

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Dovilė Jociūtė

UŽSIĖMIMŲ TVARKARAŠČIŲ PROJEKTAVIMO
SISTEMA

Magistro darbas

Darbo vadovas
doc. dr.
D.Rubliauskas

KAUNAS, 2005

1.	IVADAS	4
1.1.	Projekto tikslas ir adresatas.....	4
1.2.	Informacija apie užsakovo organizaciją	5
2.	ANALITINĖ DALIS	5
2.1.	UTPS analogai	5
2.2.	UTPS analogai užsienyje	6
2.3.	UTPS analogai naudojami Lietuvoje	8
2.4.	UTPS analogai sukurti Lietuvoje.....	10
2.5.	Produkto (UTPS) apibūdinimas.....	13
2.6.	Projektavimo metodologijos ir technologijų analizė	14
2.6.1.	Informacijos sistemų raida ir evoliucija.....	14
2.6.2.	IS struktūra, elementai ir jų tarpusavio ryšiai	15
2.6.3.	IS gyvavimo ciklas ir procesai.....	18
2.6.4.	Iniciavimo fazė	19
2.6.5.	Sistemos apibrėžimas ir analizė.....	20
2.6.6.	Sistemos projektavimas	21
2.6.7.	Sistemos kūrimas	23
2.6.8.	Sistemos diegimas.....	24
2.6.9.	Naudojimo ir palaikymo fazė	26
2.6.10.	IS gyvavimo ciklo fazių tarpusavio ryšiai	27
3.	PROJEKTINĖ DALIS.....	32
3.1.	Bendra terminologija ir paaiškinimai	32
3.2.	Projekto aprašymas ir pritaikymas vartotojo problemai	34
3.3.	Projekto kūrimo tikslai.....	34
3.4.	Klientas, pirkėjas ir suinteresuoti asmenys.....	35
3.5.	Projekto apribojimai	36
3.6.	Sutartinė nomenklatūra ir srities terminai.....	37
3.7.	Darbo apimtis.....	37
3.8.	Produkto funkcijos (Pav. Nr. 4).....	38
3.9.	Funkciniai ir duomenų reikalavimai	38
3.9.1.	Funkciniai reikalavimai	38
3.9.2.	Sąsajos ir valdymo reikalavimai	40
3.9.3.	Naudojamumo reikalavimai.....	40
3.9.4.	Našumo reikalavimai	41
3.9.5.	Darbo aplinkos reikalavimai	42
3.9.6.	Palaikymo reikalavimai	42
3.9.7.	Saugumo reikalavimai	43
3.9.8.	Legalumo reikalavimai	43
3.9.9.	Sukeliamos problemos.....	43
3.9.10.	Užduotys	44
3.9.11.	Rizika	45
3.9.12.	Vartotojo dokumentacija ir apmokymas.....	46
3.10.	Detalusis projekto planas	46
3.10.1.	Projekto aprašymas	46
3.10.2.	Projekto dalyviai	47
3.10.3.	Projekto komanda	47
3.10.4.	Standartai	47
3.10.5.	Projekto kūrimo ciklas	47
3.10.6.	Projekto tarpiniai rezultatai.....	48

3.10.7.	Projekto galutinės pateiktys	48
3.10.8.	Projekto stebėjimas	48
3.10.9.	Rizikų valdymas	49
3.10.10.	Reikalavimų ir pakeitimų valdymas	49
3.10.11.	Projekto peržiūros	49
3.10.12.	Kritiniai kompiuterių resursai	50
4.	VARTOTOJO DOKUMENTACIJA	50
4.1.	Sistemos funkcinis aprašymas	50
4.1.1.	Sistemos paskirtis	50
4.1.2.	Sistemos galimybės	50
4.2.	Sistemos vadovas	50
4.2.1.	Sistemos funkcijos	50
4.2.2.	Sistemos klaidos ir jų šalinimas	54
4.3.	Sistemos instaliavimo dokumentas	55
4.3.1.	Sistemos instaliavimo vadovas Windows aplinkoje	55
4.3.2.	Sistemos sandara	55
4.3.3.	Minimalūs sistemos reikalavimai	56
4.4.	Sistemos administratoriaus vadovas	56
4.4.1.	Sistemos veikimas	56
4.4.2.	Sistemos palaikymo vadovas	57
5.	PRODUKTO KOKYBĖS ĮVERTINIMAS	57
5.1.	Sistemos patikros rezultatai	57
5.2.	Sistemos beta testavimo rezultatai	57
5.3.	Sistemos patikimumo bandymo rezultatai	57
6.	IŠVADOS	58
7.	LITERATŪRA	60

1. ĮVADAS

Šiuolaikinė visuomenė vis spartyn žengia tobulėjimo keliu. Kompiuterinės technologijos neišvengiamai skverbiasi į pačias įvairiausias mūsų gyvenimo sritis. Kasdieninėje žmogaus buityje ir veikloje atsiranda vis daugiau naujų technologijų, lengvinančių darbą: tiek fizinį, tiek protinį. Tokiu būdu produktyvesniu bei pranašesniu tapo ir pedagogo darbas. Kiekvienas mokytojas žino, kaip sunku kasmet sudaryti naują tvarkaraštį: jį taisyti ir pertvarkyti. Dar visai neseniai tvarkaraščių sudarymui švietimo institucijose buvo švaistoma daugybė mūsų brangaus laiko. O ir šiandiena rastume nemažai tokių įstaigų, kuriose didesnę darbo dalį atlieka ranka ir tušinukas, o ne kompiuteris. Kyla klausimas, kodėl žmonės vis dar nelinkę naudotis naujųjų amžių mums teikiamomis lengvatomis? Atsakymų variantų galėtų būti keletas: kuriamos programos yra per daug sudėtingos eiliniam darbuotojui; jau sukurtos programos nepateisina vartotojų lūkesčių; programos nėra adaptuotos pagal vartotojų poreikius.

Nors tvarkaraščių projektavimo sistema nėra naujas dalykas informatikos srityje, vis dėlto čia yra vietos tobulėjimui. Tvarkaraščiai turi būti projektuojami atsižvelgiant į daugelį kriterijų – tiek bendrų, tiek individualių kiekvienai švietimo įstaigai.

Pradedant projektuoti tvarkaraščių sistemą visų pirma reikia apgalvoti, koki rezultatą norima pasiekti ir pasirinkti projektavimo priemones: kokia kalba bus projektuojama, kaip bus kuriama duomenų bazė bei kokius duomenis ji galės saugoti ir keisti, kokios bus programos galimybės. Siekti rezultato padės sukaupta informacija apie jau egzistuojančias tvarkaraščių programas, jų trūkumus, privalumus, taip pat informacija apie projekto užsakovą, jo poreikius bei reikalavimus.

1.1. Projekto tikslas ir adresatas

Projekto pavadinimas - "Užsiėmimų tvarkaraščių projektavimo sistema" (toliau UTPS). Projekto tikslas - sukurti užsiėmimo tvarkaraščių projektavimo sistemą, pritaikytą Kauno technologijos universiteto informatikos fakulteto poreikiams. Darbo metu bus galima naudotis patirtimi sukaupta mokantis Kauno technologijos universitete ir konsultacijomis su šio universiteto dėstytojais.

Projekto užsakovas ir vadovas Praktinės informatikos katedros vedėjas doc. dr. Dalius Rubliauskas.

1.2. Informacija apie užsakovo organizaciją

Kauno technologijos universitetas (toliau KTU) šiandien yra viena didžiausių ir iškiliausių šalies aukštųjų mokyklų. Tai didžiausias techniškas universitetas Baltijos šalyse.

Universitete yra šios savivaldos ir valdymo institucijos: Senatas, Universiteto taryba (visuomeninės priežiūros ir globos institucija), rektorius (jo patariamoji institucija – rektoratas), profesinės etikos kolegija, studentų atstovybė. Universitetą sudaro 13 fakultetų, 10 institutų, 4 centrai, biblioteka bei ūkio ir administracijos padaliniai.

KTU fakultetuose ir pavaldžiuose institutuose stacionariu, vakariniu ir neakivaizdiniu būdu studijuoja apie 17 tūkstančių studentų ir 450 doktorantų. Universitete rengiami 32 studijų programų bakalaurai ir 41 studijų programos magistrai. Specialiosiose profesinėse studijose rengiami 36 kvalifikacijų diplomuoti specialistai. Universitete rengiami 17 mokslo krypties daktarai.

Praktinės informatikos katedra įkurta 1990 m. reorganizavus Matematinės įrangos katedrą į Programinės įrangos ir Praktinės informatikos katedras. Praktinės informatikos katedrai vadovauja doc. dr. Dalius Rubliauskas.

2. ANALITINĖ DALIS

2.1. UTPS analogai

UTPS analogiškos programos - tai programos skirtos sudaryti tvarkaraščius. Pagal funkcionalumą programos galima suskirstyti į 3 grupes – automatinės, pusiau automatinės ir pagalbinės.

Automatinės programos atlieką visą tvarkaraščių sudarymo darbą pačios. Vartotojas dalyvauja tik įvesdamas duomenis ir nurodydamas tinkamumo parametrus.

Pusiau automatinės programos sugeneruotą grafiką duoda pataisyti vartotojui. Pataisytame grafike gali būti patikrinamos klaidos pagal pateiktus parametrus.

Pagalbinės programos tik suteikia vartotojui sąsają kurioje grafiką sudaryti paprasčiau negu tam neskirtose programose. Programa pagal pateiktus parametrus gali sukurti pamokų, dėstytojų bei kabinetų įrankinę, taip pat gali turėti patikrinimo nuo klaidų (konfliktų) galimybę.

Grafikų sudarymo programos gali būti bendro naudojimo arba specializuotos. Specializuotos programos gali būti pritaikytos universitetų, mokyklų arba vakarinių apmokymų grafikams

kurti. Kadangi mokyklų ir universitetų poreikiai labai panašūs prieš kurdami UTPS išanalizuosime ir mokyklų ir universitetų tvarkaraščių generavimo programas.

2.2. UTPS analogai užsienyje

Untis : Grafikų kūrimo programa

http://www.saiposa.ch/gp-untis_timetable_1e.htm

Untis grafikų kūrimo programa - Europos lyderis tarp grafikų kūrimo programų. Untis yra labai lanksti ir gali būti pritaikyta bet kurio tipo mokymo įstaigai: pradinei mokyklai, vidurinei mokyklai, aukštajai mokyklai, profesinei mokyklai ar universitetui. Untis programa padės padalinti mokymo pareigas tarp dėstytojų. Su Untis galima redaguoti savo grafikus per internetą ir daugeliu kitų įvairių būdų.

FET : Nemokama evoliucinė grafikų kūrimo programa

<http://lalescu.ro/liviu/fet/>

FET yra nemokama programa pagal GNU/GPL licenciją. Ji skirta automatiniams mokyklos arba universiteto grafikų sudarymui.

Infosilem : programiniai sprendimai akademiniam grafikams

<http://www.infosilem.com>

Infosilem kuria programinius sprendimus akademiniam grafikam sudaryti. Jų produktų grupė padeda sukurti pagrindinius mokymo ir egzaminų grafikus. Šiame sprendime yra įrankiai prisitaikyti netgi prie individualių studentų poreikių.

ABC Timetable

<http://www.abc-ware.com/abctime.htm>

Nemokamas sprendimas palengvinti grafikų sudarymą ir spausdinimą. Reikia tiesiog pasirinkti dienų skaičių, įvesti užsiėmimų laikus bei pavadinimus, pasirinkti dizainą, spalvas, šriftus. Viskas lengvai išsaugoma ir išspausdinama kada tik nori. ABC grafikų sudarymo sprendimas yra nemokamas.

EDT2004 : interaktyvus grafikų kūrimas

<http://edt2004.free.fr/edt-2004-english.htm>

EDT2004 yra naudingas grafikų projektavimo įrankis. EDT2204 gali dirbti vieno ir daugelio vartotojų režime. Daugelio vartotojų režime EDT2204 sukuria MySQL duomenų bazę. EDT2204 turi grafinę sąsają ir grafikus kuria visiškai automatiškai arba iš dalies automatiškai būdu.

TTABLE

<http://www.ttable.com>

Pilnas grafikų kūrimo sprendimas su tinkinamais ir automatiniais grafikais. TTABLE kuria Significant Software Solutions, Indijoje.

eventMAP Limited

<http://www.evmap.com>

Tai nuolat tobulinama grafikų kūrimo programa, kuriai taip pat siūlomi apmokymai ir palaikymas Tarp siūlomų produktų yra ir integruojami apmokymo užsiėmimų, egzaminų, dėstytojų bei patalpų planavimo įrankiai. Jie leidžia efektyviai išnaudoti esamą laiką bei kitus resursus.

Kat Timetabler

<http://www.scisoft-kat.com/>

Kat timetabler yra kuriamas kaip bendras kursų planavimo bet kurioje mokykloje ar kolegijoje įrankis. Jis turi protingą optimizavimo varikliuką kas kartu su jo planavimo galimybėmis daro jį vienu iš galingiausių grafikų sudarymo įrankių.

First Class – grafikų sudarymo programa

<http://www.humanedge.biz/firstclass/>

First Class 2000 praves per kiekvieną grafiko sudarymo žingsnį, nuo paskaitos pavadinimo įvedimo iki sudarytos lentelės atspausdinimo. Programa taip pat turi grafikų internete skelbimo galimybę.

Timetable programa, kuriama Lalescu Liviu

<http://algorithms.ro/>

Programa generuoja grafikų lenteles pasinaudodama genetiniu algoritmu. Programos kodas kuriamas pagal GNU/GPL (nemokamos programos, su atviru kodu) licenciją. Programa veikia tik Linux operacinėje sistemoje.

Didakt

<http://www.algori.se/english/home.html>

Didakt yra viena iš Europos labiausiai išsivystytų grafikų sudarymo programų, tačiau ją naudoti labai lengva. Joje daug naudingų galimybių, pvz.: Automatinis planavimas ir objektų pertempimas pele, kuri labai pagreitina rankinių grafikų kūrimą.

iMagic Timetable Master

<http://www.imagicsoftware.biz/iMagicTimetable.asp>

iMagic grafikų meistras greitai ir lengvai kuria lenteles mokykloms, koledžams ir universitetams. Automatiškai sukuria grafikus klasėms bei auditorijoms. iMagic grafikų meistras automatiškai aptinka konfliktus ir juos išsprendžia.

2.3. UTPS analogai naudojami Lietuvoje

ASC tvarkaraščiai

Pasinaudodami šia programa galėsite greitai ir paprastai sukurti savo mokyklos, gimnazijos, profesinės mokyklos, licėjaus, aukštosios mokyklos ar kitos mokslo įstaigos tvarkaraščius, kurie atitiks visus organizacinius, psichologinius bei pedagoginius reikalavimus

Jau daugiau nei 7500 mokymo įstaigų visame pasaulyje sėkmingai naudoja šią programą

Tvarkaraščių programoje labai lengva įvesti visus reikalingus duomenis: disciplinas, klases, mokytojų darbo laiką, pamokų tarpusavio ryšius, kabinetus. Galima klases suskaidyti į grupes, sujungti kelias klases bendram užsiėmimui, paskirti kelis mokytojus vienai pamokai, įvesti pamokas, kurios vyksta kas antra savaitė ir t.t.

Galima pasirinkti tikslų laiką arba laiko intervalą, kada disciplina turi vykti, kokiame kabinete ir kuris mokytojas pamoką turi vesti. Pavyzdžiui, galima nurodyti, kad matematikos pamokos negali vykti vėliau nei 6-ji pamoka, kuris nors mokytojas negali vesti pamokų antradieniais, sporto salė negali būti užimta penktadieniais po 5-os pamokos, nes tuo laiku ateina treniruotis mokyklos krepšinio komanda ir pan.

Per keliolika minučių programa patikrins tūkstančius variantų ir pati sudarys tvarkaraštį, atitinkantį užduotas sąlygas ir apribojimus:

- maksimalų langų skaičių mokytojams, dienas, kada mokytojai ar dėstytojai gali vesti užsiėmimus;

- tolygų disciplinų išdėstymą per savaitę;
- patikrins pamokų nuoseklumą pilnomis ir padalintoms klasėms;
- pamokoms paskirs kabinetus;
- patikrins visas kitas užduotas sąlygas...

Programa patikrins duomenis ir padės surasti bei pašalinti galimas klaidas. Ji taip pat patikrins, ar tvarkaraštis atitinka visas užduotas sąlygas.

Programa pateiks tvarkaraščio statistinę informaciją.

Galima atspausdinti savo tvarkaraštį. Programa automatiškai sukurs tvarkaraščius atskiroms klasėms, mokytojams bei kabinetams, taip pat sukurs bendrus tvarkaraščius klasėms, mokytojams ir kabinetams. Galima parinkti pavaduojančius mokytojus, pasirinkti, kuriems mokytojams, klasėms ar profilinių klasių mokiniams norėsite atspausdinti tvarkaraščius, atspausdinti neužimtų kabinetų sąrašus. Taip pat galima eksportuoti duomenis ir pačius tvarkaraščius į Microsoft Excel programą, html formatą ir patalpinti tvarkaraščius internete. Pavadavimų programa leidžia parinkti pavaduojančius mokytojus, kada paskirtieji mokytojai negali vesti pamokų, pavyzdžiui, dėl ligos. Programa sudaro suvestines apie nedalyvavusius mokytojus pasirinktam laikotarpiui, dėl kokių priežasčių jie nevedė pamokų. Programos vartotojo sąsaja yra paprasta ir lengvai suprantama. Pavadavimų programa yra tvarkaraščių programos sudėtinė dalis.

