

Факултет инжењерских наука Универзитета у Крагујевцу

Милан С. Радмановац

**Пројектовање информационог система
спортског центра**

мастер рад

Крагујевац, 2014.

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Машинско инжењерство

Ниво студија: Мастер академске студије

Модул: Информатика у инжењерству

Предмет: Информациони системи и базе података

Број индекса: 358/2012

Милан С. Радмановац

**Пројектовање информационог система
спортског центра**

мастер рад

Комисија за преглед и одбрану:

1. Др Милан Ерић - ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____

У оквиру овог мастер рада кандидат треба да анализира информациони систем спортског центра, проучи животни циклус развоја и моделе развоја информационих система и применом развојне технологије примени техничка решења и практично прикаже један информациони систем уз опис функција система.

Препоручена литература:

- [1]. **Полишчук Е. Ј.(2007)** „Пројектовање информационих система“, Подгорица.
- [2]. **Милошевић Б.(2013)** „CASE алати“, Висока техничка школа струковних студија, Ниш.
- [3]. **Вељовић А.** „Развој информационих система и база података“.

Крагујевац, датум

ментор:

Проф. др Милан Ерић

Резиме рада:

У првом делу рада су описане информационе технологије и информациони системи. Следи детаљнији опис информационих система који обухвата компоненте информационих система, животни циклус развоја система, као и краћи опис модела развоја информационих система. Наведени су заступљенији алати за пројектовање информационих система (CASE алати) и укратко описани. У другом делу рада приказан је пример информационог система, објашњене су подформе и могућности система. Дат је краћи опис развојног окружења система.

Кључне речи:

Информационе технологије, информациони системи, фазе развоја система, модели развоја информационих система, CASE алати.

Summary of work:

The first part of the work describes the information technology and information systems. The following is a detailed description of the information systems that include components of information systems, system development life cycle, as well as a short description of the model of development of information systems. The below are listed prevalent tools for the design of the information systems (CASE tools) and briefly described. The second part of the work shows an example of an information system, subforms and system capabilities are explained. Brief description of the development environment of the system is given.

Key words:

Information technology, information systems, phases of system development, development models of information systems, CASE tools.

Садржај:

1	Информационе технологије	3
1.1	Историјски развој информационих технологија	4
1.2	Рачунарски хардвер	5
1.3	Рачунарски софтвер	6
2	Информациони системи	10
2.1	Појмовно одређење информационих система	10
2.2	Компоненте информационих система	13
2.2.1	Хардвер	13
2.2.2	Софтвер	13
2.2.3	Базе података	14
2.2.4	Рачунарске мреже	14
2.2.5	Људски ресурси и процедуре	15
2.3	Животни циклус развоја система	15
2.3.1	Фазе развоја информационог система	17
2.3.2	Циљеви и проблеми развоја информационих система	18
2.3.2.1	Узроци неуспеха пројеката информационих система	19
2.3.2.2	Побољшања успешности информационих система	19
3	Модел развоја информационих система	20
3.1	Водопад (секвенцијални) модел	20
3.1.1	Фазе водопад модела	21
3.1.1.1	Предности водопад модела	22
3.1.1.2	Недостаци водопад модела	22
3.2	Verification & Validation Model - „V“ модел	24
3.2.1	Шта је верификација?	25
3.2.2	Шта је валидација?	25
3.3	Прототипски модел	27
3.4	Еволутивни модел	29
3.5	Инкрементални модел	30
3.6	Спирални модел - Нови приступ ка развоју софтвера	31
3.7	Поређење између водопад модела и спиралног модела	33
3.7.1	Водопад - waterfall модел	33
3.7.1.1	Предности	34
3.7.1.2	Мане	34
3.7.2	Спирални модел	34
3.7.2.1	Предности	35
3.7.2.2	Мане	35
3.7.3	Разлика између водопад модела и спиралног модела	35
4	Алати за пројектовање информационих система (CASE алати)	37
4.1	Дефиниције CASE алата	38

