiko ir U5 reikia pastumti atgal per (1,32 – 0,52) savaites, t.y. 0,8 sav. Tai yra didesnis postūmis nei pirmoji ir antroji alternatyva.

Nepaisant to, pirma ar ketvirta alternatyva pasirinkta, yra dar viena neišnaudota galimybė, padėsianti sumažinti bendrą projekto vykdymo trukmę. Kadangi U6 turi būti baigta iki U9 įvykdymo pusės, tai pastarosios užduoties antrajai pusei galima priskirti visus galimus išteklius. Antroji užduoties dalis bus įvykdyta per 1,32 savaites, jei prie jos dirba du ištekliai, bet jei dirba trys ištekliai, tai ši dalis bus įvykdyta per:

$$\frac{1,32 \times 2}{3} = 0,88 \text{ (sav.)}$$

Taip paskirsčius išteklius, projekto trukmė sumažėtų dar per 0,44 savaites, o bendra U9 trukmė taptų 2,2 sav.

Visų užduočių trukmės prie jų dirbant skirtingam išteklių skaičiui pavaizduoti 8 lent. Čia yra atimtas „paslėptas“ buferis, t.y. reikšmės padaugintos iš 0,66 dydžio. Paskutinė užduotis vykdoma iš kitų (kliento) išteklių ir nėra žinoma, kiek jų buvo paskirta šiai užduočiai vykdyti, dėl to įverčiai šiai užduočiai nėra pateikti.

8 lentelė. Užduočių numatomos trukmės, dirbant skirtingam išteklių skaičiui

Užduotis	Trukmė 1 ištekliui	Trukmė 2 ištekliui	Trukmė 3 ištekliui
U1	2,64	1,32	0,88
U2	1,32	0,66	0,44
U3	1,98	0,99	0,66
U4	4,95	2,48	1,65
U5	1,32	0,66	0,44
U6	3,3	1,65	1,10
U7	1,32	0,66	0,44
U8	3,96	1,98	1,32
U9	5,28	2,64	1,76
U10	--	--	--

Apibendrinant alternatyvų analizės rezultatus, galima pasakyti, jog pirma alternatyva yra geriausia, jeigu norime sumažinti projekto trukmę (9 lent.). Analizuojant situaciją ne kiekybiškai, bet kokybiškai, nebūtinai projektų vadovas turi siekti trumpiausios projekto trukmės bet kokia kaina. Gali kilti klausimas priklausomai nuo projekto ir jo užduočių specifikos, ar tikrai vieną išteklių priskirti tam tikrai užduočiai. Jeigu yra didelis užduoties neapibrėžtumas, gali būti per daug rizikinga užduočiai skirti tam tikrą išteklių, net jei dėl to visa projekto trukmė sumažėtų. Reikėtų siekti rizikos ir suplanuoto laiko balanso.

9 lentelė. Alternatyvų analizės rezultatų apibendrinimas

Projekto trukmės pokytis (sav.)	A1	A2	A3	A4
Dėl U5	+0,44	+0,75	+0,42	+0,8
Dėl U9	-0,44	--	--	-0,44
Viso:	0	+0,75	+0,42	+0,36

Pasirinkus pirmąją išteklių perskirstymo alternatyvą, nubraižomas naujas projekto užduočių planas (10 lent.).

10 lentelė. Projekto užduočių ir išteklių pasiskirstymas pasirinkus 1-ąją alternatyvą

Užduotis	Savaitė															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1					R2, R3											
U2					R1											
U3						R1										
U4						R2, R3										
U5								R1, R2								
U6									R1							
U7															R2	
U8									R2, R3							
U9										R2, R3		R1, R2, R3				
U10															R4	

Kad būtų geriau suprasti, kiek kokia užduotis truks pasirinkus šią alternatyvą, jų trukmės pavaizduotos 11 lent. Pilka spalva paryškintos laiko vertės šios alternatyvos atveju.