Mimosa

<http://www.mimosasoftware.com/>

Programos pagrindinė idėja yra labai paprasta. Praktiškai neįmanoma automatiškai sukurti gero tvarkaraščio. Todėl programa to padaryti ir nesistengia - ji tik sukuria aplinką maksimaliai patogiai sudėlioti ir derinti tvarkaraštį, pateikdama efektingą ir reikalingą informaciją.

Programa nereikalauja nei ypatingų kompiuterių, nei ypatingų dirbančiojo žinių. Ji turėtų veikti bet kuriame kompiuteryje, kuriame veikia Windows95 operacinė sistema. Programos vartotojui pakanka turėti pradinį kompiuterinį raštingumą ir, žinoma, žinoti, kaip sudaromas pamokų tvarkaraštis.

Programos galimybės - tikrai didelės. Programa leidžia:

- klases skirstyti į kiek norima grupių (ribiniu atveju grupių bus tiek, kiek klasėje yra mokinių);
- sujungti kelių klasių ar jų grupių mokinius per atskirų dalykų pamokas;

- tos pačios klasės mokiniams vienu metu sudėti kelias skirtingas pamokas (pavyzdžiui, kelias skirtingas kalbas ar menus) ar priskirti kelis mokytojus dėstyti tą patį dalyką (pavyzdžiui, kai vieną dalyką pasirenka daugiau nei 30 mokinių);
 - sudarinėti atskirus pamokų tvarkaraščius kiekvienam mokiniui (pavyzdžiui, vienuoliktokams);
 - uždėti norimus apribojimus (laisvas pamokas ar dienas) mokytojams ir mokiniams.
 - nurodyti užsiėmimų svarbą (privalomas ar pasirenkamas dalykas, maža ar didelė mokinių grupė jį pasirinko);
 - nurodyti automatiškai sudėlioti visas ar dalį pamokų arba sudėti jas po vieną rankiniu būdu;
 - atspausdinti individualius pamokų tvarkaraščius mokytojams, mokiniams ar klasėms;
 - atspausdinti kabinetų užimtumo tvarkaraščius;
- padaryti internetinę tvarkaraščių versiją.

2.4. UTPS analogai sukurti Lietuvoje

Tvarkaraščio sudarymas realizuotoms mokykloms

<http://www.soften.ktu.lt/~mockus/german/tvarkb.html>

Programą 2001.06.07 sukūrė:

IFM6/1 gr.stud. Marius Černiauskas,

IFM6/2 gr.stud. Ula Leonavičiūtė,

IFM6/2 gr.stud. Mindaugas Kuprys.

Programa sukurta naudojant Java kalbą. Programa įdėta serveryje, tai reiškia, jog ji prieinama kiekvienam vartotojui, turinčiam internetinį ryšį. Programos sąsaja aiški ir neapkrauta. Vartotojas gali aiškiai matyti, kokie pasirinkimai galima vienu ar kitu atveju. Programa visą darbą atlieka pati be vartotojo įsikišimo, jam tereikia pasirinkti baudos taškus vienam ar kitam punktui tokiems kaip: "langai mokiniui", "langai mokytojui" ir kt. Taip pat vartotojas gali nurodyti apribojimus, pvz: „dvi besitęsiančios pamokos“, „mokytojų nedarbo dienos“ ir „maksimalus pamokų skaičius per diena“.

Pasirinkus optimizavimo veiksmą vartotojas turi galimybę pasirinkti optimizavimo iteracijų skaičių ir tikimybę, kad dalykas bus keičiamas vietomis, tai garantuoja geresnį optimizavimo proceso veikimą, tačiau tai prailgina šio proceso trukmę.

Kaip būtina sąlyga reikia pažymėti jog vartotojas turi atsiųsti savo duomenis "failo_vardas.txt" formate ir atlikęs tvarkaraščio sudarymo veiksmus juos gali atsisųsti. Programa neleidžia atlikti rankinio dalykų koregavimo per interneto naršyklę, todėl visus rankinius sudaryto tvarkaraščio pakeitimus vartotojai turės atlikti savo kompiuteryje.

Programa galima naudotis laisvai. Tačiau visi veiksmai susiję su kopijavimu tūrētu būti suderinti su autoriais.

Tvarkaraščių kūrėjas

<http://www.ibn.lt>

Tvarkaraščių sudarymo programa lietuvių kalba (profesinėms, vidurinėms mokykloms, gimnazijoms ir kitoms mokymo įstaigoms)

Programa sugeba sukurti visos mokyklos tvarkaraštį, į jį sudėdama visas reikiamas pamokas, jei tik tai yra menkiausia galimybė, patikrindama tūkstančius įvairiausių galimų variantų. Net jeigu pasirodo, kad sudarius tvarkaraštį, jame kas nors turi būti pakeista, su šia programa per keletą minučių galima pakoreguoti tvarkaraštį ar sudaryti naują. Programa tinka ir mokykloms su profiliniu mokymu.

Programa leidžia tokius dalykus:

Įvesti ir išsaugoti pagrindinius duomenis: klases, mokytojus, kabinetus, pamokas, savaitės dienas kada vyksta pamokos ar užsiėmimai, skambučių laiką, kopijuoti vienos klasės pamokas kitai ir kt.

- Aprašyti įvairias sąlygas: laiką, kada gali vykti pamokos, mokytojų darbo laiką, sąryšius tarp skirtingų disciplinų (kaip jos turi būti paskirstytos per savaitę, kokios disciplinos negali eiti viena po kitos, ar jos reikalauja pasiruošimo namie...), langų mokytojams laiką, pietų pertraukas, dėstomų pamokų skaičių ir daugelį kitų dalykų.

- Sukurti tvarkaraštį: programa turi galingą algoritmą, leidžiantį jai sukurti tvarkaraštį per keletą minučių, sudedant į tvarkaraštį visas pamokas ir užsiėmimus. Vėliau galima tvarkaraštį redaguoti, tobulinti, keičiant sąlygas ar net rankiniu būdu išdėstant pamokas. Jeigu pasirodo, kad tvarkaraščio sukurti neįmanoma, programa turi testavimo įrankius, leidžiančius surasti problemą, ar tai būtų įvestų duomenų klaida, ar pernelyg griežtos jai užduotos sąlygos. Programa tinka ir profilinių klasių tvarkaraščiams sudaryti.

- Atnaujinti tvarkaraštį per visus mokslo metus, atspausdinti kiekvienos klasės, kabineto ar mokytojo tvarkaraštį, ar bendrą mokyklos tvarkaraštį. Taip pat tvarkaraščius galima eksportuoti į Excel lenteles ar html (interneto puslapių) formatą ir skelbti Internetą mokyklos interneto puslapyje ar vietiniame kompiuterių tinkle.

- Įsigydami Tvarkaraščių programą, kartu gausite ir Pavadavimų programą, kuri leis nesunkiai išspręsti situacijas, kai reikia parinkti pavaduojančius mokytojus.

- Programa turi ir kitas kalbas rusų, lenkų, anglų.

- Programa dirba Windows aplinkoje (Windows 95, 98, 2000, ME, XP).

Pamokos

<http://avj.w3.lt/>

Programos PAMOKOS galimybės:

- Dvi pamainos viename tvarkaraštyje;
- klasę galima skirstyti į 2-5 grupes;
- galima sujungti kiek norima klasių, ir tada skirstyti į 1-5 grupes;
- laisvų dienų ar valandų norimą dieną pasirinkimas mokytojams;
- programinis „pamokų paskirstymo“ lentelės suformavimas;
- programinis pamokos formavimas iš „pamokų paskirstymo“ lentelės;
- programinis „likučių“ įrašymas;
- tvarkaraščio koregavimas, leidžiantis sukeisti pamokas vietomis kartu keičiant ir pamokas, kurios trukdo sukeitimui;
- kabinetų skirstymas;
- tvarkaraščio spausdinimas mokytojams, klasėms, mokyklai, kabinetams;
- tvarkaraščio archyvą galima sukaupti iki 15 tvarkaraščio variantų.
- programiškai suformuojamos pamokos pagal moksleivių pasirinkimą fiksuojant kiekvienoje grupėje moksleivių, kurie lankys šį dalyką, kodus (praktiškai dar neišbandyta - programinis modulis padarytas 2000 m. spalio mėn. pagal vienos mokyklos realią moksleivių pasirinkimo lentelę);

- kasdieninis tvarkaraščio koregavimas ir vestų pamokų apskaita (praktiškai nebuvo išbandyta - programa paruošta 1998 m. - šį modulį galima būtų įdiegti, jei mokykloms būtų aktualu).

- užprogramuoti reikalavimai moksleivių tvarkaraščiams, bet palikta galimybė laisvai rinktis:

- tą pačią dieną gali būti ne daugiau kaip 2 tikslieji dalykai (matematika, chemija, fizika, informatika);

- tą pačią dieną gali būti ne daugiau kaip 2 lengvieji dalykai (muzika, darbai, dailė, teatras, choreografija, kūno kultūra);

- dvigubos lietuvių kalbos pamokos vyresnėse klasėse tik antradienį, trečiadienį, ketvirtadienį ne vėliau, kaip ketvirta pamoka;

- dalykai turintys 2 savaitines pamokas dedami kas antra diena;

- po kūno kultūros nepageidaujama daryti matematikos;

pamokos, kuriose dalyvauja dalis klasės, dedamos nuo 5 pamokos ir vėliau.

2.5. Produkto (UTPS) apibūdinimas

Programų sistemos funkcijos

Programa turi turėti galimybę įvesti pradinis duomenis, sugeneruoti tarpinius rezultatus ir leisti vartotojui pataisyti tarpinius rezultatus taip gaunant galutinį rezultatą.

Sistemos kontekstas

Programa bus kuriama programavimo kalba Visual Basic, todėl ji turi veikti Windows operacinės sistemos aplinkoje. Programa bendraus su Microsoft Exel programa naudodamasi OLE sąsaja, todėl kompiuteryje jau turi būti instaliuotas Microsoft Office paketas su Microsoft Exel programa.

Vartotojo charakteristikos

Tipiškas UTPS vartotojas, tai su informacinėmis sistemomis susipažinęs dėstytojas arba administratorius. UTPS vartotojas turi būti susipažinęs su UTPS vartotojo instrukcija.

Vartotojo problemos

UTPS turi padėti vartotojui išspręsti tvarkaraščių sudarymo problemą. UTPS vartotojas turi žinoti tvarkaraščio dedamuosius duomenis. t.y. studentų grupes ir jų kieki, paskaitų

pavadinimus ir jų kiekį, auditorijų kiekį, auditorijų tipus, darbo vietų auditorijose kiekį, kokios auditorijos tinka kokiems užsiėmimams bei kurios studentų grupės lanko kokius užsiėmimus. Taigi vartotojas turi ne tik turėti pradinis duomenis, bet ir žinoti ryšius tarp duomenų.

Vartotojo tikslai

Vartotojo tikslas- -naudojant UTPS, t.y. įdedant kuo mažiau pastangų, sukurti tvarkaraštį atitinkantį ir studentų ir dėstytojų poreikius. UTPS turi padėti vartotojui pasiekti tikslą, panaudodama pradinis duomenis sugeneruoti optimalų tvarkaraštį ir leisdama patogiai bei greitai parengti savo pateiktą rezultatą.

Bendri apribojimai

UTPS turėtų būti pritaikyta dirbti su informacinėmis sistemomis susipažinusiems vartotojams, todėl ją projektuojant tikimasi, kad vartotojai įves tik logiškai teisingus ir atitinkančius įvedimo laukus duomenis.

2.6. Projektavimo metodologijos ir technologijų analizė

2.6.1. Informacijos sistemų raida ir evoliucija

Kompiuterizuotos IS įvairiomis formomis kuriamos jau daugiau nei 40 metų - nuo paprastų taikomųjų programų, kurių duomenims apdoroti, konkrečiam procesui ar funkcijai palaikyti, iki šiuolaikinių didelės apimties kompanijų resursų planavimo sistemų.

Kuriant pirmąsias IS, jų projektuotojai ir kūrėjai paprastai būdavo programuotojai, ir sistemų projektavimas priklausydavo nuo jų kompetencijos. Šios ankstyvosios sistemos dažniausiai būdavo mokslinės ar karinės. Pradėjus taikyti IS verslo srityje, atsirado poreikis apibrėžti IS vystymo teorines gaires. Ankstyvosios sistemos būdavo prastai dokumentuotos, ir jas sukūrę programuotojai būdavo vieninteliai žinantys, kaip jos sukonstruotos ir kaip veikė. Tai tapdavo dideliu barjeru atsiradus poreikiui tokias sistemas palaikyti, vystyti ir sujungti su kitomis sistemomis. Tobulėjant technologijoms augo galimybės kurti sudėtingesnes ir kompleksines IS, todėl struktūrizuotų darbo metodų poreikis vis labiau augo.

Visos verslo IS turi gyvavimo ciklą, kuris prasideda verslo ir galimybių analize, pereina kūrimo fazę ir baigiasi naudojimu, palaikymu bei likvidavimu. Paprastai pagrindinis dėmesys ir priemonės, kuriant IS, skiriami būtent kūrimo fazei, tai reiškia, kad sistemos veikimas ir palaikymas, kaip rutininis darbas, paliekamas ne sistemų projektavimo specialistams.

IS kūrimo fazė paprastai matoma kaip kompleksinis uždavinys, reikalaujantis daug darbo ir metodologinių priemonių. Sistemų kūrimo gyvavimo ciklas dažnai vadinamas *krioklio modeliu*. Tai seniausias ir didžiausias tradicijas turintis modelis. „Krioklio“ požiūrio filosofija kildinama iš kitų inžinerijos procesų ir išskaido sistemų kūrimą į atskirus žingsnius, siekiant supaprastinti visumą. Skirtingais atvejais šių žingsnių kiekis ir pavadinimai skiriasi, tačiau paprastai į juos įeina: galimybių tyrimas, sistemų analizė, sistemų kūrimas, diegimas, įvertinimas ir kartais operavimas bei palaikymas. Toks požiūris ir iš jo kildinami variantai, paprastai turintys iteracijos sąlygų, istoriškai tapo pagrindu kuriant daugybę IS kūrimo metodologijų. Šios metodologijos gali būti labai tinkamos programavimo ir programinės įrangos inžinerijai, tačiau informacijos sistemų kūrimas nėra vien tik inžinerijos, o veikia multidisciplininė problema, apimanti įvairiausias sritis - nuo socialinių mokslų ir verslo administravimo iki inžinerijos. Jei šio fakto, kuriant IS, nepaisoma, išauga tikimybė susidurti su problemomis. Kaip vien tik inžinierinės paradigmos laikymosi rezultatas sukurta daug informacijos sistemų, kurios iš tiesų nepatenkina organizacijos keliamų reikalavimų. Viena klasikinių problemų - nepriimami domėn keleto organizacijos narių (būsimųjų sistemos vartotojų) poreikiai. Neretai koncentruojamasi ties tais procesais, kuriuos lengva automatizuoti, o ne tais, kuriuos reikėtų automatizuoti. Tokios problemos skatino kurti informacijos sistemų teoriją, leisiančią pažvelgti į IS problematiką iš aukštesnio lygio, apimant įvairiausias problemas, ieškant joms sprendimų, kurie išliktų aktualūs bėgant laikui, naujų galimybių bei jų realizavimo metodų.

2.6.2. IS struktūra, elementai ir jų tarpusavio ryšiai

Siekiant geriau suprasti IS sistemų ypatumus ir jų skirtumus, reikia aptarti IS struktūrą, jų elementus, šių elementų tarpusavio ryšius. Veiklos sistemos, kurios dalis yra IS, struktūrinės dalys yra: klientas (kaip VS veiklos rezultatų gavėjas), produktas (kaip VS veiklos rezultatas), verslo procesas (VS veiklos taisyklės), dalyviai, informacija ir technologijos. Atsižvelgiant į IS specifiką ir poziciją veiklos sistemoje (IS yra VS informacinės veiklos posistemė), galima išskirti tokius bendriausius IS elementus:

- Techninė įranga;
- Programinė įranga;
- Duomenys ir informacija;
- Žmonės;

- Taisyklės.

Techninė įranga

Techninė IS įranga – tai fizinė bazė, kurioje realizuoti programiniai IS sprendimai. Šią bazę galima skirti į dvi dalis:

1. kompiuteriai, jų periferinė įranga, kiti elektroniniai įrenginiai, skirti informacijai gauti, kaupti, apdoroti, išvesti;
2. tinklų įranga.