4.2	Елементи CASE алата.....	39
4.2.1	Врсте CASE алата	40
4.3	Могућност CASE алата за моделирање пословних података	40
4.3.1	ORACLE Designer	41
4.3.2	Entity Relationship Diagrammer (ERD)	42
4.3.3	BPwin (Bussines Process Windows)	43
4.3.3.1	IDEF0	44
4.3.3.1.1	IDEF0 приступ	44
4.3.4	ERwin	45
4.3.4.1	IDEF1X	46
4.3.4.1.1	Преглед	46
4.3.5	MS visio	47
4.4	Развојно окружење информационих система - Microsoft Access	48
4.4.1	Примена	49
4.4.2	Карактеристике.....	50
4.4.3	Microsoft Access Runtime - извршавање	51
4.4.4	Развој.....	52
4.4.5	Функције и употреба Access-а	53
4.4.5.1	Предности Access-а	53
4.4.5.2	Посебне особине Access-а.....	53
4.4.6	Начин рада Access-а.....	54
5	Опис реалног система и апликације	57
5.1	Надлежности и послови реалног система	57
5.1.1	Организациона структура	57
5.1.2	Спортска хала.....	58
5.1.4	Структурна систем анализа пословног система и његова декомпозиција	59
5.1.5	ER дијаграм	62
5.2	Речник података спортске хале	62
5.3	Опис апликације	63
6	Закључак	73
7	Литература	74

1 Информационе технологије

Информационе технологије (IT) (енгл. Information technology) се, од стране Америчке асоцијације за информационе технологије, дефинише као "изучавање, дизајн, развој, имплементација и подршка или управљање рачунарским информационим системима, софтверским апликацијама и хардвером". Информационе технологије користе рачунаре и рачунарске програме да претворе, ускладиште (сместе), штите, обраде, безбедно шаљу и примају информације [1].

Термин „информационе технологије” често обухвата и знатно шире поље области технологије. Све оне активности којима се професионалци информационих технологија баве, од инсталација апликативних програма до пројектовања сложених рачунарских мрежа и информационих система. Неке од тих активности су: умрежавање и инжењеринг рачунарског хардвера, дизајнирање софтвера и база података, као и управљање и администрација информационим системом.

Информационе технологије је општи термин који описује технологију која помаже производњи, манипулацији, складиштењу, комуникацији и дистрибуцији информација.

Први који је употребио термин „Информационе технологије” био је Jim Domsik из Michigan-а и то новембра 1981. године [2]. Термин је употребио како би модернизовао до тада коришћени израз Data processing (обрада података). У то време Domsik је радио као рачунарски менаџер у аутомобилској индустрији.

Стандарде у овој области дефинисале су организације АВЕТ¹ и ACM² [2].

У својој основи, информационе технологије се баве побољшањем различитих људских напора у решавању проблема кроз пројектовање, развој и коришћење технолошки заснованих система и процеса који побољшавају ефикасност и ефективност информација и релевантних знања у различитим стратешким, тактичким и оперативним ситуацијама. У идеалном случају, то се постиже кроз обраћање пажње на информационе потребе људи у решавању проблема и задатака, у пружању технолошких помагала, укључујући рачунарске системе засноване на хардверу и софтверу и процесе у вези, који помажу у овим задацима. Активности информационе технологије допуњују и побољшавају,

¹ Accreditation Board for Engineering and Technology, регистрована као акредитационо тело за инжењеринг и технологију, је невладина организација која акредитује едукационе програме у примењеним наукама, рачунарству, инжењеринг и технологији.

² The Association for Computing Machinery је америчка организација за међународно научно друштво које се бави рачунарством.

али такође превазилазе границе традиционалног инжењеринга, стављајући нагласак на информационе основе инжењерских активности, насупрот заснивању на физичким наукама у традиционалним инжењерским подухватима.

Информационе технологије се састоје од хардвера и софтвера који омогућава прибављање, представљање, складиштење, пренос и коришћење информација. Успешност у информационим технологијама зависи од способности да се носимо са укупном архитектуром система, његових интерфејса са људима и организацијама и веза са спољашњим окружењима. Такође је критична његова способност да информације претвара у знање.