11 lentelė. Projekto užduočių trukmės 1-osios alternatyvos atveju

Užduotis	Trukmė 1 ištekliui	Trukmė 2 ištekliui	Trukmė 3 ištekliui
U1	2,64	1,32	0,88
U2	1,32	0,66	0,44
U3	1,98	0,99	0,66
U4	4,95	2,48	1,65
U5	1,32	0,66	0,44
U6	3,3	1,65	1,10
U7	1,32	0,66	0,44
U8	3,96	1,98	1,32
U9	5,28	2,64	1,76
U10	--	--	--

Iš projekto užduočių ir išteklių paskirstymo lent. 10 galima matyti dar kelis laiko tarpus, kai ne visi ištekliai pilnai panaudojami. Reikėtų pamėginti priskirti R1 išteklių užduočiai U4 nuo to laiko, kai šis išteklius baigs vykdyti U3. Tokiu būdu šios nedidelės laiko atkarpos dalis (8-ąją savaitę) sumažėtų trečdaliu.

5.5. Buferių pritaikymas

Tolesnis žingsnis buvo užtikrinti, kad projekto eigoje, nepaisant netikėtų įvykių ir aplinkybių, bus galima suvaldyti projektą taip, kad vėluojant tam tikroms užduotims ir pasikeitus išteklių ar jų kompetencijų poreikiui, visas projektas būtų įgyvendintas laiku. Kritinės grandinės technika siūlo taikyti įvairius buferius neapibrėžtumui valdyti projekte. Tam visų pirma reikia

identifikuoti kritinę grandinę. Nagrinėjamu projektu atveju kritinė grandinė galėtų būti užduočių U1, U4, U5, U8-U10 seka.

Pirmiausia buvo pridėtas projekto buferis, kuris yra lygus pusei kritinės grandinės ilgio. Šiuo atveju kritinė grandinė sudaro 13,64 savaites ir projekto buferis sieks 6,82 savaites. Šis buferis buvo pridėtas kritinės grandinės gale (12 lent.). Tokiu atveju visa projekto planuojama trukmė tapo lygi 20,46 savaitėms.

Toliau nustatytas „maitinimo“ buferis, skirtas apsaugoti kritinę grandinę nuo užduočių, nesančių joje, vėlavimų. Šis buferis yra lygus pusei „maitinimo“ užduočių įvykdymo trukmei. Tokios užduotys yra tos, dėl kurių vėlavimo, nebus galima laiku pradėti užduoties, kuri patenka į kritinę grandinę. Tokios užduotys nagrinėjamu atveju buvo U2, U3, U6. Jos buvo planuotos „kaip įmanoma vėliau“ (angl. „as-late-as-possible“) principu ir jeigu būtų pavėluota jas įvykdyti laiku, būtų vėluojama pabaigti visą projektą. „Maitinimo“ buferiai pridedami jų susikirtimų taškuose su kritinės grandinės užduotimis.

12 lentelė. Užduočių planas su projekto buferiu

Užduotis	Savaitė																		
	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1	R2, R3																		
U2	R1																		
U3			R1																
U4			R2, R3																
U5					R1, R2														
U6						R1													
U7												R2							
U8						R2, R3													
U9								R2, R3	R1	R2, R3									
U10											R4								
Projekto buferis																			

Taigi pirmąjį „maitinimo“ buferį galima įterpti U3 ir U4 užduočių sankirtoje ir jis yra lygus U2 ir U3 trukmių sumos pusei. Šiuo atveju U3 ir U4 sudaro taip vadinamą „maitinimo grandinę“. Antrasis „maitinimo“ buferis pridedamas užduoties U6 ir U9 antros dalies susikirtimo taške. Pastarąją užduotį paprastumo dėlei galima skaldyti į dvi užduotis ir vadinti U9A ir U9B. Projekto planas su „maitinimo“ buferiu pavaizduotas 13 lent. Užduotis U7 nepriklauso „maitinimo grandinės“ užduotims, nes ji niekaip neįtakoja kritinės grandinės ir apskritai dėl jos vėlavimo projektas nevėluotų, todėl galima atsisakyti ją nagrinėti šiuo tyrimo atveju.