Plačiausiai paplitę yra asmeniniai kompiuteriai. IS naudojami kompiuteriai paprastai nėra niekuo ypatingi, tad plačiau apie juos kalbėti nėra reikalo. Naudojami periferiniai įrenginiai priklauso nuo užduočių, kurias turi atlikti IS. Tai dažniausiai būna su informacijos įvedimu/išvedimu susiję įtaisai, pritaikyti tos informacijos pobūdžiui (pavyzdžiui galima paminėti gana specifinius įrenginius, skirtus akies rainelės ar pirštų atspaudu nuskaitymui). Taip pat IS vis dažniau naudojami tradiciškai tam nenaudoti įrenginiai (pvz., mobilieji telefonai, kišeniniai kompiuteriai), ar kuriami nauji, kurių pirminė paskirtis būna orientuota į naudojimą IS (pavyzdžiui, elektroninėms knygoms skaityti skirti įrenginiai, interneto televizoriai ir pan.). Tokie specifiniai įrenginiai dažniausiai neturi tokios didelės skaičiuojamosios galios, kaip asmeniniai kompiuteriai, tačiau interneto technologijų ypatumai to nereikalauja – dažniausiai pakanka, kad įrenginiai palaikytų interneto protokolų rinkinį TCP/IP ar specialų suderinamą protokolą (pvz., WAP), juose veiktų interneto naršyklė.

Serveriams keliami kur kas didesni patikimumo ir našumo reikalavimai, nes jiems tenkantis krūvis labai didelis. Dažnai serveriuose, kurių veikimo kokybė yra kritiškai svarbi visai IS, taikomas pertekliškumo principas: dubliuojamos sudėtinės serverio dalys, kad vienai sugedus ją pakeistų kita. Taip pat serveriuose diegiama galimybė pakeisti sugedusias dalis nenutraukiant jų darbo. Paprastai serveriuose naudojami specialūs komponentai, kuriems keliami kur kas aukštesni reikalavimai nei darbo stotims.

Kita, kur kas specifiškesnė ir priklausanti nuo interneto technologijų specifikacijų dalis yra tinklų įranga. Kabeliai turi būti suderinami su naudojamos tinklo topologijos reikalavimais, duomenims perduoti gali būti naudojamas radijo ryšys ar šviesolaidis. Faktiškai prie interneto gali būti prijungtas bet kokios topologijos tinklas. Tuo pasirūpina specialūs įrenginiai, atsakingi už tinklų sujungimus, paketų komutaciją ir duomenų perdavimo procesus: maršrutizatoriai, kartotuvai, tiltai, komutatoriai. Kompiuterinius įrenginius su tinklu jungia tinklo plokštės ar įvairūs modemai. Vietiniame tinkle gali būti naudojami specifiniai standartai, tačiau prijungimas prie interneto vyksta naudojant TCP/IP protokolų rinkinį. Visgi

IS ypatumas tas, kad IS atveju dažniausiai vidiniam tinklui naudojami interneto protokolai. Tai leidžia žymiai supaprastinti ir paspartinti tinklų tarpusavio sujungimą ir prijungimą prie interneto.

Susijungimui su interneto paslaugų teikėju (ISP – *Internet Services Provider*) gali būti naudojami įvairūs fiziniai kanalai bei protokolai, pvz., DSL (Digital Subscriber Line), kabelinės televizijos kanalai, satelitinis ryšys, netgi susijungimas elektros tinklais.

Programinė įranga

Programinė įranga yra bene svarbiausia IS dalis. Ji, remdamasi IS taisyklėmis, koordinuoja žmonių darbą su technine įranga, atliekant manipuliacijas su informacija, apdoroja įvestus duomenis ir t.t. Visas programinės įrangos galimybes ir jos pasiūlą išties sudėtinga aprėpti, nes siūloma labai daug komercinių ir atvirojo kodo produktų, turinčių labai plačias ar labai specializuotas galimybes.

Duomenys ir informacija

IS globalus pobūdis lemia tai, kad ženkliai išauga duomenų ir informacijos, kurią turi apdoroti IS, kiekiai ir turinys. Jeigu senesnių IS atveju sistemos operuodavo pakankamai specifiniais duomenimis (apskaitos, vadybos, buhalteriniais ir pan.), tai dabar IS surenkamų, apdorojamų, saugomų ir perduodamų duomenų kiekiai yra labai dideli ir šie duomenys gali būti labai skirtingi. Pavyzdžiui, organizacijos IS gali kaupti įėjimo kontrolės, projektų valdymo, žmoniškųjų resursų planavimo, sandėlio, klientų veiksmų organizacijos interneto svetainėje ir jos lankomumo ir pan. duomenis. Kiekviena globalios sistemos posistemė gali generuoti skirtingo intensyvumo duomenų srautą; turint omenyje IS būdingą glaudžią integraciją su organizacijos veiklos sistema tokių posistemių, priklausomai nuo organizacijos dydžio ir kompleksiskumo, skaičius gali varijuoti; bendras IS gaunamų ir apdorojamų duomenų ir informacijos srautas bus lygus posistemių generuojamų srautų sumai. Sparčiai tobulėjančios ir pingančios techninės ir programinės informacijos kaupimo bei apdorojimo priemonės (diskiniai kaupikliai, duomenų archyvavimo įrenginiai, DBVS PĮ) leidžia kaupti net tuos duomenis, kurių esamu momentu nereikia, ir atsiradus poreikiui atlikti retrospektyvią jų analizę. Greta duomenų, kurie yra IS objektas, egzistuoja ir tarnybiniai sistemos duomenys, tokie kaip prisijungimų prie sistemos žurnalai, vartotojų autorizacijos duomenys ir t.t. Duomenys ir informacija, kuriuos sistemos vartotojai gauna iš sistemos, priklauso nuo jų turimų teisių sistemoje.

Žmonės

Išskiriamos tokios žmonių grupės:

- Vartotojai;
- aptarnaujantis personalas.

Vartotojai yra svarbi sudedamoji IS dalis. Kuriant IS reikia atsižvelgti į jų poreikius, bei galimybes. Aptarnaujantis personalas prižiūri bei konfiguruoja IS sistemas, taip pritaikydamas jas vartotojų poreikiams.

Taisyklės

Sistemos vartotojų darbą su sistema nustato taisyklės, reglamentuojančios darbo procedūras. Jos skirtos vartotoją supažindinti su darbo sistemoje eiga, padeda išmokti efektyviau atlikti savo darbą. IS atveju neretai tokias taisykles pateikti vartotojui sudėtinga, arba egzistuoja didelė tikimybė, kad vartotojas jų neskaitys. Pripratę prie šiuolaikinės programinės įrangos draugiškumo vartotojui ir ergonomiškumo jie greičiausiai iš pradžių bandys pradėti darbą su sistema intuityviai. Kai kuriais atvejais, jei vartotojai turi galimybę rinktis, sistemos draugiškumo vartotojui klausimas yra ypač aktualus (pvz., e-parduotuvės atveju, kai klientas visuomet gali pasirinkti konkurento sistemą). Todėl taisyklės reikėtų stengtis pakeisti apgalvota vartotojo sąsaja ir paprastu bei aiškiu darbo su sistema procesu (angl. *workflow*), kuris pats nukreiptų vartotoją reikiama linkme. Kitas būdas – pateikti pagalbinius tekstus pačioje sistemoje (pagalbos puslapių, kontekstinės pagalbos, laukų paaiškinimų pavidalu), kad vartotojui būtų kuo lengviau perprasti darbo sistemoje taisykles. IS atveju įprastas reiškinys pateikti atgalinio ryšio (*feedback*) galimybę, kuri leistų operatyviai gauti pagalbą. Tam organizacijose kuriamos specialios konsultacijų tarnybos (*help desk*), galinčios pagelbėti iškilus sunkumams; svetainėse matomoje vietoje pateikiami kontaktiniai telefonai, el. pašto adresai, paklausimo formos.

2.6.3. IS gyvavimo ciklas ir procesai

IS, nepriklausomai nuo sudėtingumo ir darbų, būtinų jų įvykdymui, apimties, savo vystymosi eigoje pereina tam tikras būsenas: nuo būsenos, kai sistemos dar nėra, iki būsenos, kai sistemos jau nėra. Sistemos gyvavimo pakopa, turinti aiškiai išreikštus kokybinius ypatumus, vadinama stadija. Laikotarpis, reikalingas vienareikšmiškai nustatomam kokybiniam sistemos pokyčiui realizuoti, vadinamas etapu (arba faze). Išskiriamų IS gyvavimo ciklą etapų kiekis, jų pavadinimai ir turinys įvairiuose šaltiniuose bei situacijose

įvairiu laipsniu skiriasi, nes šios charakteristikos didele dalimi priklauso nuo konkretaus projekto įgyvendinimo sąlygų bei dalyvių patirties. Tačiau logika ir pagrindinis IS vystymo proceso turinys paprastai yra toks pats. Apibendrinant įvairius šaltinius, galima išskirti tokias pagrindines IS gyvavimo ciklo fazes:

- Inicijavimas;
- Sistemos apibrėžimas ir analizė;
- Sistemos projektavimas;
- Kūrimas;
- Diegimas;

Naudojimas ir palaikymas.

2.6.4. Iniciavimo fazė

Iniciavimo fazė (initiation, inception). Kitur dar įvardijama kaip galimybių analizė (*feasibility study*). Šio etapo metu identifikuojama ir atpažįstama problema. Tai gali būti situacija, kai esanti IS nesusidoroja su jai keliamomis užduotimis, kai valdantysis lygmuo suvokia, kad jiems trūksta kažkokios gyvybiškai svarbios informacijos, ir pan. Šis etapas paprastai kyla iš vadybos problemų, kurias ir turi išspręsti bei palaikyti IS.

Projektas neturėtų būti pradėtas, kol nebuvo pakankamai išdiskutuoti jo tikslai, laiko normos, taikytinas požiūris bei priežiūros bei kontrolės viso projekto gyvavimo ciklo metu mechanizmai. Šio etapo rezultatas turėtų būti galimybių ataskaita, aiškiai nusakanti laukiamą naudą, kaštus, planavimo apribojimus, bei resursų gavimo būdus bei kelius. Gali būti identifikuojamos sąsajos ir priklausomybė nuo kitų susijusiose srityse veikiančių projektų. Turi būti suplanuotas žmogiškųjų ir techninių resursų poreikis. Projekto erdvė (organizacijos padalinių apimties ir darbuotojų įtraukimo požiūriu) yra kritiškai svarbi jo sėkmei. Ankstyvas planavimas leidžia identifiuoti situacijas, kai labiau verta įsigyti/panaudoti trečiųjų šalių sukurtą programinę įrangą, ar pasamdyti darbuotojus, turinčius reikalingų projektui atlikti įgūdžių.

Skaidant šią fazę kaip galimybių analizės procesą, galima išskirti tokius poetapius:

- Techninių galimybių analizė;
- Ekonominių galimybių analizė;
- Teisėtumo galimybių analizė;

- Operacinių galimybių analizė (ar tiks esamu momentu naudojamos organizacinės procedūros naudoti naujos IS aplinkoje?);
- Laiko planavimo galimybės (kiek užtruks projektas, ar šis projektas gali būti įvykdytas laiku, tada, kai jo reikia).

Tai, kad IS atveju verslo sąlygos pritaikomos interneto informacijos sistemoms, o ne sistemos prie verslo sąlygų, kaip būdinga tradicinėms IS, įtakoja ypatingą trečiųjų šalių sukurtų sprendimų analizės svarbą. Svarbu sekti naujausias IS raidos tendencijas, kad galima būtų laiku jas panaudoti savo organizacijoje. Todėl gana dažnas reiškinys, kai naujų IS galimybių atsiradimas gimdo poreikį kurti/pertvarkyti turimą sistemą, o ne atvirkščiai.

2.6.5. Sistemos apibrėžimas ir analizė

Sistemos apibrėžimo ir analizės (*system definition and analysis*) fazės metu sudaromas bendras būsimos sistemos vaizdas. Tai išreiškiama nurodant, kokios reikės informacijos, kokių duomenų apdorojimui, kas turės vykti apdorojimo metu. Šiame etape naudingos tokios priemonės, kaip duomenų srautų diagramos, kurios bus ypatingai naudingos vėlesniuose sistemos kūrimo etapuose.

Šio etapo metu identifikuojami vartotojų poreikiai, problemos ir IS nauda, kurios jie tikisi, kur kas detaliau, nei prieš tai vykdytame etape. Pagrindiniai sukuriami dokumentai – funkcinis aprašymas (kartais vadinamas vartotojo specifikacija, ar reikalavimų specifikacija) ir bent bendrais bruožais nusakytas sistemos testavimo planas, kuris bus naudojamas vėliau priimant sistemą vartoti.

Šio etapo reikšmė galutinei projekto sėkmei yra ypatingai didelė, nedidelis resursų sutaupymas šiame etape vėlesniuose gali virsti didelėmis ir brangiomis problemomis.

Šios fazės metu atliekami išsamūs tyrimai, susisteminami poreikiai, jie kompleksiskai patikslinami, įvertinami informacijos ištekliai. Gautos informacijos pagrindu išrenkamas tinkamiausias IS modelis, nustatomi reikalingi ištekliai.

Pagrindiniai metodai, naudojami poreikių analizei:

- Interviu su įvairaus lygmens darbuotojais – nuo galutinių vartotojų iki aukščiausio lygio vadovų;
- Esamų verslo procesų, dokumentų srautų ir mainų analizė;
- Anketų išsiuntinėjimas;

- Esamų procedūrų stebėjimas, tiesiogiai leidžiant laiką organizacijos struktūriniuose padaliniuose, stebint ir analizuojant jų vykdomas procedūras, jų tarpusavio sąveiką.

Tačiau čia taip pat pasireiškia IS specifika, dažnai lemianti, kad neįmanoma ar labai sudėtinga išanalizuoti būsimų sistemos vartotojų poreikius. Čia naudojamas hipotetinis poreikių modeliavimas, įvertinant gausybę faktorių, tokių kaip vartotojų kultūriniai skirtumai, pirkimo įgūdžiai, socialinė ir turinė padėtis, t.t. Šių faktorių įvairovė ir neapibrėžtumas daro šį reikšmingą IS gyvavimo ciklo etapą lemtingą ir gana rizikingą. Kuo daugiau sistema remiasi sąveika su išoriniais vartotojais, tuo labiau hipotetinis yra vartotojų poreikių nustatymas.

2.6.6. Sistemos projektavimas

Sistemos projektavimo (*system design, technical design*) fazėje pagrindinė keliama užduotis – pritaikyti reikalavimams skirtingus techninius sprendimus. Šio etapo metu vėlgi reikėtų įvertinti, ar nėra jau sukurtų programinės įrangos paketų, atitinkančių IS išskeltus reikalavimus, išnagrinėti egzistuojančių sistemų ypatumus, vartotojo sąsajas, bei apsvarstyti galimybes pakartotinai panaudoti jau sukurtą programinį kodą ar komponentus.

Kaip šios veiklos rezultatas sukuriama detali techninė specifikacija programavimo darbams (arba techninė dalis konkursui skelbti, jei įranga bus perkama iš trečiosios šalies), o taip pat sistemos testavimo planas, kuriuo bus naudojamosi testuojant sistemą. Šio etapo metu gali būti baigta ruošti sistemos naudojimo dokumentacija –skirta tiek vartotojams (vartotojo vadovas – *user manual*), tiek, didesnės IS atveju, techniniam personalui, palaikančiam IS veikimą.

Projektavimo etapo metu susiejami vartotojo poreikiai ir programinės bei techninės įrangos galimybės. Be verslo poreikių projektavimo techninių sprendimų visumoje, projektavimo etape turėtų būti užtikrinamos tokios sistemos savybės, kaip patikimumas, saugumas, pajėgumas ir plečiamumas.

Išskiriamos tokios charakteringos šios fazės veiklos:

- Bazinių projektinių darbų atlikimas;
- Atskirų techninių užduočių sudarymas;
- Konceptualaus projektavimo atlikimas;
- Techninių specifikacijų ir instrukcijų sudarymas;

- Projektavimo veiklos rezultatų pristatymas, ekspertizė ir patvirtinimas.

Pagrindinis principas, kurio reikia laikytis projektavimo metu: geriausia sistema – tai pati paprasčiausia sistema, atitinkanti keliamus reikalavimus. Projektavimas gali būti atliekamas rankinėmis priemonėmis arba automatizuotai. Pastaruoju atveju naudojamos CASE (*Computer Aided Systems Engineering*) priemonės, leidžiančios ženkliai sutrumpinti programinių projektų sukūrimo terminus, efektyviau panaudoti komandinį darbą, ir orientuotos į vizualines programinės įrangos komponentų specifikavimo priemones. CASE technologijų teikiamos vizualinės projektavimo priemonės leidžia sudaryti IS projektą, lengvai suprantamą žmonėms, kurį vėliau galima automatizuotai konvertuoti tiesiai į programinį kodą.

Kompiuterizuotoje IS paprastai sistemos užduotys skaidomos į smulkesnius modulius, šie taip pat detalizuojami, tokiu būdu prieinant prie paprastų ir lengvai suvokiamų bei realizuojamų detalizuotų elementų.