Почетни напори у пружању система заснованих на информационим технологијама, били су у вези са имплементацијом и коришћењем нових технологија за подршку канцеларијским пословима. Они су еволуирали од електричних писаћих машина и електронских рачуноводствених система ка напредном технолошком хардверу као што је факсна машина и лични рачунар који обављају важне функције међу којима су електронска обрада списка, рачуноводство и обрада текста. Данашњи главни аспект информационих технологија тиче се умрежавања и интернета.

1.1 Историјски развој информационих технологија

У раним данима људске цивилизације, развој је био могућ једино кроз људски напор односно рад. Људска способност за коришћење природних ресурса, довела је до тога да се човечанство развија не само кроз свој сопствени напор и рад, већ и кроз употребу доступних природних ресурса. За класичан пример у економији обично се узима земљиште као појам који подразумева природне физичке ресурсе. У то време, највећи број људских друштава се заснивао на малим земљопоседницима. Доступност финансијског капитала током индустријске револуције увела је трећи основни економски ресурс и омогућила развој хијерархијских корпорација. Овај период је углавном повезан са централизацијом, масовном производњом производа и стандардизацијом.

У другом делу индустријске револуције откривена је електрична енергија а нешто касније и полупроводници. Ова открића су довела до информатичког, или доба информационих технологија [2]. Међу многобројним потенцијално критичним алатима базираним на информационим технологијама срећемо: машине база података, е-пошту, алатке вештачке интелигенције, уређаје за пренос факсимила (факс машине), програмске језике четврте генерације,

локалне мреже (LAN³), дигиталне мреже са интегрисаним сервисима (ISDN⁴), кабловске и ADSL⁵ мреже за приступ интернету, уређаје и дискове за оптичко складиштење података (CD-ROM, DVD и сл.), личне рачунаре, паметне (напредније) мобилне телефоне, ручне и таблет рачунаре, алгоритме за паралелну обраду, софтвер за обраду текста, софтверске пакете за инжењеринг помоћу рачунара, пакете рачуноводственог софтвера итд. Скоро све што подржава прикупљање, представљање, пренос и коришћење информација, може се назвати производом информационих технологија.

1.2 Рачунарски хардвер

Рачунарски хардвер обухвата све уређаје и рачунарске компоненте које су неопходни за рад рачунарских система [3]. Класификација рачунарских система може бити по више основа:

- Према начину рада и приказивању података:
 - Аналогни,
 - Дигитални,
- Према величини и снази:
 - Суперрачунар,
 - Велики рачунар (mainframe),
 - Минирачунар,
 - Радне станице,
 - Микрорачунар.

Прва подела рачунарског хардвера односи се на начин рада и приказивања података. Код аналогних рачунара користе се електричне шеме у којима се излазни резултати добијају у виду неке аналогне електричне величине (напони, јачина струје и сл.), док се код дигиталних рачунара користи бинарни бројни систем и резултат обраде се приказује у дигиталном облику (број, слово, знак).

Друга подела рачунарског хардвера је према величини и снази. Разликују се:

- **Суперрачунари** на бази нових технологија паралелног процесирања, хлађења и примене вештачке интелигенције;
- **Велики рачунари (mainframe)** који се користе као централни рачунари за извођење обимних и сложених информационих захвата, са могућношћу прикључивања великог броја корисника;

³ Или локална рачунарска мрежа је скуп рачунара који су повезани у једну рачунарску мрежу, на релативно малом простору, као што су канцеларија, више канцеларија или зграда.

⁴ Integrated Services Digital Network је ознака је за дигиталну телефонску технологију.

⁵ Асиметрична дигитална претплатничка линија (енгл. ADSL - Asymmetric Digital Subscriber Line) је асиметрична у смислу брзине преноса података тј. могућност бржег преноса података у download-у (ка кориснику), него што је то у upload-у пренос података од корисника ка мрежи.

- **Минирачунари** намењени за мање изражене информационе захтеве. Карактеристике мини рачунар се мањају великом брзином, тако да они по својим карактеристикама “прекривају” и део великих рачунара, радних станица, микрорачунара;
- **Радне станице** намењене за специјалне задатке, нпр. графичке радне станице, станице за мултимедије;
- **Микрорачунари** намењени за непосредног корисника, који могу бити изведени као терминали, РС, notebook, PLC (Programmable logic controller)⁶ и сл. Њихове карактеристике се стално усавршавају.