13 lentelė. Projekto planas su “maitinimo” ir projekto buferiais

Užduotis	Savaitė																					
	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
U1	R2, R3																					
U2	R1																					
U3			R1																			
Mait. Buferis 1																						
U4				R2, R3																		
U5						R1, R2																
U6							R1															
Mait. Buferis 2																						
U7															R2							
U8								R2,														
U9A										R2, R3												
U9B											R1, R2, R3											
U10																						
Projekto buferis																						

Iš lentelės matoma, kad projekto planuojama įvykdymo trukmė padidėjo šiek tiek daugiau nei per 5 savaites. Užfiksuotas faktinis nagrinėjamo projekto vėlavimas – 4 savaitės. Taip pat svarbu pažymėti, kad paskutiniuosius projekto eigos savaitėmis buvo skirta dvigubai daugiau išteklių, nei viso projekto eigoje. Taikant šį patikslintą kritinės grandinės metodą planuojama projekto trukmė (įskaitant buferius) lygi 20,46 savaitės. Netaikant metodo, buvo numatyta projektą atlikti per 19 savaitių, o pagal faktinius duomenis jis atliktas per 23 savaites, paskutines 4 savaites skiriant dvigubai daugiau išteklių, nei buvo numatyta projekto pradžioje. Sunku įvertinti per kiek laiko projektas būtų pasibaigęs neskiriant papildomų išteklių jo pabaigoje.

Lyginant faktinę projekto trukmę su planuota trukme pritaikius metodą, buvo pasiekta apie **11%** greitesnis numatytas projekto įvykdymas. Šis dydis apskaičiuotas, neatsižvelgiant į didesnę išteklių kiekį skirtą projekto pabaigoje. Galime galvoti apie kelis skirtingus šių papildomų išteklių poveikius. Vienas jų – papildomas išteklių priskyrimas tiesiogiai padidino projekto vykdymo greitį. Tokiu atveju padvigubinus išteklių skaičių, dvigubai greičiau projektas įvykdytas tame laikotarpyje. Tuomet projekto vykdymo vėlavimas be šių papildomų išteklių būtų siekęs 8 savaites ir taip padidinęs projekto trukmę taikant metodą iki apytiksliai **24%**. Laikysime tai maksimaliu šio metodo efektyvumo matu, kai lyginame projekto plano trukmę su faktine projekto įvykdymo trukme.

Vertindami projekto planavimo ar jo trukmės prognozavimo efektyvumą, lyginame planuojamą projekto trukmę taikant metodą (20,46 sav.) su planuota projekto trukme, kai netaikomas joks kritinės grandinės technikos variantas (19 sav.). Taikant mūsų metodą planuojama projekto trukmė išauga **7,7%**. Nors ilgesnė projekto trukmė nėra patraukli verslui, tačiau

realistiškesnis projekto planas ir projekto baigimas laiku yra geresnė kombinacija nei optimistiškesnis projekto planas ir pavėluotas projekto įvykdymas.

REZULTATAI IR IŠVADOS

Apibrėžtas kritinės grandinės metodas, leidžiantis planuoti projektą neapibrėžtumo sąlygomis didesnę dėmesį skiriant projekto ištekliams ir jų gebėjimui atlikti tam tikras užduotis. Išgrynintos galimos projektų užduočių grupės ir parodyta, kaip galima pasinaudoti turima informacija apie užduotis ir projektui skiriamus išteklius. Metodas sukurtas remiantis programų sistemų projektų niuansais ir nebūtinai gali tikti kitų sričių projektams.

Pritaikius metodą realiam programų sistemų projektui, jo planuojama trukmė buvo panaši į faktinę projekto įvykdymo trukmę, atsižvelgiant į tai, kad analizuojamas projektas vėlavo apie mėnesį ir jam baigiantis buvo skirta daugiau išteklių nei numatyta planavimo etape. Šiuo požiūriu metodas **nepadeda sumažinti planuojamos projekto trukmės**, tačiau pateikia ją artimesnę tikrovei.

Metodo nauda labiausiai atsiskleidžia **efektyvesniu projekto trukmės prognozavimu**. Šiuo metodu pavyko pateikti realesnę jo trukmės prognozę. Tai reiškia, kad nustatyta projekto pradžia yra labai panaši į tą, kuri turėjo būti pasirinkta, kad projektą pabaigti laiku. Tai yra labai svarbus aspektas, nes numatyti teisingą projekto trukmę su turimais ištekliais ir užduočių mastu leidžia ir baigti projektą laiku. Priešingu atveju projekto eigoje gali prireikti daugiau išteklių, neįvykdžius projekto laiku vykdytojui gali grėsti finansinės baudos, gali prastėti jo reputacija, tad šis rezultatas yra ypač aktualus.