IS projektavimo etape definuojamos ir projektuojamos šie elementai:

- Techninė įrangos platforma: parenkamas kompiuterių tipas, tinklo galimybės, įvedimo bei išvedimo įrenginiai, kaupikliai;
- Programinė įranga: programavimo kalba, PĮ paketas, duomenų bazių valdymo sistema;
- Išvedama informacija: ataskaitų bei sistemos išvaizdos projektavimas;
- Įvedama informacija: formos, dokumentai, įvedamų duomenų tikrinimo procedūros;
- Vartotojo sąsaja;
- Modulinė architektūra: sudėtinės programinės įrangos dalys ir jų tarpusavio sąveika;
- Testavimo planas, testavimo duomenys;

Perėjimo prie IS planas: kaip bus pereinama nuo esamos organizacijos būsenos prie naujos IS;
Dokumentacija: vartotojo ir techninė.

2.6.7. Sistemos kūrimas

Sistemos kūrimo (*system development, system build, system production, implementation*) etape atliekamas projektinių darbų koordinavimas ir operatyvi kontrolė, kuriamos posistemės, jos apjungiamos ir testuojamos. Tipinė šio etapo struktūra:

- Reikiamos techninės ir programinės įrangos įsigijimas;
- Programavimo ir programų derinimo stadija: programuojami, derinami ir autonomiškai testuojami kuriamos IS moduliai. Autonominio testavimo rezultatai protokoluojami, rašomos trūkstamos sistemos naudojimo instrukcijos, kiti jos eksploatavimo dokumentai. Dokumentacijos sudėtis ir jos apipavidalinimo reikalavimai nurodomi techninėje užduotyje;
- Sistemos surinkimo ir aprobavimo stadija. Pagrindiniai tikslai:
 - Sujungti visus sistemos komponentus į vieną visumą ir patikrinti to sujungimo korektiškumą;
 - Patikrinti, ar visi techninės užduoties reikalavimai yra įgyvendinti, ir pašalinti sistemoje esančius techninės užduoties reikalavimų pažeidimus;
 - Parengti sistemą tiražavimo ir diegimo darbams.

Sujungus sistemos komponentus į vieną visumą, atliekamas struktūrinis sistemos testavimas, tikslinama eksploatacinė dokumentacija ir atliekami kiti planuoti sistemos bandymai.

Sistemos bandymų rezultatus verta protokoluoti, vėliau šie protokolai gali būti panaudojami ne tik priimant sistemą, bet ir ją plečiant bei tobulinant.

Kaip atskiri šios fazės procesai, o kai kada ir kaip atskiros fazės gali būti išskiriami testavimo etapai: integralumo testavimo metu tikrinamas techninis sistemos korektiškumas ir vientisumas, atitikimas techninei specifikacijai, atsparumas numatytiems ar net viršijančioms numatytas veikimo sąlygoms, neplanuotiems vartotojų veiksams. Kitaip tariant, integralumo testavimo metu tikrinama, *kaip* sistema dirba.

Priimtimumo testavimas (*acceptance testing*), skirtingai nei integralumo testavimas, numato tikrinimą, ar sistema atlieka *tai, ko iš jos reikia* organizacijai, ar sistema parengta vartotojų darbui su ja, ir ar sistema pilnai atitinka visus jai iš anksto numatytus reikalavimus.

Priimtimumo testavimas yra bene menkiausiai suprantamas sistemos vystymo etapas, ir nėra tikslių metodų, kaip tai turėtų būti atliekama. Nuo to laiko, kai nustatyti reikalavimai

kuriamai sistemai, ir šio etapo gali būti praėjęs ženklus laiko tarpas, ir reikalavimai galėjo pakisti. Nuolatiniai reikalavimų kitimai sistemos vystymo procese – tai faktorius, neretai neigiamai ir gana stipriai įtakojantis galutinio produkto priimtumo testavimo rezultatus.

IS atveju priimtumo testavimas dėl vartotojų auditorijos anonimiškumo, jų įvairovės ir tikslaus poreikių nežinojimo tampa labai komplikotas. Neretai jis vyksta jau įdiegus sistemą. Tai nuolatinio pobūdžio procesas, kurio eigoje išsiaiškinti trūkumai šalinami, sistema keičiama ir tobulinama.

2.6.8. Sistemos diegimas

Šis etapas vadinamas įvairiai – instaliavimo, atidavimo eksploatacijai, priėmimo naudojimui ir pan. Tai – sistemos pateikimo naudotojams stadija. Ją galima nagrinėti užsakovo ir vykdytojo požiūriu.

Užsakovo požiūriu vyksta įsigytosios programų sistemos diegimas. Jo atliekami darbai nepriklauso nuo to, kokių būdu sistema buvo įsigyta, piršta ar sukurta. Sistema yra instaliuojama kompiuteriuose, naudotojai mokomi su ja dirbti, duomenų bazėse kaupiami sistemai dirbti reikalingi duomenys ir atliekami kiti parengiamieji darbai. Jei sistema diegiama vietoje kitos anksčiau naudotos sistemos, tai šioje stadijoje demontuojama ankstesnioji sistema.

Vykdytojų atliekami darbai priklauso nuo to, kaip ir kam sistema buvo kuriama. Jeigu ji buvo sukurta pagal konkretų užsakymą, tai vykdytojai užsiima diegimo darbais. Kartais jie kurį laiką vykdo autorinę įdiegtosios sistemos priežiūrą. Jeigu sistema buvo kuriama ją parduoti rinkoje, tai vykdytojai šioje stadijoje užsiima sistemos distributyvo ir jos dokumentacijos tiražavimu, reklama ir kitais su sistemos pardavimu rinkoje susijusiais darbais.

Žiūrint iš techninės pusės, sistemos įdiegimo procesas gali būti išskaidytas į tokius etapus:

- Techninės įrangos įdiegimas: laidų tiesimas, techninės įrangos sumontavimas, išdėstymas patogiu vartotojui naudoti būdu. Gali prireikti organizacijos patalpų pertvarkymo, todėl svarbu numatyti, kiek laiko gali užimti šis etapas;
- Programinės įrangos diegimas, jei naudojama esama techninė įranga;
- Vartotojų apmokymas: priklausomai nuo esamos IS ir kitų organizacinių procesų skirtumo nuo naujosios IS, tam gali prireikti įvairaus lygio pastangų – nuo minimalių

(jei vartotojams jau pažįstama sistemos vartotojo sąsaja, darbo su kompiuteriais ir specializuota taikomąja programine įranga principai, jei neprireiks ženkliai keisti jų darbo metodų) iki ilgalaikių ir kompleksinių (jei nauja IS pasižymi dideliu kokybiniu skirtumu nuo esamų procesų, jei prireiks keisti darbo įpročius, keisis darbo aplinka, vartotojams prireiks naujų įgūdžių, norint naudotis IS). Kadangi IS skirtos kuo efektyvesniam vartotojų darbui, vartotojai turi būti kuo geriau supažindinti su specialiai jiems sukurta sistema, kad perprastų visas jos funkcijas, kartais netgi veikimo principus, žinotų visas sistemos galimybes, kurių tik jiems gali prireikti darbe.

- Esamų duomenų perkėlimas į naująją sistemą. Jei duomenų formatai labai skirtingi, šis procesas gali pareikalauti didelių laiko sąnaudų.

Dažnai perėjimas prie naujos IS užima pakankamai ilgą laiko tarpą, kuris gali būti labai nepalankus nuolatinės konkurencijos sąlygomis veikiančiai organizacijai. Todėl skiriama keletas metodų naujų IS įdiegimui:

- Tiesioginis pakeitimas: paprastai vienos dienos vakarą vartotojai baigia naudoti seną sistemą, o kitą dieną pradeda naudoti naują. Tai įmanoma tik tada, kai yra galimybė perkelti visus senosios sistemos duomenis į naująją;

- Paralelinis perėjimas: tam tikram laiko tarpui (paprastai keletui savaitių ar mėnesių) lygiagrečiai naudojamos abi sistemos – senoji ir naujoji. Šis metodas leidžia nuoseklų perėjimą prie naujos IS, leidžia išvengti psichologinio vartotojų diskomforto, perėjus dirbti į naują aplinką, ir problemų bei darbo sutrikimų, kylančių susidūrus su nepažįstama sistema;

- Palaipsninis perėjimas: taikomas didelėse sistemose, pasižyminčiose aiškia moduline struktūra. Tokios sistemos gali būti diegiamos palaipsniui, skirtingu metu įdiegiant atskirus sistemos modulius. Šis metodas leidžia išvengti problemų, kylančių vienu ėjimu įdiegiant naują sistemą, ir kartu leidžia nedelsiant imti naudotis naujos sistemos teikiamais privalumais. Taip pat, palaipsniui diegiant sistemą vartotojams lengviau prie jos priprasti ir išmokti ją naudotis, nes problemų jiems gali iškilti tik toje srityje, kurioje įdiegiamas naujas modulis, o ne su visa sistema iš karto. Be to, pavieniai moduliai, jei jų vartotojo sąsaja ir darbo principai sutampa su likusiais, paspartina vartotojų apmokymą naudotis naująja IS;

Pilotinis perėjimas: nauja sistema iš pradžių diegiama tik viename ar keliuose organizacijos padaliniuose. Tai leidžia laiku išaiškinti sistemos trūkumus, bandant ją realioje organizacijos aplinkoje, apmokyti darbo su sistema dalį vartotojų, kurie vėliau gali apmokyti kitus darbuotojus bei padėti jiems spręsti iškilusias problemas.

2.6.9. Naudojimo ir palaikymo fazė

Šioje stadijoje sistema yra naudojama konkrečioms uždaviniais spręsti, vykdomas jos aptarnavimas, ji yra tobulinama. Sistemos aptarnavimas apima duomenų bazių administravimą, periodišką kontrolinį testavimą, eksploatacijos metu pastebėtų klaidų šalinimą ir kitus panašaus pobūdžio darbus. Sistemai pasenus, pradedamas naujos jos versijos kūrimas – gyvavimo ciklas kartojasi iš naujo.

Priklausomai nuo sistemos kūrėjų ir užsakovų susitarimo, sistemos naudojimą ribotą ar neribotą laiko tarpą gali lydėti sistemos palaikymas. IS palaikymas – tai gali būti sistemos priežiūros darbų atlikimas, optimizavimas, veikimo kokybės analizė, saugumo monitoringas, naujų funkcijų įdiegimas, klaidų ištaisymas, pertvarkymas pritaikant prie kintančių organizacijos veiklos sąlygų.

Išskiriami trys IS palaikymo kryptys:

- Tobulumo siekimas: sistema nuolat tobulinama, nors ir veikia tvarkingai, ieškoma naujų, efektyvių sprendimų. Laikomasi principo „Visados yra kur tobulėti“;
- Adaptyvus palaikymas: nuolat keičiantis organizaciniais procesams, keičiasi ir IS keliami reikalavimai, tad sistema yra adaptuojama, kad nuolat patenkintų jai keliamus reikalavimus;
- Korektyvus palaikymas: kad ir kaip akylai ir nuodugniai bebūtų sistema ištestuota jos kūrimo ir diegimo metu, tik realaus naudojimo sąlygomis, naudojant realius duomenis ir susiduriant su realiomis, nestandartinėmis ir nenumatytomis situacijomis aptinkamos paslėptos problemos. Ši palaikymo kryptis orientuota į tokių problemų fiksavimą ir sprendimą.

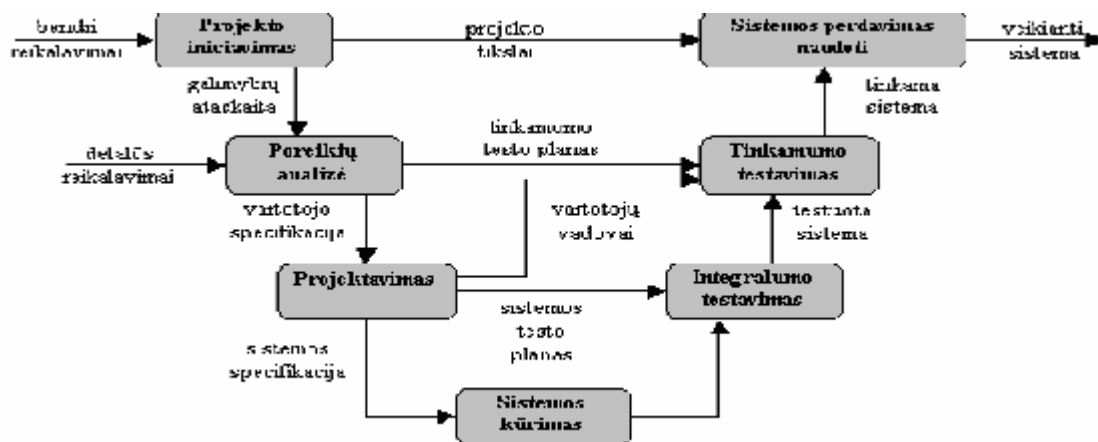
Neatsiejama naudojimo ir palaikymo fazės dalimi kai kuriuose šaltiniuose laikomas mokymasis (kai kur jis netgi išskiriamas į atskiras IS gyvavimo ciklo fazes – . Nuolatos apklausiant sistemos vartotojus (jie geriausiai žino sistemos trūkumus ir privalumus) išsiaiškinamos silpnosios ir stipriosios sistemos vietos. Kiekviena sistema, ypač komercinėse organizacijose, kur pokyčiai vyksta labai greitai, sulaukia tokio momento, kai ji nebeatitinka jai keliamų reikalavimų, ir iškyla poreikis naujos IS kūrimui. Štai šioje vietoje žinios apie ankstesniųjų IS vystymą, jo problemines sritis, patirtis dalyvaujant projektavime, aiškinantis reikalavimus sistemai tampa neįkainojamos.

IS atveju sistemos palaikymą palengvina tai, kad beveik nereikalinga vartotojų kompiuterių priežiūra (atnaujinimų diegimas, perkonfigūravimai ir pan.). Nėra svarbu, koku įrenginiu jie

naudosis (stacionariu ar nešiojamuoju, kišeniniu kompiuteriu ar PDA) – svarbu, kad šie įrenginiai turėtų prieigą prie interneto ir naršyklę.

2.6.10. IS gyvavimo ciklo fazių tarpusavio ryšiai

Aukščiau aptarėme, kokie etapai sudaro IS gyvavimo ciklą, koks jų eiliškumas ir kokia kiekvieno jų vidinė struktūra. Tačiau platesniam gyvavimo ciklo vaizdai parodyti reikėtų parodyti, kaip yra tarpusavyje susiję šie etapai, kokia informacija turima kiekvieno jų pradžioje ir kokia informacija, kaip etapo veiklos rezultatas, perduodama tolesniems etapams. Tai labai gerai atskleidžia vadinamoji „V schema“ (pav. Nr.1).



1 pav. IS gyvavimo ciklo etapai ir tarpusavio ryšiai

Tiesa, šiame paveikslėlyje pateikiamas IS gyvavimo ciklas ne visai sutampa su aukščiau aptartais ciklo etapais, tačiau toks ciklo vaizdavimas leidžia geriau suprasti sistemos būsenų kaitą jos gyvavimo ciklo eigoje. Mūsų aptartame IS gyvavimo cikle integralumo ir tinkamumo testavimo etapai integruoti į sistemos kūrimo etapą, o veikianti sistema dar patenka į papildomą etapą – naudojimo ir palaikymo fazę. Palaikymo fazė, nepavaizduota šioje schemoje, suteikia IS gyvavimo ciklui iteratyvumo – kai nuo galutinės sistemos būsenos grįžtama į ankstesnius jos etapus ir sistema tobulinama.

Gyvavimo ciklo fazių reikšmė ir problematika

Pirminės projekto fazės turi lemiamą reikšmę projekto sėkmingumui, nes jose priimami fundamentalūs sprendimai, įtakojantys kituose etapuose priimamus sprendimus ir nulemiantys informacijos sistemos kokybę. Paprastai 30 procentų indėlis į galutinį rezultatą įnešamas projekto inicijavimo bei sistemos apibrėžimo ir analizės etapuose, 20 procentų – projektavimo etape, 20 procentų – sistemos kūrimo etape ir 30 procentų – diegimo bei naudojimo ir palaikymo etapuose.

Be to, sistemos projektavimo stadijoje padarytų klaidų aptikimui sugaištama maždaug dvigubai daugiau laiko, nei vėlesnėse stadijose padarytoms klaidoms aptikti, o jų ištaisymas atsieina penkis kartus brangiau. Todėl pirminėse projekto stadijose reikia kuo atidžiau išnagrinėti visą įmanomą informaciją ir priimti tik tvirtai argumentuotus, į perspektyvą orientuotus sprendimus.

Dažniausiai pirminėse stadijose įvykdomos tokios klaidos:

- Klaidos nustatant užsakovo interesus;
- Per didelio dėmesio skyrimas nereikšmingiems, šalutiniams dalykams;
- Neteisingas pradinio uždavinio formuluotės interpretavimas;
- Neteisingas ar nepakankamas detalių supratimas;
- Neišbaigtos sisteminių reikalavimų funkcionalumo specifikacijos;
- Klaidos nustatant reikalingus resursus ir terminus;
- Retas tikrinimas, ar suderinti tarpusavyje įvairūs etapai, ir kontrolės iš užsakovo pusės nebuvimas (užsakovas neįtraukiamas į sistemos kūrimą).