1.3 Рачунарски софтвер

Софтвер представља скуп инструкција или програма на основу којих хардвер извршава одређене задатке (обрада података, улаз, излаз итд.). Док је хардвер предвиђен за опште намене, софтвер се може односити на специфичне намене [3]. Софтвер се дели у две групе и то:

- Системски софтвер и
- Апликативни софтвер.

Системски софтвер је скуп програма који контролишу и подржавају рад рачунарског система. Он је општијег типа у односу на апликативни софтвер и обично независан од типа апликативног софтвера. Његов примарни задатак је да подржи рад апликативног софтвера тако што управља основним функцијама рачунара. Системски софтвер се групише на следећи начин:

- Софтвер за управљање системом (оперативни систем) је главна компонента системског софтвера. Сам назив указује да овај софтвер надгледа извршавање свих операција рачунара, укључујући праћење рада рачунара, рад у случају прекида рада програма, контролисање процеса на улазу и излазу из рачунара и др. Једна од најважнијих карактеристика софтвера за управљање системом је портабилност, способност оперативних система да раде на различитим типовима рачунара;
- Софтвер за подршку систему обухвата:
 - *Помоћне програме* који омогућавају реализацију помоћних активности као што су сортирање, копирање или трајно чување (back up);
 - *Програми за праћење перформанси* прате одвијање послова на рачунарском систему. При томе се аутоматски формирају извештаји са статистиком коришћења информационих ресурса. Посебно су важни извештаји о процесорском времену, меморијском простору,

⁶ Дигитални рачунар који се обично користи за аутоматизацију индустријских електропроцеса, као што су контрола машина код фабричке линије за монтажу забавне возње.

улазно/излазним уређајима и апликативном софтверу. Ови извештаји се користе ради анализе рачунарског система и посебно информационог система, а у циљу планирања ефикаснијег рада у будућности;

- *Програми за праћење безбедности система* омогућавају заштиту информационих ресурса од неовлашћеног (неауторизованог) приступа или деструкције, што се остварује на различите начине (идентификациони кôд или лозинка). Ови програми такође контролишу рад информационих ресурса и не дозвољавају деструкцију информационих система;
- Софтвер за развој система омогућава превођење програма написаних у изворном кôду у објектни или машински кôд. То се постиже применом компајлера (преводилица) или интерпретера, који су најчешће програмски системи за развој рачунарског система.

Апликативни софтвер се састоји из инструкција (наредби) које управљају радом рачунара и односе се на одређену област или апликацију, по чему је ова група добила назив. Апликативни софтвер може бити у виду:

- Електронских табела,
- Управљања подацима,
- Обраде текста,
- Графике,
- Мултимедија,
- Комуникација.

Електронске (spreadsheet) табеле представљају једноставан тип апликативног софтвера у коме корисник (прогрмер) уноси у поља табеле одређене нумеричке или текстуалне податке и математичке релације између редова и колона табеле. Ове релације се дефинишу помоћу макро наредби, које могу укључити аритметичке операције, статистичке функције, логичке операције и текстуалне операције. Софтер овог типа погодан је као симулациони софтвер и у сличају честих измена релација између елемената (редова и колона) у електронској табели. Најчешће коришћени софтверски пакет ове врсте је Excel.

Апликативни софтвер базиран на **управљању подацима** је карактеристичан за апликативни софтвер јер се управљањем подацима овим софтвером подржава меморисање, претраживање и мнипулација подацима. Разликују се два основна типа управљања подацима:

- Систем датотека и
- Систем база података.

Датотека је скуп повезаних слогова организованих на исти начин (алфабетски, хронолошки или хијерархијски). Недостатак система датотека и софтвера базираног на њима је понављање, издвојеност и неконзистентност података. Софтвер базиран на **базама података** је савременији јер су подаци у њима интегрисани заједно са унапред познатим релацијама. Тиме је умањено понављање (редундатност) података, али се зато појављује нови проблем - сигурност података и сложеност организације база података. И поред ових проблема софтвер базиран на базама података је присутнији због смањења редундансе, могућности дељења (дистрибуције) података и повећања интегритета података. Типични софтвери базирани на базама података су ORACLE, Access, MySQL и други.