Patikslintas kritinės grandinės metodas leidžia **maksimaliai panaudoti numatytus fiksuotus išteklius** projektui. Tai buvo galima pasiekti universaliai naudojant išteklius skirtingoms užduotims pagal išteklių gebėjimą jas atlikti. Šis požiūris nėra įprastas, nes čia tam tikru metu vienas išteklius gali atlikti nebūtinai savo kompetenciją tiesiogiai atitinkančią užduotį (pvz., sistemų analitikas gali testuoti). Toks išteklių panaudojimas nebūtinai gali būti įgyvendinamas ne informacinių sistemų organizacijose.

Visų galimybių nagrinėjimas išteklių kvalifikacijos klausimu leidžia spręsti **išteklių neapibrėžtumo** problemą projektuose, nes projektų vadovas turi visą informaciją apie turimus išteklių gebėjimus ir numatytų užduočių specifiką – kvalifikacijos poreikį. Išteklių klasifikavimas jų gebėjimų atžvilgiu sudaro sąlygas turėti įvairias projekto užduočių tinklo sudarymo **alternatyvas** ir suteikia **daugiau galimybių spręsti išteklių konfliktus**. Tokiu būdu yra iš anksto pasiruošama galimam išteklių kiekiui, jų kvalifikacijos ar užduočių specifikos kaitos galimybei projekto eigoje. Tai yra lankstesnis nei įprastas kritinės grandinės metodas ir projektų vadovas gali komplektuoti išteklius taip, kad tai leistų maksimizuoti galimybes projektą baigti laiku, atsiradus nenumatytoms aplinkybėms. Tai ypač aktualu, kai projektų vadovas gali laisvai rinktis išteklius projekto vykdymui.

Metodas pritaikytas tais atvejais, kai išteklių kiekis yra ribotas ir projekto vykdymo eigoje ištekliai nesikeičia. Tikime, kad tai yra realesnė situacija, nei projektas su neribotais ištekliais, kurių atvejis dažnai nagrinėjamas mokslinėje literatūroje. Kadangi ištekliai yra riboti, geriausia pasiekti didžiausią jų panaudojimą, kad neliktų „prastovų“. Šis požiūris prieštarauja kritinės grandinės išteklių buferiui (kai nekritiniai ištekliai laikomi nepilnai apkrauti), nes siekėme parodyti **kvalifikacijos svarbą užduotims**. Netikėtai prireikus papildomo išteklių užduočiai, nepilnai apkrautas nekritinis išteklius neišspręs problemos, jei jis neturės gebėjimų atlikti šią užduotį.

Straipsniuose giriančiuose kritinės grandinės techniką, nėra kreipiamas pakankamas dėmesys patiems ištekliais – jie dažniausiai yra nevaržomai perskirstomi ir taip pavyksta bendrą projekto trukmę lengvai sumažinti. Šiuo darbu buvo siekiama pateikti būdą planuoti projektą, kuris atspindėtų realias galimybes sudėlioti projekto užduotis ir išteklius taip, kad projekto trukmė būtų mažiausia, bet kartu ir apsaugant projektą nuo netikėtumų ir neapibrėžtumo poveikio jo eigoje.

Šis darbas ne tik atskleidžia kitokį požiūrį į išteklių paskirstymą planuojant projektus, bet ir atveria kitas tyrimų galimybes. Toliau galima nagrinėti išteklių grupavimą ne tik pagal gebėjimų tipus, bet ir pagal gebėjimų lygį. Taip būtų įvesta dar daugiau apribojimų siekiant sudaryti tinkamiausią tvarkaraštį.

Kitas galimų tyrimų objektas galėtų būti projektų planavimas, remiantis kokybinio planavimo principais, tai yra nebūtinai siekiant mažesnės projekto trukmės, bet žvelgiant plačiau ir išskiriant teigiamus tokio planavimo aspektus (galbūt didesnė projekto trukmė yra geriau, jei ištekliai gali būti efektyviau panaudoti kituose projektuose ir pan.).