IS gyvavimo ciklo santykis su organizacijos sistemos ir programinės įrangos gyvavimo ciklais

Kaip jau minėta, IS yra sudėtinė organizacijos veiklos sistemos dalis, ir kartu pati savaime veiklos sistema. Visai natūralu, kad veiklos sistemos gyvavimo ciklą taip pat sudaro šie pagrindiniai etapai: inicijavimas, projektavimas, kūrimas ir naudojimas bei palaikymas. IS veikla apsiriboja tik šešiomis pagrindinėmis operacijomis su informacija: gavimu, perdavimu, saugojimu, paieška, apdorojimu ir pateikimu, taigi yra siauresnio pobūdžio veiklos sistema.

Tradicinių IS atveju sistema būdavo parenkama kaip pagalbinis veiklos sistemos funkcionavimo įrankis, taigi IS gyvavimo ciklas tapdavo sudėtinė VS gyvavimo ciklo dalimi. Kadangi ankstyvosios IS būdavo naudojamos konkrečioms funkcijoms automatizuoti, tai jų pritaikymo ribos buvo pakankamai siauros, todėl galima teigti, jog tradicinės IS gyvavimo ciklas persidengdavo su VS gyvavimo ciklu tik vėlesniuose etapuose (visai jie nesutapdavo iniciavimo etape, gana silpnai – projektavimo, ir daugiau kūrimo bei naudojimo etapuose). Augant IS įtakai verslo procesų formulavimui, šie gyvavimo ciklai įgavo tendenciją persidengti vis glaudžiau. Šiuolaikinių IS gyvavimo ciklas jau pralenkė VS gyvavimo ciklą, nuo IS gyvavimo cikle generuojamų naujų galimybių, kurių užuomazgos suformuojamos iniciavimo ar projektavimo etapuose, priklauso veiklos sistemos iniciavimo etapo pagrindą sudarančios ir naujos VS kūrimą ar esamos pokyčius inicijuojančios idėjos ar išaiškėjusios

problemos. Tad galima teigti, kad IS ne tik pagilino IS ir VS gyvavimo ciklą tarpusavio santykį, bet ir pastatė IS prieš VS, IS tampant VS gyvavimo ciklo iniciatoriumi.

Bet kokia šiuolaikiška kompiuterizuota IS negali egzistuoti be programinės įrangos. PĮ kūrimas yra sudėtinė ir viena pagrindinių veiklų informacijos sistemos gyvavimo cikle, tad galima teigti, kad PĮ gyvavimo ciklas beveik identiškas IS gyvavimo ciklui. Paprastai bet kokia sudėtingesnė programinė įranga kuriama ne kaip atskiras produktas, o kaip integrali kažkokios didesnės sistemos (IS) dalis.

Egzistuoja tarptautinis standartas, reglamentuojantis programinės įrangos gyvavimo ciklą – ISO/IEC 12207. Pagal šį standartą, PĮ gyvavimo ciklo procesai sugrupuoti į tris klases:

- Pagrindiniai (pirminiai) procesai: įsigijimas, tiekimas, vystymas, eksploatavimas, palaikymas;
- Pagalbiniai procesai, užtikrinantys pagrindinių procesų vykdymą: dokumentavimas, konfigūracijos valdymas, kokybės užtikrinimas, verifikacija, atestacija, įvertinimas, auditas, problemų sprendimas;
- Organizaciniai procesai, inicijuojami gyvavimo ciklo procesams pradėti, kontroliuoti ir gerinti: projektų valdymas, projekto infrastruktūros kūrimas, gyvavimo ciklo gerinimas, mokymas.

Šis standartas nustato tvirtą sąsają tarp sistemų ir programų. Visų pirma, standartas remiasi bendrais sistemų inžinerijos principais. Pagrindiniai sistemų inžinerijos komponentai (tokie kaip analizė, projektavimas, gamyba, įvertinimas, testavimas, įdiegimas, kokybės užtikrinimas/kontrolė, platinimas ir pan.) leidžia pritaikyti juos ir programų inžinerijai. Programų inžinerijoje „gamyba“ suvokiama kaip „programavimas“.

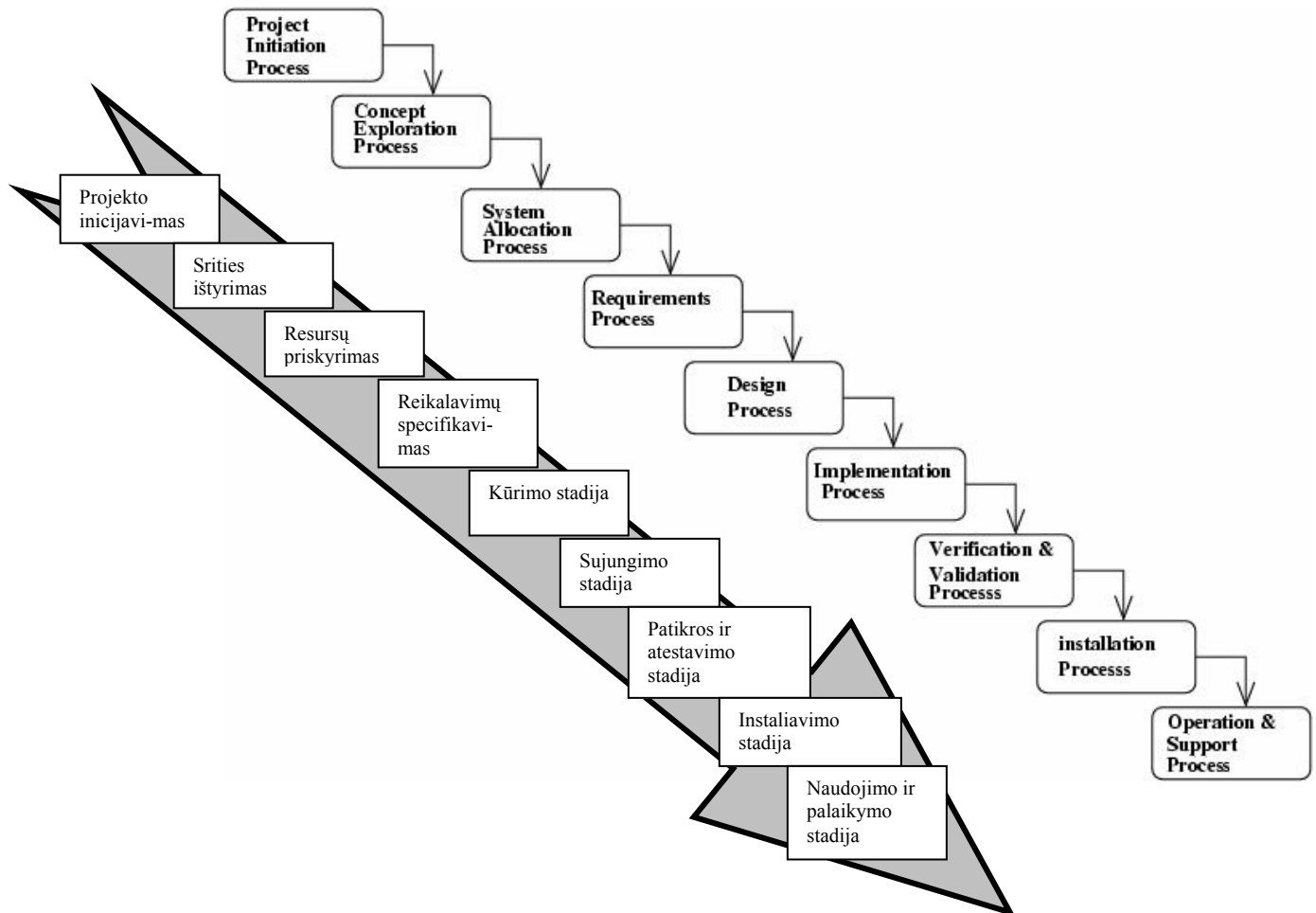
Antra, standarte programinė įranga traktuojama kaip integrali visos sistemos dalis, atliekanti joje specifinę funkciją ar funkcijas. Tai įgyvendinama atskiriant programinę įrangą nuo sistemos, ją vystant (lygiagrečiai su kitais sistemos komponentais) ir vėl integruojant atgal į sistemą.

Tačiau nepaisant glaudžios sąsajos tarp IS ir programinės įrangos gyvavimo ciklą, kokybinis šios sąsajos pobūdis yra kitoks, nei IS ir VS gyvavimo ciklą atveju. PĮ gyvavimo ciklas yra tik sudėtinė IS gyvavimo ciklo dalis, kurios paskirtis – sistemos funkcionalumo įgyvendinimas.

IS kūrimo ir gyvavimo ciklo modeliai (Pav. Nr. 2)

Krioklio modelio problemos:

- Nelankstus projekto skaidymas į skirtingas stadijas
- Kelia sunkumų norint reaguoti į vartotojų reikalavimų pasikeitimus
- Šis modelis taikomas tik tada, kai reikalavimai yra aiškiai suprantami.



2 pav. Kaskadinis (Krioklio) modelis. Aiškiai atskirtos specifikacijos ir kūrimo fazės

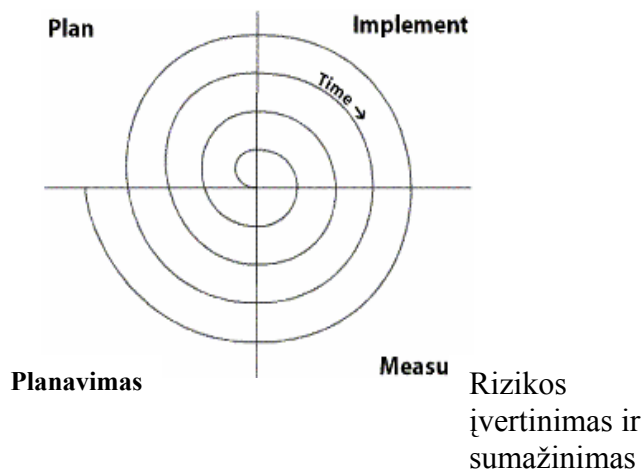
Spiralinis modelis - kai programa vystoma ciklais (spirale). (Pav. Nr.3)

- procesas pateikiamas kaip spirale, o ne kaip veiksmų seka su pakartojimais
- Kiekvienas spiralės ciklas yra proceso fazė
- Nėra fiksuotų fazių, tokių kaip specifikacija ar projektavimas - spiralės ciklai pasirenkami pagal poreikį
- Rizika yra aiškiai įvertinama ir mažinama proceso metu

Spiralinio modelio sektoriai:

- Tikslo nustatymas
- kiekvienai fazei nustatomas specifinis tikslas

- Rizikos įvertinimas ir sumažinimas
- rizika įvertinama ir vykdomi papildomi veiksmai siekiant sumažinti pagrindinius rizikos faktorius
- Kūrimas ir atestavimas
- sistemos vystymo modelių pasirenkamas vienas iš bendrų modelių
- Planavimas
- projektas peržiūrimas ir suplanuojama kita spiralės fazė.



3 pav. Spiralinis modelis - kai programa vystoma ciklais (spirale).

Kiti IS kūrimo ir gyvavimo ciklų modeliai:

- Prototipinis modelis (geras pirmas žingsnis, kuomet klientas turi pagrįstus reikalavimus, bet nenusimano apie detales, kūrėjas privalo atsispirti spaudimui išplėsti pradinį prototipą iki darbinio produkto)
- Greito taikymo ir kūrimo modelis (Rapid Application and Development (RAD) Model) (įgalina pakartotinai naudoti programinės įrangos komponentus su ypatingai trumpu kūrimo ciklu)
- Augantis modelis (PĮ praplečiama mažomis, naudingomis dalimis, kurių kiekviena pridedama prie jau sukurtos dalies)
- Dvipusis (Win-Win) spiralinis modelis (išsiaiškinami PĮ reikalavimai, apibrėžti kliento ir kūrėjo derybomis, kur kiekviena pusė siekia subalansuoti techninius ir biznio suvaržymus)
- Konkuruojantis kūrimo modelis (panašus į spiralinį modelį, dažnai naudojamas kuriant kliento/serverio architektūra)
- Komponentais grįstas kūrimas (spiralinio modelio variacija, kurioje programa yra kuriama iš PĮ komponentų, vadinamų klasėmis)

- Taisyklingų metodų modelis (griežta, tiksli matematinė notacija naudojama apibrėžti, kurti ir tikrinti kompiuterines sistemas)

Ketvirtos kartos technikos (Fourth Generation (4GT) Techniques) (PĮ įrankis naudojamas pirminio kodo generavimui aukšto lygio pateikiamų techninių reikalavimų PĮ sistemai)

3. PROJEKTINĖ DALIS

3.1. Bendra terminologija ir paaiškinimai

Reikalavimų tipai:

- Funkciniai reikalavimai. Sistemos paslaugų aprašymai dar turėtų paaiškinti, kaip sistema reaguosoti į ypatingus duomenų įvedimus ir kaip sistema elgsis ypatingose situacijose.
- Nefunkciniai reikalavimai. Sistemos paslaugų arba funkcijų apribojimai, tokie kaip laiko apribojimai, kūrimo proceso apribojimai, standartai, ir pan.
- Srities reikalavimai. Sistemos taikymo srities reikalavimai, atspindintys tos srities charakteristikas.
- Projekto tikslai. Priežastys ir biznio procesai kurie verčia vykdyti šį projektą.
- Reikalavimai produktui. Reikalavimai, kurie apibrėžia, kad pateiktas produktas privalo elgtis specifiniu būdu, pvz. Vykdyimo greitis ir patikimumas ir pan.

Patikra ir atestavimas

Patikra ir atestavimas turi parodyti, kad ši programinė įranga yra tinkama skirtam tikslui. Tai nereiškia kad ši programa bus visai be defektų, tačiau ji bus pakankamai gera numatomam naudojimui, ir naudojant bus pasiektas reikiamas tikrumo (pasitikėjimo) lygis. Reikalavimus tikrinti reikia jau tuo momentu kai jie tik pradedami rašyti, kadangi pagal juos turi būti sukurtas veikiantis produktas kuris atlieka tai ko iš jo tikisi vartotojas.

Patikra. Programinė įranga turi atitikti specifikaciją

Atestavimas. Programinė įranga turi daryti tai ko vartotojas reikalauja

Reikalavimų identifikavimas

Reikalavimo numeris - tai unikalus reikalavimo kodas pagal kurį jį galima identifikuoti;

Reikalavimo tipas - tai sekcijos numeris, kuris pažymi kurioj reikalavimų specifikavimo dokumento vietoj reikalavimas yra. Šis numeris nėra būtinas, nes reikalavimas jau turi unikalų identifikatorių (reikalavimo numerį). Reikalavimo tipas tarnauja kaip priminimas, su kuria projekto dalimi reikalavimas rišasi, taip pat padeda pašalinti dublikatus bei to paties reikalavimo santrumpas.

Įvykio, naudojimo sąlygos identifikavimas

Įvykio numeris, identifikuoja biznio(veiklos) įvykį prie kurio šis reikalavimas priskiriamas.

Naudojimo sąlygos numeris identifikuoja produkto vartojimo atvejį kada šis reikalavimas yra aktualus. Šis identifikatorius yra svarbus, kadangi vienas reikalavimas gali būti susijęs su keletu įvykių.

Reikalavimo aprašymas - tai vieno sakinio ilgio reikalavimo santrauka.

Reikalavimo prasmė paaiškina, kodėl reikalavimas yra svarbus. Įsigilinimas į reikalavimo prasmę dažnai padeda suprast priežastį kodėl šis reikalavimas sudarytas.

Tinkamumas – tai objektyvus reikalavimo atitikimo matavimas. Tai kriterijus pagal kurį galima nustatyti ar tam tikras sprendimas atitinka reikalavimą.

Vertė vartotojui. - tai apytikslis nustatymas kiek vartotojui svarbus atitikimas šiam reikalavimui. Paprastai vartotojai įvertina reikalavimus balais nuo 1 iki 5 (1 žymi mažiausią svarbą o 5 didžiausią). Jeigu neįmanoma atitikti visų reikalavimų, tai reikalavimai suskirstomi pagal prioritetus.

Reikalavimų priklausomybės atseka reikalavimus kurie turi įtakos kitiems reikalavimams. Norint atsekti priklausančius reikalavimus reikia parinkti teisingus reikalavimų pavadinimus.

Konfliktai. Ši sekcija pažymi konfliktus tarp reikalavimų. T. y. jeigu vienas reikalavimas bus įgyvendintas, nebus įmanoma įgyvendinti kito reikalavimo. Konfliktai tarp reikalavimų nėra blogai, nes kai jie yra pažymėti ir sekami, tai tampa įmanoma juos išspręsti.

Reikalavimo istorija. Šioje sekcijoje rašoma visa pakeitimo bei reikalavimų atitikimo istorija.