Siekiant geriau įvertinti patikslinto metodo naudą, būtų galima jį pritaikyti daugiau įvairių programų sistemų projektų ir įvertinti to rezultatus. Be to, būtų vertinga pritaikyti šį metodą planuojant naują projektą, kai šis dar nėra pasibaigęs. Galbūt projekto eigoje toks tvarkaraščio sudarymo būdas atskleistų naujų niuansų. Svarbu pažymėti ir tai, kad suplanuota projekto trukmė pagal šį metodą nebūtinai išliks tokia ir projekto eigoje. Nežinoma yra ir tai, kaip projektų vadovas sugebėtų išnaudoti numatytus buferius bei kaip perplanuotų užduotis, atsiradus nenumatytiems veiksniams. Visgi šio tyrimo tikslinė problematika buvo projektų planavimo etape, tad minėtų klausimų atsakymų reikėtų ieškoti projektų valdymo, bet ne planavimo tyrimuose.

ŠALTINIAI

[AR00] Christian Artigues, François Roubellat, A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes, 2000, prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S0377221799004968>

[ASB10] Pongtip Aroonvatanaporn, Chatchai Sinthop, Barry Boehm, *Reducing Estimation Uncertainty with Continuous Assessment: Tracking the “Cone of Uncertainty”*, 2010, prieiga per internet: <http://csse.usc.edu/csse/TECHRPTS/2010/usc-csse-2010-525/usc-csse-2010-525.pdf>

[BPS04] Bertsimas D., Pachamanova D., Sim M., *Robust linear optimization under general norms*, 2004, [žiūrėta 2014-12-06], prieiga per internetą: <http://www.mit.edu/~dbertsim/papers/Robust%20Optimization/Robust%20Linear%20Optimization%20under%20General%20Norms.pdf>

[Boe88] Barry W. Boehm, A spiral model of software development and enhancement, 1988, [žiūrėta 2015-05-24], prieiga per internetą: http://portal.ou.nl/documents/114964/2986739/T24331_02.pdf

[BPSW99] A. Burns, S. Punnekkat, L. Strigini, D. R. Wright, *Probabilistic Scheduling Guarantees for Fault-Tolerant Real-Time Systems*, 1999, [žiūrėta 2015-05-02], prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=814306>

[Cle09] David Cleden, *Managing Project Uncertainty*, 2009, [žiūrėta 2015-05-02], prieiga per internetą: <https://goo.gl/u7ciHd>

[Con98] Steve McConnell, *Software project survival guide*, Microsoft Press, 1998

[Faz59] Fazar, W., *Program Evaluation and Review Technique*, The American Statistician, Vol. 13, No. 2, (April 1959), p.10, [žiūrėta 2014-12-05], prieiga per internetą: <http://eric.ed.gov/?id=ED015533>

[FLY+11] Jing-chun Feng, Lei Li, Nan Yang, Yu-zhen Hong, Min Pang, Xiong Yao, Li-cheng Wang, *Critical chain construction with multi-resource constraints based on portfolio technology in South-to-North Water Diversion Project*, 2011, [žiūrėta 2016-05-01], prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S1674237015301575>

[FSS13] De Fatima Segger Macri Russo, Sbragia R., Sin Oih Yu A., *Determining Factors in the Unforeseeable Uncertainty Management in Innovation Projects* 2013, [žiūrėta 2014-10-12], prieiga per internetą: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6641762&queryText%3DDetermining+Factors+in+the+Unforeseeable+Uncertainty>

[Gre10] Daniel R. Greening, *Enterprise Scrum: Scaling Scrum to the Executive Level*, 2010, prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5428541>

[**Gol97**] Goldratt, *Critical Chain*, [žiūrėta 2015-06-07], prieiga per internetą:
http://www.goldratt.co.uk/resources/critical_chain/

[**GSF98**] Goodrich M.A., Stirling W.C., Frost R.L., *A Theory of Satisficing Decisions and Control*, 1998, [žiūrėta 2014-12-06], prieiga per internetą:
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=725348&queryText%3Dsatisficing>