3.2. Projekto aprašymas ir pritaikymas vartotojo problemai

Ši projekto sritis “Užsiėmimų tvarkaraščių projektavimo sistema” buvo pasirinkta žinant, kad bet kuriame universitete nemažai darbo užima tvarkaraščių kūrimas. Įvykdžius projektą tikimasi palengvinti universiteto personalo darbą.

Projekto įvykdymas motyvuojamas magistro laipsnio gavimu, bei galimu universiteto personalo dėkingumu už palengvintą tvarkaraščių sudarymo darbą.

Užsiėmimų tvarkaraščių projektavimo sistemos vartotojo problema yra gana rimta, nes akademiniam universiteto personalui kuriant studentų bei dėstytojų tvarkaraščius sunku atsizvelgti į abiejų pusių poreikius, todėl sudaryti grafikai neatitinka dėstytojų bei studentų interesų.

3.3. Projekto kūrimo tikslai

Užsiėmimų tvarkaraščių projektavimo sistema turi palengvinti universiteto personalo darbą sudarinėjant studentų mokymosi bei dėstytojų darbo tvarkaraščius. Galutinė užsiėmimų tvarkaraščių projektavimo sistema turi padėti vartotojui sutaupyti laiko sudarant užsiėmimo tvarkaraščius. Jeigu vartotojas pripažins, kad sistema jam padeda sutaupyti laiko, reiškias sistema yra tinkama savo tikslui.

3.4. Klientas, pirkėjas ir suinteresuoti asmenys

Klientas (arba žmogus suinteresuotas programos kūrimu)

Projekto vadovas - Doc.dr. Dalius Rubliauskas. Projekto vadovas suinteresuotas teisingo projekto pristatymu atestacinei komisijai ir tolimesniu panaudojimu informatikos fakultete.

Projekto pirkėjas

Sukurto produkto vartotojas - KTU Informatikos fakulteto akademinis personalas.

Kadangi užbaigtas produktas nieko nekainuos, projektas bus naudojamas tik jeigu jis atitiks kliento poreikius ir palengvins tvarkaraščių kūrimą.

Projekto pirkėjas taip pat yra ir Magistrinio darbo gynimo komisija, kadangi ji pagal pateiktą darbą turi įvertinti studentės galimybes dirbti tiriamąjį darbą.

Kitos suinteresuotos šalys

Projekto kūrėjas – KTU informatikos fakulteto II kurso magistrantė Dovilė Jociūtė. Projekto kūrėjas suinteresuotas sukurti užbaigtą produktą, ir jį apgynus gauti magistro laipsnį.

Projekto vartotojai.

Galutinis projekto vartotojas

Akademinis KTU, Informatikos fakulteto personalas. Projektas bus vartojamas, jei atitiks vartotojo poreikius ir palengvins darbą.

Vartotojų dalyvavimas kuriant produktą

Dalyvaus šios vartotojų grupės :

vartotojai su teorinėm informacinių sistemų kūrimo žiniom ir vartotojai suprantantys praktinę tvarkaraščių kūrimo pusę.

Projekto kūrėjas konsultuosis su visais projekto vartotojais. Jeigu vartotojas turės teorinių IS kūrimo žinių (pvz. projekto vadovas), tai jis galės patart projekto kūrimo procese iškilus IS kūrimo problemoms. Jeigu vartotojas gerai išmanys praktinę tvarkaraščių kūrimo pusę, tai jis galės padėt pritaikyt sistemą prie vartotojų poreikių.

3.5. Projekto apribojimai

Projekto kūrimo suvaržymai

Projektas bus įgyvendintas Windows operacinės sistemos aplinkoje, todėl projektuojant sistemą reikia atsižvelgti į Windows operacinės sistemos galimybes.

Projektas bus įgyvendinamas Windows OS aplinkoje, kadangi dauguma KTU akademiųjų vartotojų gerai susipažinę su šia aplinka.

Sprendimo kūrimo aplinka

Sprendimas bus kuriamas Microsoft Visual Basic aplinkoje, kaip savarankiška programa. Programa naudosis OLE sąsaja bendraujant su Microsoft Exel programa, tam, kad vartotojai galėtų pataisyti sugeneruotus grafikus, todėl naudojamaj sistemoj jau turės būti suinstaliuotas Microsoft Office paketas su Microsoft Exel programa. Projekto kūrėjas gerai susipažinęs su Microsoft Visual Basic aplinka, kurioje yra visi įrankiai greitai ir lengva sukurti vartotojo sąsają bei pasinaudoti kitu naudingumu funkcionalumu.

Kitos susijusios programos

Pagrindinė programa naudosis Microsoft Exel programos paslaugomis, per OLE sąsają, tam, kad pateikti vartotojų grafikus jiems suprantamoj ir lengvai modifikuojamoje aplinkoje.

Papildomi programiniai reikalavimai sistemai

Pprograma naudosis Microsoft Exel OLE serverio paslaugomis, todėl sistemoje jau turės būti suinstaliuotas Microsoft Office paketas su Microsoft Exel programa. Pasinaudojus Microsoft Exel OLE serverio paslaugomis, supaprastėja programos kūrimas, nes nebereikia kurti programos dalies atsakingos už grafikų taisymą. Microsoft Exel aplinka pažįstama daugeliui kompiuterių vartotojų, taigi, pasinaudojant ja atkris vartotojų apmokymo būtinybė.

Darbo su programa aplinka

Programa veiks bet kuriame x86 architektūros kompiuteryje su Microsoft Windows operacine sistema, bei Microsoft Office paketu (su suinstaliuota Microsoft Exel programa). Tai reiškia, kad programa veiks vartotojo kompiuteryje ir naudosis jo resursais. Kadangi programa bus kuriama Microsoft Visual basic aplinkoje, jos tipas - taikomoji programa. Tokio tipo programos veikia vartotojo kompiuteryje ir naudojami vartotojo resursais, taigi programos

virtotojas naudosis dabartine esama savo darbo vieta. Kadangi programos virtotojas naudosis savo ergonomiška darbo vieta, tai skaičiavimų laikas nėra kritinis faktorius.

Projektui skirto laiko apribojimai

Projektas turi būti užbaigtas iki antrojo semestro pabaigos. Projekto vykdymo eigoje teorinė bei praktinė dalys turi būti pristatytos dėstytojui. Kadangi magistro studijos baigiasi mokslo metų pabaigoje, magistrinis darbas tam laikui turi būti paruoštas pristatyti magistrinio darbo vadovui bei magistrinio darbo gynimo komisijai. Vystant magistrinio darbo projektą reikia naudotis darbo Vadovo konsultacijomis bei nurodymais.

Projekto biudžetas

Projekto biudžetas – asmeninės projekto kūrėjo lėšos laiko bei sugebėjimų resursai.

3.6. Sutartinė nomenklatūra ir srities terminai

Projektas – tai „Užsiėmimų tvarkaraščių projektavimo sistema“ kuri yra šio magistrinio darbo tikslas.

Vartotojo sąsaja. Kalbant tvarkaraščių generavimo kontekste taip vadinama šio projekto (UTPS) sąsaja. Kalbant tvarkaraščių taisymo kontekste taip bus vadinama Microsoft Exel programos sąsaja.

UTPS

Santrumpa „Užsiėmimų tvarkaraščių generavimo sistema“.

3.7. Darbo apimtis

Dabartinė situacija

Šiuo metu tvarkaraščiai yra sudaromi rankiniu būdu naudojant tam nepritaikytą programinę įrangą. Norima sukurti tvarkaraščiams sudaryti pritaikytą programinę įrangą, tam kad tvarkaraščiai būtų sudarinėjami lengviau, greičiau ir geriau.

Darbo kontekstas.

Akademinis KTU personalas moka naudotis Microsoft Exel programine įranga Microsoft Windows operacinės sistemos aplinkoje.

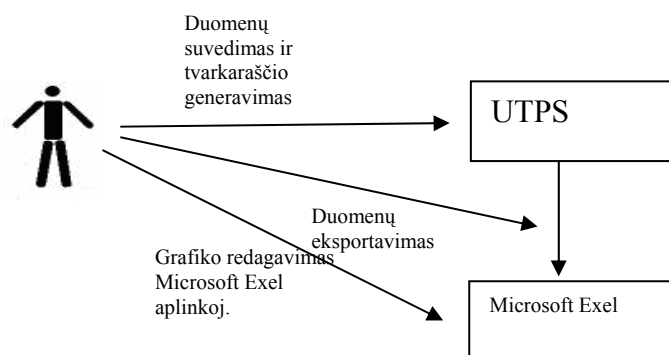
Darbo padalinimas

Programa atlieka grafikų generavimą bei eksportavimą į patogią redagavimui aplinką – Microsoft Exel. Vartotojas suveda pradinis duomenis į programą, bei pasinaudodamas Microsoft Exel aplinka paredaguoja galutinius rezultatus.

3.8. Produkto funkcijos (Pav. Nr. 4)

sukurti grafiką,

pervesti grafiką į redagavimui patogią aplinką – Microsoft Exel.



4 pav. Produkto funkcijos

3.9. Funkciniai ir duomenų reikalavimai

3.9.1. Funkciniai reikalavimai

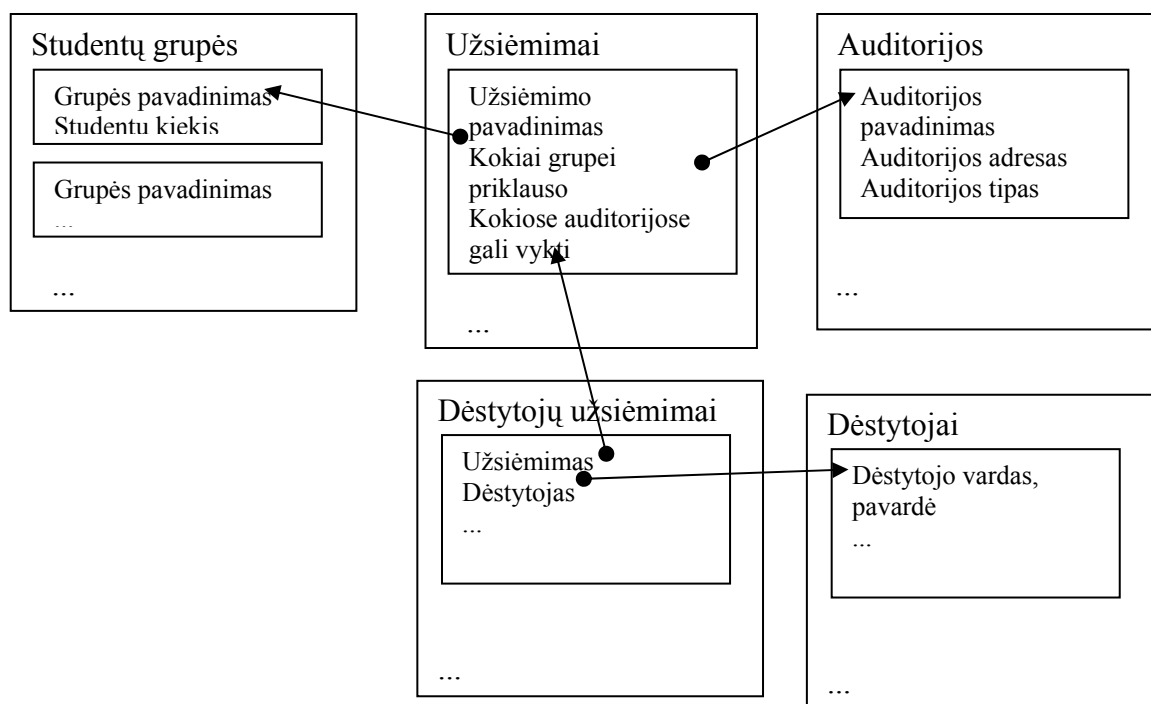
Kadangi sistema turi naudotis apmokytas personalas ji privalo veikti tik su teisingais duomenimis. Jeigu duomenys neteisingi sistema turėtų pranešti apie klaidą ir paprašyti įvesti teisingus duomenis. Jeigu sistema nesugeba atlikti užduoties turint esamus duomenis ji turėtų pranešti apie ypatingą situaciją, nurodyti galimą problemos šaltinį bei paprašyti pakoreguoti įvedamus duomenis. Vartotojas turi turėti teisę įvesti pradinis duomenis ir keisti galutinius sistemos rezultatus.

Reikalavimai duomenims (Pav. Nr. 5)

Pagrindinės duomenų grupės:

- Studentų grupelės

- Užsiėmimai.
- Auditorijos.
- Dėstytojai



5 pav. Reikalavimai duomenims

Kiekviena studentų grupelė turi turėti studentų kiekį.

Kiekviena auditorija turi:

- Turėt tipą.
- Turi turėt vietų skaičių.

Kiekvienas dėstytojas gali priklausyti bent vienam užsiėmimui ir kiekvienas užsiėmimas gali turėti vieną arba daugiau dėstytojų, todėl tarp jų turi būti ryšys daug su daug. Kadangi tokio ryšio realizuoti esamomis programavimo struktūromis negalima įvedamas papildomas duomenų įrašas: Dėstytojo užsiėmimas, kuris privalo būti susietas su vienu užsiėmimu, bei vienu dėstytoju. Kiekvienas dėstytojas turi gali būti susietas su vienu arba daugiau „Dėstytojo užsiėmimų“.

Kiekvienas užsiėmimas turi:

- Būti priskirtas grupei.
- Turi žinot kuriose auditorijose gali vykti.

- Turi būt žinomas paskaitų kiekis.

Turi būti susietas su vienu arba daugiau „Dėstytojo užsiėmimu“.

3.9.2. Sąsajos ir valdymo reikalavimai

Sąsaja

UTPS turi pateikti informaciją suprantamai vartotojui, susipažinusiam su UTPS vartotojo instrukcija forma.

Produkto stilius

UTPS turi atlikti užduotis per priimtina laiką kiekį.

UTPS turi būti sukurta per du semestrus (apytiksliai 8 mėn.).

UTPS turi padėti kiek įmanoma išvengti galimų klaidų bei subjektyvumo sudarant tvarkaraščius.

UTPS turi padėti atsižvelgti į pedagogų bei studentų kolektyvo poreikius ir galimybes, ir jei tik įmanoma, išvengti nereikalingų 'langų' studentam bei dėstytojams.

UTPS turi padėti kiek įmanoma sumažinti mokyklos administracijos darbo krūvį.

UTPS turi padėti kiek įmanoma optimaliau išnaudoti kabinetus ir kitas mokyklų patalpas.

UTPS turi kiek įmanoma atsižvelgti į visus psichologinius, higieninius bei pedagoginius reikalavimus, keliamus mokiniams ir mokytojams.

3.9.3. Naudojamumo reikalavimai

Naudojimo lengvumas.

UTPS vartotojo sąsaja turi būti įmanoma naudotis vartotojui susipažinusiam su UTPS vartotojo instrukcija.

UTPS neturi duoti klaidų pranešimų jeigu duomenys yra teisingi.

UTPS turi pati pareikalauti įvesti reikiamus duomenis.

UTPS turi pranešti kada ji generuoja grafiką ir kada generavimas yra baigtas.

Išmokimo lengvumas.

Bet kuris akademinis personalo atstovas susipažinęs su UTPS vartotojo instrukcija ir rankiniu grafikų generavimu turėtų sugebėti dirbti su sistema.

Bet kuris akademinis personalo atstovas susipažinęs su UTPS vartotojo instrukcija ir rankiniu grafikų generavimu turėtų sukurti vidutinio didumo grafiką (vertinant iš vartotojo pozicijos) per jam priimtina laiką tarpą.

Suprantamumo ir mandagumo reikalavimai

Bet kuris vartotojas susipažinęs su programų – meistrų veikimu turėtų lengvai suprasti UTPS veikimą. Produktas turėtų naudoti terminus ir žodžius, kurie yra gerai suprantami KTU akademiniam personalui.

3.9.4. Našumo reikalavimai

Greičio ir uždelimo reikalavimai

Vartotojo ir sistemos bendravimo uždelimas turėtų neviršyti 1 sekundės.

Vidurinio dydžio grafiko generavimo laikas turėtų neviršyti pusės valandos.

Programa turėtų atnaujinti grafiko generavimo progreso indikatorių bent kas 2 minutes.

Patikimumo ir prieinamumo reikalavimai

Programa turėtų būti pakankamai patikima, kad ją naudojantis nebūtų beprasmiškai prarastas vartotojo darbas. Ja turėtų būti įmanoma naudotis jei vartotojo kompiuteris teisingai sukonfiguruotas.

Lankstumo ir klaidų toleravimo reikalavimai

Įvykus klaidai programą turėtų būti įmanoma vykdyti nuo tos vietos, kurios rezultatai turimi. Idealiu atveju turėtų būti turimi įvesti, bei modifikuoti duomenys.

Gyvenimo ciklo reikalavimai

Kadangi palaikymui lėšų nėra skiriama, programa turėtų gyvuoti tol, kol ją galės palaikyti jos autorius. Tikimas gyvenimo ciklas be papildomo finansavimo - 1 metai. Su papildomu finansavimu 2 metai (tol kol ją pakeis nauja programos versija).