[**Hal12**] Nicholas G. Hall, *Project Management: Recent Developments and Research Opportunities* 2012, [žiūrėta 2014-09-25], prieiga per internetą:
<http://link.springer.com/content/pdf/10.1007%2Fs11518-012-5190-5.pdf>

[**HDK12**] Philip M. Huang, Ann G. Darrin, Andrew A. Knuth, *Agile Hardware and Software System Engineering for Innovation*, 2012, [žiūrėta 2014-12-10], prieiga per internetą:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6187425>

[**HL02**] Willy Herroelen, Roel Leus, *Project Scheduling Under Uncertainty Survey and Research Potentials*, 2002 Research Report 0225, [žiūrėta 2015-03-20], prieiga per internetą:
<http://www.sciencedirect.com/science/article/pii/S0377221704002401>

[**HR01**] Willy Herroelen, Roel Leus, On the Merits and Pitfalls of Critical Chain Scheduling, [žiūrėta 2016-05-08], prieiga per internetą:
<http://www.sciencedirect.com/science/article/pii/S0272696301000547>

[**JJ10**] Ma Jun-jie, Qi Jian-xun, *Study on Critical Path Method with Fixed Time Parameter in Network Planning Technology*, 2010, [žiūrėta 2014-12-10], prieiga per internetą:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5945077>

[**KMI15**] Eetu Kupiainen, Mika V. Mäntylä, Juha Itkonen, *Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies*, 2015, Information and Software Technology, volume 62, prieiga per internetą:
<http://www.sciencedirect.com/science/article/pii/S095058491500035X>

[**LAN11**] R. Leus, C. Artigues, F. Talla Nobibon, *Robust optimization for resource-constrained project scheduling with uncertain activity durations*, 2011, [žiūrėta 2015-04-26], prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6117887>

[**LL09**] Sylvain Lenfle, Christoph Loch, *Lost Roots : How Project Management Settled on the Phased Approach (and compromised its ability to lead change in modern enterprises)*, 2009, [žiūrėta 2015-05-16], prieiga per internetą:
<http://www.insead.edu/facultyresearch/research/doc.cfm?did=42948>

[**LRS05**] Thomas G. Lechler, Boaz Ronen, Edward A. Stohr, *Critical Chain: A New Project Management Paradigm or Old Wine in New Bottles*, 2005, Engineering Management Journal, Vol.14 No.4, [žiūrėta 2016-03-10], prieiga per internetą: <http://goo.gl/GYXVtU>

[Mic08] Michael E. Kuhl, *A Dynamic Crashing method for Project Management using Simulation-Based Optimization*, Proceedings of the 2008 Winter Simulation Conference, [žiūrėta 2014-12-06], prieiga per internetą: <http://dx.doi.org/10.1109/WSC.2008.4736344>

[MSM13] Marcelo Marinho, Suzana Sampaio, Hermano Moura, *An Approach Related to Uncertainty in Software Projects*, 2013, [žiūrėta 2015-05-02], prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6721910&tag=1>

[SG09] Qian Shi, Ting Gong, *An improved project buffer sizing approach to critical chain management under resources constraints and fuzzy uncertainty*, 2009 International Conference on Artificial Intelligence and Computational Intelligence, [žiūrėta 2014-10-21], prieiga per internetą: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5376276>

[Sha14] Iris Sharfer, *Time effort estimation method for reducing uncertainty in release of SW product*, 2014, prieiga per internetą: <http://goo.gl/ciowkZ>

[Ste01] Herman Steyn, *An investigation into the fundamentals of critical chain project scheduling*, 2001, [žiūrėta 2015-05-10], prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S0263786300000260>

[VDH08] Stijn Van de Vonder, Erik Demeulemeester, Willy Herroelen, *Proactive heuristic procedures for robust project scheduling: An experimental analysis*, European Journal of Operational Research 189, 2008, [žiūrėta 2015-04-26], prieiga per internetą: <http://www.sciencedirect.com/science/article/pii/S0377221706011805#>

[VDH04] Stijn Van de Vonder, Erik Demeulemeester and Willy Herroelen, *An investigation of efficient and effective predictive-reactive project scheduling procedures*, 2004, [žiūrėta 2015-04-26], prieiga per internetą: https://lirias.kuleuven.be/bitstream/123456789/122885/1/OR_0466.pdf