3.9.5. Darbo aplinkos reikalavimai

Veikimo aplinka: Programa bus naudojama sėdimoj vartotojo darbo vietoj. Programos naudojimo aplinkos higieniniai, triukšmo ir apšvietimo reikalavimai turi atlikti Darbo kodekso nustatytas normas.

Aparatūrinė aplinka: Programa bus naudojama x86 palaikomam kompiuteryje turinčiam pakankamai resursų, kad būtų galima naudotis windows 2000 operacine sistema.

Sąveikos programos: Programa bus OLE klientas ir per šia sąsają sąveikaus su Microsoft Exel OLE serveriu.

Gamybos reikalavimai: Produktas bus platinamas kaip ZIP archyvas. Produktą reikės įdiegti į sistemos vartotojo programoms skirtą katalogą ir nustatyti teises, kad vartotojai neturintys administratoriaus teisių negalėtų keisti programos turinio. Produktą galės įdiegti tik vartotojas susipažinęs su įdiegimo instrukcija.

3.9.6. Palaikymo reikalavimai

Palaikymas: Produktas bus palaikomas tokiais intervalais, kokiais laisvo laiko turės jo autorius. Ypatingos palaikymo sąlygos: Naują produkto versiją tikimasi išleisti po metų senos versijos naudojimo, kada bus sukaupta ir susisteminta senos versijos naudojimo patirtis.

Palaikomumo reikalavimai: Produktas turėtų reikalauti minimalaus apmokymo (supažindinimo su vartotojo instrukcija) ir konsultacijų su autorium (elektroniniu paštu iškilus papildomiems neaiškumams).

Pritaikomumo reikalavimai: Produktas turėtų veikti Windows 2000 ir Windows XP KTU akademinio personalo darbo vietose. Nėra tikimasi, kad produktą ateityje bus galima naudoti kitose aplinkose.

Instaliavimo reikalavimai: Reikėtų įsitikinti, kad sistema sutinka su programos naudojimo reikalavimais. Instaliuoti programą turėtų užtrukti ne daugiau kaip 20 minučių sistemos administratoriaus darbo laiko.

3.9.7. Saugumo reikalavimai

Priėjimo reikalavimai: Pradiniai duomenys turėtų būti prieinami tik generuojant grafikus juos naudojantiems vartotojams. Pakeistus galutinius rezultatus UTPS vartotojai turės patys išplatinti juos duomenų vartotojams.

Apsauga nuo duomenų keitimo: Pradinius duomenis keisti gali tik juos suvedę UTPS vartotojai.

Privatumo reikalavimai: UTPS sistemoje bus saugomi pradiniai duomenys apie dėstytojus bei studentų grupes. Kad užtikrinti saugumą tokie duomenys turėtų būti saugomi privačiame UTPS vartotojo kataloge, prieinamame tik jam pačiam ar ji pavaduojantiems asmenims.

Imuniteto reikalavimai: UTPS sistema turi pasinaudoti OS teikiamomis programos apsaugos nuo keitimo galimybėmis.

3.9.8. Legalumo reikalavimai

Suderinamumo reikalavimai: UTPS turi neturi pažeisti jokių LR įstatymų arba teisės aktų.

Standartizavimo reikalavimai: UTPS autorius turi naudotis jam pateiktais magistrinio darbo kūrimo reikalavimais.

3.9.9. Sukeliamos problemos

Kokias naujas problemas aplinkai sukels UTPS?

UTPS reikalaus, kad jos vartotojo kompiuteryje būtų suinstaliuota Microsoft Exel programa. Tai neturėtų būti didelė problema, kadangi dabar grafikai generuojami rankiniu būdu panaudojant Microsoft Exel programą.

Kokie vartotojai bus paveikti šios sistemos sukūrimo? Grafikus sudarantis akademinis personalas turės išmokti naudoti UTPS programa. Kadangi programa duos rezultata kaip Microsoft Exel failą, tai neturėtų sudaryt didelės problemos, reikės išmokti tik kaip teisingai suvesti duomenis į UTPS sistemą.

3.9.10. Užduotys

Ką reikia padaryti, kad sistema būtų paruošta darbui? Jeigu kompiuteris atitinka ir aparatūrinius ir programinius reikalavimus, tai tereikia įdiegti UTPS į vartotojų programų katalogą (paprastai c:\Program Files\UTPS) pasinaudojant sistemos administratoriaus sąskaita, bei įsitikint, kad vartotojas turi teisę naudotis UTPS paleidžiamaisiais bei papildomais failais be teisės juos keisti. Jei UTPS sistema jau įdiegta, tai reikia apmokyti personalą ja naudotis ir supažindinti su vartotojo dokumentacija.

Kūrimo fazės

Projektui kurti bus pritaikytas Krioklio gyvavimo ciklas (pav. Nr. 2)

Visoms fazėms įvykdyti (bandomajame modelyje) pilnai turėtu užtekti su x86 technologija suderinto kompiuterio, su veikiančia Windows 2000 operacine sistema ir suinstaliuotu Microsoft Exel.

3.9.11. Rizika

Valdomos rizikos

Labiausiai tikėtinos ir valdymo rizikos:

Neteisingai nustatyti UTPS užsakovo interesai.

Per didelio dėmesio skyrimas nereikšmingoms UTPS dalims.

Neteisingas UTPS tikslo nustatymas.

Neteisingas ar nepakankamas detalių supratimas.

Neišbaigtos sisteminių reikalavimų funkcionalumo specifikacijos.

Klaidos nustatant reikalingus resursus ir terminus.

Retas tikrinimas, ar suderinti tarpusavyje įvairūs etapai, ir kontrolės iš užsakovo pusės nebuvimas (užsakovas neįtraukiamas į sistemos kūrimą).

Nevaldomos arba mažai valdomos rizikos

Mažai valdomos rizikos:

Programinės įrangos komponentų nesuderinamumas.

Programavimo kalbos netikimas atitinkamai užduočiai.

3.9.12. Vartotojo dokumentacija ir apmokymas

Vartotojo dokumentacijos sudarymo planas

Vartotojui bus pateikta vartotojo instrukcija bei konsultacijos su sistemos autorium. Vartotojo instrukcija bus sudaryta iš:

Sistemos funkcinis aprašymas, kuriame yra trumpai apžvelgiamos sistemos galimybės ir paskirtis. Šis dokumentas kartu su vartotojo vadovu, padės nuspręsti, ar tai sistema, kurios vartotojams reikia.

Sistemos vadovas, kuriame turi būti aprašytos visos sistemos funkcijos ir galimybės bei jų naudojimas. Jame privalo bus pilnas pranešimų apie klaidas sąrašas su nuorodomis, kaip atstatyti galimus klaidų padarinius. Svarbiausias reikalavimas šiam dokumentui - pilnumas.

Sistemos instaliavimo skyrius, kuris skiriamas sistemos administratoriams. Jame bus nurodytos detalės, kaip konkrečioje aplinkoje instaliuoti sistemą, sistemą sudarantys failai, minimali reikalingos techninės įrangos konfigūracija.

Sistemos administratoriaus vadovas, kuriame turi būti aprašyti pranešimai, kaip sistema bendrauja su kitomis sistemomis, ir kaip reaguoti į šiuos pranešimus.

3.10. Detalusis projekto planas

3.10.1. Projekto aprašymas

Projekto paskirtis. Projekto paskirtis sukurti užsiėmimo tvarkaraščių projektavimo sistemą pritaikytą Kauno technologijos universiteto informatikos fakulteto poreikiams.

Projekto tikslai. Projekto tikslas - ne tik sukurti ir įdiegti sistemą, bet ir pritaikyti ją prie organizacijos darbuotojų poreikių.

3.10.2. Projekto dalyviai

Projekto organizacija. Projekto adresatas - Kauno technologijos universitetas (toliau KTU). Tai viena didžiausių ir iškiliausių šalies aukštųjų mokyklų, bei didžiausias techniškas universitetas Baltijos šalyse.

Projekto vadovas. Pabaigus etapą arba esant poreikiui (kai iškyla probleminiai klausimai) vykdomos konsultacijos su praktinės informatikos katedros vedėju doc. dr. Dalium Rubliausku.

3.10.3. Projekto komanda

Projekto komandą sudaro:

Pareigos	Asmuo	Atsakomybės	El. paštas
Projekto vadovas	doc. dr. Dalium Rubliausku	Projekto valdymas, konsultavimas	
Projekto vykdytoja	Dovile Jociūtė	Projekto dokumentavimas, planavimas, programavimas ir palaikymas	josiute@yahoo.com

3.10.4. Standartai

Projektavimo etapo metu bus nustatomi projekto kūrimo standartai. Šiame skyriuje jie bus pateikti pasibaigus projektavimo etapui.

3.10.5. Projekto kūrimo ciklas

Projektas kūrimo ciklas skaidomas į šias fazes:

1. Projektavimas
2. Dokumentavimas
3. Programavimas
4. Testavimas
5. Bandomoji eksploatacija
6. Stabilizavimas po bandomosios eksploatacijos

Vartotojo pusės dokumentavimas

3.10.6. Projekto tarpiniai rezultatai

Rezultatas	Pateikimo data	Pastabos
Projektavimo dokumentai	2005.01.30	Pateikiami dokumentai peržiūrai ir pastaboms
Projektavimo dokumentų pastabos	2005.01.30	Pateikiamos pastabos ir komentarai dėl sistemos
Projektavimo dokumentų patvirtinimas	2005.01.30	Gaunamas tarpinis etapo pažymys
Bandomosios versijos pateikimas	2005.01.30	Pristatomas evoliucinis prototipas
Galutinis pastabų sąrašas	2005.05.23	Pateikiamas bandomosios versijos pastabų sąrašas
Galutinės versijos pateikimas	2005.05.23	Instaliuojama į kliento kompiuterį galutinė sistemos versija
Dokumentacijos pateikimas	2005.05.23	Pateikiama klientui sistemos dokumentacija
Projekto pabaiga	2005.05.23	

3.10.7. Projekto galutinės pateiktys

Projekto galutinės pateiktys yra:

- Projekto paraiška
- Reikalavimų specifikacija
- Projektavimo metodologijos ir technologijų analizė
- Projekto planas
- Testavimo medžiaga:
 1. Testavimo planas
 2. Testavimo rezultatai ir išvados
- Vartotojo dokumentacija
- Kokybės vertinimo ataskaita
- Terminų ir santrumpų žodynas
- Programa

3.10.8. Projekto stebėjimas

Projektą stebės projekto vadovas, kuriam bus pristatomos ataskaitos apie projekto progresą ir pabaigus kiekvieną etapą bus pristatomos etapo pateiktys.

3.10.9. Rizikų valdymas

Projekte rizikos bus valdomos pagal žemiau pateiktą rizikų valdymo procedūrą. Rizikos peržiūrimos ir informacija apie įvykusias/potencialias rizikas atnaujinama kiekvieno reguliaraus projekto komandos susirinkimo metu.

Rizikų valdymo procedūra

1. Identifikuojama rizika.
2. Rizika registruojama, nurodant:
 - Rizikos pavadinimą
 - Rizikos aprašymą
 - Rizikos įtaką projektui
 - Rizikos prevencijos būdus
 - Veiksnius, kuriuos reikėtų atlikti nutikus rizikai

Periodiškai rizikos peržiūrimos ir patikrinamos

3.10.10. Reikalavimų ir pakeitimų valdymas

- Pokyčių poreikis gali būti inicijuotas vykdytojo arba kliento, kai tik atsiranda poreikis pakeitimams.
- Pakeisto dokumento patvirtinimas reiškia sutikimą pokyčiui.

Pakeitimų valdymo procedūra: Inicijuotas pokytis yra aprašomas esamoje projekto reikalavimų specifikacijoje.

3.10.11. Projekto peržiūros

Yra numatoma viena projekto peržiūra pirmojo semestro pabaigoje. Projekto peržiūrai reikia pristatyti šias pateiktis:

- Projekto paraiška
- Reikalavimų specifikacija
- Projektavimo metodologijos ir technologijų analizė
- Programos prototipą

3.10.12. Kritiniai kompiuterių resursai

Projekte nėra identifikuojami jokie kritiniai kompiuterių resursai.

4. VARTOTOJO DOKUMENTACIJA

4.1. Sistemos funkcinis aprašymas

4.1.1. Sistemos paskirtis

UTPS sistema skirta palengvinti grafikų kūrimo darbą KTU akademiniam personalui. Sistema turi sugebėti generuoti redaguojamus grafikus studentų grupėms, besimokančioms KTU. Tam tikslui ji turi saugoti pradinis duomenis, skirtus grafikų generavimui, bei leisti juos nesunkiai keisti.

4.1.2. Sistemos galimybės

Pagrindinės sistemos galimybės:

- studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos papildymas;
- pradinės studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos saugojimas;
- pradinės studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos užkrovimas;
- studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos keitimas;

grafikų generavimas lengvai redaguojamame Microsoft Exel formate.

4.2. Sistemos vadovas

4.2.1. Sistemos funkcijos

Pradinės informacijos valdymo funkcijos

UTPS sistema naudodamasi pradine informacija savarankiškai generuoja grafikus. Tam, kad pradinės informacijos nereiktų kiekvieną kartą suvedinėti iš naujo ji vykdo šias informacijos valdymo funkcijas:

- pradinės studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos saugojimas;

- pradinės studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos užkrovimas;
- studentų grupių, auditorijų, užsiėmimų bei dėstytojų informacijos keitimas.

Rezultatų kūrimo ir informacijos eksporto funkcijos

UTPS sistema yra skirta grafikų generavimui. Tam kad grafikus būtų lengva keisti, jie turi būti tinkamo formato, todėl pagrindinės sistemos funkcijos yra šios:

- Grafikų generavimas vidiniame duomenų formate;

Grafikų eksportavimas į išorinį excel formatą.

UTPS funkcijų naudojimas

Visos sistemos funkcijos yra valdomos per interfeiso panelę. Čia yra mygtukai, kurie realizuoja tam tikrą funkciją. Funkcijų sąryšis su mygtukais:

Mygtuko pavadinimas	Mygtuko funkcija
Toliau	Peršoką į sekantį programos lapą
Užkrauk pradinis duomenis	Užkrauna pradinis studentų grupių, auditorijų, užsiėmimų bei dėstytojų duomenis iš anksčiau sukurto projekto failo.
Išsaugok pradinis duomenis	Išsaugo pradinis studentų grupių, auditorijų, užsiėmimų bei dėstytojų duomenis į projekto failą.
Generuok	Generuoja tvarkaraštį Microsoft Exel formate.
Išsaugok tvarkaraštį	Išsaugo sugeneruotą tvarkaraštį.
Uždaryk	Baigia programos darbą.
Dadėti	Papildo atitinkamo programos lapo duomenis.
Ištrinti	Pašalina pažymėtus programos lape duomenis. Prieš naudojanti reikia pažymėti vieną eilutę duomenų lauke.

Pradinių duomenų suvedimas

Studentų grupės. Suvedame grupių pavadinimus bei studentų skaičių:

The screenshot shows the 'Upsiėimio Tvarkaraėio Generavimo Sistema' window with the 'Studentu grupės' tab selected. The interface includes a header with 'Auditorijos', 'Destytojai', and 'Ušiemimai' tabs. Below the header, there are input fields for 'Pavadinimas' and 'Studentų kiekis', along with 'Dadėti' and 'Ištinti' buttons. A table displays the entered data:

Pavadinimas	Studentu kiekis
IFM-9/1	20
IFM-9/2	24
IFM-9/3	25
IFM-9/5	15

At the bottom, there are buttons for 'Toliau', 'Upkrauk pradinius duomenis', 'Išsaugok pradinius duomenis', 'Generuok', 'Išsaugok tvarkaraėtą', and 'Uždaryk'.

Auditorijos. Suvedame auditorijų pavadinimus, adresus bei vietų skaičių:

The screenshot shows the 'Upsiėimio Tvarkaraėio Generavimo Sistema' window with the 'Auditorijos' tab selected. The interface includes a header with 'Studentu grupės', 'Auditorijos', 'Destytojai', and 'Ušiemimai' tabs. Below the header, there are input fields for 'Pavadinimas', 'Adresas', 'Tipas', and 'Vieta skaiėius', along with 'Dadėti' and 'Ištinti' buttons. A table displays the entered data:

Pavadinimas	Adresas	Tipas	Vieta skaiėius
508	studentu.g	Auditorija	28
204	studentu.g	Auditorija	28
521	studentu.g	Auditorija	28
512	studentu.g	Laboratorija	15
522	studentu.g	Auditorija	15
206	studentu.g SC	Laboratorija	15
105	studentu.g SC	Laboratorija	15
202	studentu.g	Laboratorija	15
320	studentu.g	Auditorija	28
212	studentu.g	Auditorija	28
301	studentu.g	Auditorija	28

At the bottom, there are buttons for 'Toliau', 'Upkrauk pradinius duomenis', 'Išsaugok pradinius duomenis', 'Generuok', 'Išsaugok tvarkaraėtą', and 'Uždaryk'.

Dėstytojai. Suvedamos dėstytojų pavardės bei jų darbo laikas:

Upsiēmimo Tvarkarađiø Generavimo Sistema

Studentų grupės Auditorijos **Dėstytojai** Upsiēmimai

Vardas, Pavardė Dirba val. per dieną

Vardas, Pavardė	Val. per dieną
R. Jasinevicius	8
K. Pocius	8
V. Stuikis	8
R. Seinauskas	8
A. Ostreika	8
S. Tamulevicius	8
R. Damasevic...	8
K. Plukas	8
A. Lenkevicius	8
S. gudas	8
H. Pranevicius	8
A. Venckausk...	8
A. Riskus	8
S. Maciulevicius	8
R. Seinauskas	8
B. Kilda	8
R. Plestys	8
N. Sarafiniene	8

Užsiēmimai. Suvedame užsiēmimų pavadinimus, jų modulius priskiriame grupę, dėstytojus, auditorijas bei paskaitų skaičių:

Upsiēmimas

Pavadinimas:

Kodas:

Studentų grupės: IFM-9/1
IFM-9/2
IFM-9/3
IFM-9/5

Dėstytojai: R. Jasinevicius
K. Pocius
V. Stuikis
R. Seinauskas

Auditorijos: 508
204
521
512
522

Kiekis (per dvi savaites):

Pastaba: pažymėkite tuos studentus ir tuos dėstytojus kurie dirbs vienoje auditorijoje. Pažymėkite tas auditorijas kuriose gali dirbti visi studentai ir dėstytojai kartu.

Gauname rezultatą:

Pavadinimas	Kodas	Grupės	Dėstytojai	Auditorijos	Kiekis per dvi s...
Taikomoji dis...	P170M001	IFM-9/3	K.Plukas	320	2
Algoritmai ir pr...	P175M002	IFM-9/3	H.Pranevicius	202	1
Kompiuterine...	P175M007	IFM-9/3	A.Lenkevicius	202	2
Duomenų ba...	T120M005	IFM-9/3	S. gudas	202	1

4.2.2. Sistemos klaidos ir jų šalinimas

Išorinės klaidos

UTPS programa yra padaryta „VB runtime 6“ bibliotekos pagrindu, todėl norint, kad ji veiktų kompiuteryje būti instaliuota ši biblioteka. Jeigu ji nėra suinstaliuota paleidžiant programą bus gaunamas pranešimas: „'A required .DLL file was not found', 'Runtime error' (caution, this error has lots of different causes) or 'Error: the file msvbvm60.dll could not be found.'“.

Vidinės klaidos

UTPS programoje vidinės klaidos gali būti sąlygotos kompiuterio gedimų, blogos kompiuterio konfigūracijos arba neteisingų duomenų. Vidinės klaidos gali būti šios:

Klaidos pranešimas	Kieno sąlygota	Kaip taisyti
Nepavyko nuskaityti pradinį duomenų	Kompiuterio gedimas, arba neteisinga konfigūracija	Patikrinti ar laikmenoje nėra klaidų ir ar turimi leidimai skaityti failą.
Neteisingas pradinį duomenų formatas	Neteisingi duomenys	Patikrinti duomenų failo struktūrą, arba iš naujo suvesti duomenis į programą rankiniu būdu
Nepavyko išsaugoti pradinį duomenų	Kompiuterio gedimas, arba neteisinga konfigūracija	Patikrinti ar laikmenoje nėra klaidų ir ar turimi leidimai rašyti failą.
Neteisingi grupės duomenys	Neteisingi duomenys	Patikrinti ar grupės duomenys įvesti teisingai
Neteisingi grupių duomenys	Neteisingi duomenys	Patikrinti ar grupių duomenys įvesti teisingai
Neteisingi auditorijų duomenys	Neteisingi duomenys	Patikrinti ar auditorijų duomenys įvesti teisingai

Neteisingi dėstytojų duomenys	Neteisingi duomenys	Patikrinti ar dėstytojų duomenys įvesti teisingai
Neteisingi užsiėmimų duomenys	Neteisingi duomenys	Patikrinti ar užsiėmimų duomenys įvesti teisingai
Neįmanoma sugeneruoti grafiko su esamais duomenimis	Neteisingi duomenys	Atsižvelgiant į UTPS sistemos apribojimus pataisyti duomenis
Neįmanoma inicializuoti OLE sąsajos	Kompiuterio gedimas, arba neteisinga konfigūracija	Įsitikinti, kad kompiuteris atitinka programinės įrangos reikalavimus, ir yra veikiantys OLE servisi.

4.3. Sistemos instaliavimo dokumentas

4.3.1. Sistemos instaliavimo vadovas Windows aplinkoje

Norint suinstaliuoti UTPS sistemą pirmą reikia patikrinti ar instaliuojamas kompiuteris atitinka programinės ir aparatūrinės įrangos reikalavimus (žr. Minimalūs sistemos reikalavimai).

Jeigu kompiuteris atitinka minimalius sistemos reikalavimus, instaliavimo procesas vykdomas taip:

- 1) Persijungiama į administratoriaus režimą.
- 2) „Program files“ arba atitinkamam programų kataloge sukuriama „UTPS“ pakatalogis.
- 3) Į „UTPS“ patalogį sukeliama visi pateikti programos failai.
- 4) UTPS naudotojams įkeliamos nuorodos į vykdomąjį „UTPS“ failą.
- 5) Suninstaliuojama „VB runtime 6“ biblioteka.
- 6) Perkraunamas kompiuteris
- 7) Patikrinamas programos veikimas paleidžiant „UTPS“ vykdomąjį failą.

4.3.2. Sistemos sandara

UTPS sistema sudaryta iš:

UTPS vykdomojo failo, pavadinimu UTPS.exe.

„VB runtime 6“ bibliotekos, kurios instaliacinis failas vadinasi „VB6.0.exe“. Šiame instaliaciniame faile yra supakuoti Visual Basic programoms reikalingos dll bibliotekos tokios, kaip: msvbvm60.dll, oleaut32.dll, olepro32.dll.

4.3.3. Minimalūs sistemos reikalavimai

Aparatūrinės įrangos reikalavimai:

- 1) Pentium III klasės arba panašus x86 architektūros procesorius.
- 2) Laisvos kieto disko vietos bent 5 mb.
- 3) Operatyviosios atminties bent 256 mb.

Programinės įrangos reikalavimai

- 1) Microsoft Windows 2000 arba panaši sistema.
- 2) Microsoft Office XP arba adekvatus Microsoft Office paketas su suinstaliuota Microsoft Exel programa.

4.4. Sistemos administratoriaus vadovas

4.4.1. Sistemos veikimas

Sistema bendrauja su Microsoft Exel programa per OLE sąsają. OLE sąsają realizuoja Microsoft Windows operacinė sistema, naudodama šias sistemines bibliotekas:

- ole2.dll
- ole2disp.dll
- ole2nls.dll
- ole32.dll
- oleacc.dll
- oleaccrc.dll
- oleaut32.dll
- olecli.dll
- olecli32.dll
- olecvt32.dll
- oledlg.dll
- oleprn.dll
- olepro32.dll
- olesvr.dll
- olesvr32.dll
- olethk32.dll

Daugiau informacijos apie OLE sąsajos veikimą galima rasti Microsoft puslapyje adresu:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclib/html/_mfc_OLE_Classes.asp

4.4.2. Sistemos palaikymo vadovas

Norint, kad UTPS sistema gerai veiktų reikia įsitikinti, kad kompiuterio aparatūrinė įranga nesugedusi ir jis teisingai sukonfiguruotas.

Esant kompiuterio arba konfigūracijos ar aparatūros klaidoms UTPS sistema vartotojui parodys klaidos pranešimą. Pagal klaidos pranešimą reikia nustatyti klaidos priežastį ir ją pašalinti (žr. „Sistemos klaidos ir jų šalinimas“).

Jeigu klaidos priežastis pašalinta, o programa iš naujo parodo klaidos pranešimus reikia kliento kompiuteryje atlikti programos darbo inspekciją. T. y. suinstaliuoti Microsoft Visual Basic 6.0 aplinką ir atsidarius UTPS pagrindinės programos kodą pažingsniui atlikti visus veiksmus sistemoje sukeliančius klaidos pranešimą. Taip galima gauti papildomos informacijos padedančios nustatyti klaidos šaltinį arba aptikti klaidą pačioje UTPS sistemoje. Jeigu randama UTPS sistemos klaidą, ją būtina ištaisyti ir visiems vartotojams suteikti naują, pataisytą programos versiją.

5. PRODUKTO KOKYBĖS ĮVERTINIMAS

5.1. Sistemos patikros rezultatai

Alfa testavimo stadijos metu buvo atlikta UTPS sistemos patikra pagal turimą specifikaciją. Sistemos patikra parodė, kad programinė įranga atitinka turimus reikalavimus ir kuriama teisingai.

5.2. Sistemos beta testavimo rezultatai

Beta testavimo stadijos metu produktas buvo pristatytas naudotojui, su tikslu išsiaiškinti ar atitinka vartotojo poreikius. Sistemos Atestavimas parodė, kad sistema atitinka svarbiausius vartotojo poreikius.

5.3. Sistemos patikimumo bandymo rezultatai

Baigiant Beta testavimo stadiją buvo atliktas UTPS sistemos patikimumo testavimas. Testavimo metu buvo bandoma nustatyti ar sistema yra pakankamai patikima eksploatacijai. Šio testavimo metu buvo išbandytos visos sistemos galimybės su įvairiais duomenimis. Surasti nesklandumai buvo pašalinti. Nustatyta, kad UTPS sistema pašalinus nesklandumus yra pakankamai patikima eksploatacijai.

6. IŠVADOS

1. Šio projekto metu buvo kuriama Užsiėmimų Tvarkaraščių Generavimo Sistema. UTPS skirta padėti vartotojui išspręsti tvarkaraščių sudarymo problemą.
2. UTPS vartotojas turi turėti tvarkaraščio sudedamuosius duomenis:
 - Studentų grupes ir jų kieki.
 - Paskaitų pavadinimus ir jų kieki.
 - Auditorijų kieki ir jų tipus, mokymosi vietų auditorijose kieki.
3. UTPS vartotojas taip pat privalo žinoti ryšius tarp duomenų:
 - Kokios auditorijos tinka kokiems užsiėmimams.
 - Kurios studentų grupės lanko kokius užsiėmimus.
 - Kokie dėstytojai moko kurias grupes.
4. Vartotojo patirtis tvarkaraščių kūrimo srityje gana didelė ir svarbiausias jo poreikis buvo gauti pagalbą kuriant grafikus, todėl buvo pasirinktas toks sistemos variantas, kai sistema iš turimų duomenų sugeneruoja grafiką lengvai redaguojamame formate, tam kad jį paskui būtų galima nesunkiai pakeisti.
5. Kūrimo procesas buvo pažingsniui dokumentuojamas. Tam buvo naudojamas Krioklio projekto valdymo modelis. Šis modelis pasižymi tuo, kad sekančio proceso kokybė labai priklauso nuo prieš tai buvusio, todėl buvo stengiamasi kuo geriau atlikti sistemos analizės ir projektavimo procesus.
6. Analizės proceso pradžioje buvo atliekama apibrėžimo, projektavimo, kūrimo, diegimo ir palaikymo fazių metodologijos ir technologijų analizė. Taip nustatyti kiekvienos fazės tikslai kurių buvo siekiama.
7. Sistemos apibrėžimo ir analizės fazės metu buvo sudaromas bendras būsimos sistemos vaizdas, bei nustatomos projektavimo gairės.
8. Sistemos projektavimo fazės pagrindinis tikslas – pritaikyti reikalavimams skirtingus techninius sprendimus.
9. Sistemos kūrimo metu buvo siekiama naudojantis pasirinktais techniniais sprendimais sukurti sistemą atitinkančią išanalizuotus reikalavimus.
10. Sistemos palaikymo fazė tęsiasi tol kol sistema yra naudojama ir moraliai nepasenusi. Išvadas apie sistemos palaikymo fazę galima daryti tik užbaigus IS gyvavimo ciklą.
11. Projekto rizikos buvo valdomos atskiriant svarbias ir nesvarbias rizikas. Projekto kūrimo pradžioje svarbiausios atrodė šios rizikos:
 - Neteisingai nustatyti UTPS užsakovo interesai.
 - Per didelio dėmesio skyrimas nereikšmingoms UTPS dalims.
 - Neteisingas UTPS projekto tikslo nustatymas.

- Neteisingas ar nepakankamas detalių supratimas.
 - Neišbaigtos sisteminių reikalavimų funkcionalumo specifikacijos.
 - Klaidos nustatant reikalingus resursus ir terminus.
 - Retas tikrinimas, ar suderinti tarpusavyje įvairūs etapai, ir kontrolės iš užsakovo pusės nebuvimas (užsakovas neįtraukiamas į sistemos kūrimą).
12. Projekto kūrimo proceso metu atsirado rizika, susijusi su klaidingų terminų nustatymu, kai sunkiausi ir skubūs darbai atliekami neturint laiko, o lengviems ir nesvarbiems skiriamas likęs laikas.
13. Kuriant UTPS Nebuvo stengiamasi sukurti idealios, be klaidų ir kupinos galimybių sistemos, tačiau buvo kuriama tokia sistema kuri atitiktų vartotojų poreikius.

Atlikus sistemos patikrą ir atestavimą buvo nustatyta, kad sukurta sistema atitinka surinktus reikalavimus, bei vartotojo poreikius, todėl galima spręsti, kad projektas įvykdytas sėkmingai. Galutines išvadas apie IS galima daryti tik užbaigus IS gyvavimo ciklą.

7. LITERATŪRA

1. Žalkauskas V. Šiuolaikinių kompiuterių programų ir tinklų žodynas, 2005.
 2. Bareika E., Krivickas J., Motiejūnas K., Keršienė V., Ambrazas A. Programinės įrangos projektų valdymas. 2003, p. 22-75.
 3. Highsmith C. Writing Effective Use Cases. 1999, p. 13-15.
 4. Boctor D. Microsoft Office 2000: Visual Basic for Applications Fundamentals. 2001, p. 233-244.
 5. Shank D., Roberts M., Myers T. Microsoft Office 2000 : Visual Basic programmers guide. 1999, p. 3-219.
 6. Robbins J. Debugging Applications. 2000, p. 6-8.
 7. C McConnell S. Software Project Survival Guide. 1997, p. 51-172.
 8. Schwalbe K. Information Technology Project Management. 2005, p. 4-6, 22-29, 53-58.
 9. Adams J.R. The principles of project management. 1997, p. 4-7, 38.
 10. Višniakas I., Slivinskas K. Objektų kokybės vertinimo ir patikimumo skaičiavimų metodikos nurodymai: Patikimumo teorija . 2005, p. 9-11
 11. Petzold C., Programirovanyje dlia Windows95. T. 1 (1997), p. 9-11.
- Panaudoti elektroniniai šaltiniai:**
12. Programavimas VBA kalba, Sokas A. Mokomoji knyga. 2005, p. 16-51. Prieiga per internetą: <http://www.vtu.lt/leidiniai/elektroniniai/metodiniai_vba.pdf>
 13. Barzdžiukienė L. D., Celiešienė V., Kaulakienė A. Baigiamasis studijų darbas. Kalbininkų patarimai: teorija ir tvarkyba. 2-asis pataisytas ir papildytas leidimas. 2005, p. 7-97. Prieiga per internetą: <<http://www.vtu.lt/leidiniai/elektroniniai/baigiamasis.pdf>>
 14. Tamošaitis R. Projektų vadybos metodiniai nurodymai. 2004, p. 5-6. Prieiga per internetą: <http://www.vtu.lt/leidiniai/elektroniniai/projektu_vad_met_nurodymai.pdf>
 15. Technikos biblioteka. Prieiga per internetą: <www.tb.lt/>
 16. Mokslinė elektroninė biblioteka. Prieiga per internetą: < www.elibrary.lt/>
 17. Bubl: interneto šaltinių katalogas įvairiomis akademinėmis temomis. Prieiga per internetą: <bubl.ac.uk/link/>
 18. Lib.ru: Mokovo M. Elektroninė biblioteka. Mokslo populiarī literatūra vadovėliai ir kt. Prieiga per internetą: <www.lib.ru/>
 19. Engineering e-library, Sweden. Prieiga per internetą: <<http://eels.lub.lu.se/>>

20. Mireba: mokslo informacijos elektroninė biblioteka. Prieiga per internetą:
<www.mireba.lt>

SUMMARY

That project's purpose is to create the Schedule Projection System.

There is given information about analogous Schedule Projection Systems in Lithuania and other countries. It was used "fall" model for that system's creating control. The program was created in Microsoft Visual Basic environment and it uses OLE interface for communicating with Microsoft Excel program. Schedule Projection System runs in any computer with x86 architecture, Microsoft Windows operating system and Microsoft Office pack.

Systems documentation:

- Projection methodology and technology analysis.
- Requirements specification.
- Detailed project's plan.

System's working documentation:

- System functional description.
- System manager.
- System installation document
- System administrator document

After program's revise and testing procedures the system was evaluated like specious for exploitation.