Обрада текста се одвија у неколико фаза:

- *Унос и уређивање текста* који чине: куцање текста, исправљање грешака, копирање, премештање и брисање делова текста, претраживање и замена делова текста;
- *Обликовање (форматирање) текста* се састоји од следећих фаза: обликовање слова, обликовање пасуса, обликовање странице;
- *Штампање текста*.

Данашњи програми за обраду текста имају могућност приказа текстуалног материјала на екрану тачно онако како ће тај материјал изгледати када се одштампа. Пример оваквог програма је Microsoft Word.

Програм за **рад са графиком** омогућава кориснику да креира, складишти и приказује или штампа разне цртеже или графике. Пример оваквог програма је Corel Draw.

Основне категорије графичког софтвера су:

- *Презентација графика* која дозвољава корисницима да направе псеудо-тродимензионалне слике, суперпониране вишеструке слике, високоосветљене одређене аспекте на цртежу и направе цртеж слободном руком. Софтвери за презентацију графике типично садрже алате за цртање, презентацијске темплејте, различите стилове фонтова, алате за састављање вишеструких слика у комплентну презентацију;
- *Инжењеринг графика* – различити софтверски програми за инжењеринг су на располагању у скраћеном времену развијања апликација и повећане продуктивности особа за скицирање и инжењеринг;
- *Анализирање графика* – ове апликације додатно обезбеђују способност приказивања претходно анализираних података, као што су статистички подаци, у графичким форматима као стубасти графикони, линијски графикони, пита графикони и расути дијаграми. Графикони могу укључити елементе различитих текстура, ознака и заглавља.

Мултимедија се односи на садржај који користи комбинацију различитих облика садржаја. Ово је у супротности са медијима који користе само најосновније компјутерских екрана, као што су само текст или традиционална форма штампаног или ручно произведених материјала. Мултимедија укључује комбинацију текста, звука, слике, анимације, видеа или садржај интерактивног облика.

Комуникација - Email или е-пошта (електронска пошта) је могућност слања порука путем Интернета и једна је од најчешћих врста комуникације данас на Интернету. Нажалост, непожељни феномени као што су спам и рачунарски вируси, довели су до тога да је коришћење ове практичне и јефтине врсте комуницирања отежано.

2 Информациони системи

Информациони систем (IS) (енгл. Information system) је интегрисани скуп компоненети за сакупљање, снимање, чување, обраду и преношење информација [4].

Пословна предузећа зависе од информационих система за управљање својим операцијама и деловањима, одржавање конкурентности на тржишту, понуду различитих услуга и унапређење личних способности и капацитета. На пример, модерне корпорације зависе од рачунарских информационих система да би подржале своје финансијске рачуне и пословне трансакције, управљале људским ресурсима; општинске управе зависе од информационих система за понуду основних услуга својим грађанима; појединци користе информационе системе да би унапређивали своја знања, за куповину, управљање банковним рачунима и трансакцијама, као и за различите финансијске операције.

Изумом и доступношћу нових информационих технологија, јављају се нове могућности. Пошто су информациони системи омогућили различите људске активности, самим тим су извршили утицај на друштво. Убрзали су обављање свакодневних активности, утицали на структуру организација, изменили начине понуде и потражње производа на тржишту, као и начине и схватање рада. Информације и знање, данас представљају витални економски ресурс.

Историја система поклапа се са историјом рачунарских наука која је почела много пре него што се модерна дисциплина компјутерске науке појавила у двадесетом веку. Што се тиче протока информација и идеја, бројни застарели информациони системи и данас постоје који се стално ажурирају да промовишу етнографске приступе, да се обезбеди интегритет података, као и да се унапреде друштвене ефективности & ефикасности целог процеса. Генерално, информациони системи су фокусирани на обраду информација у оквиру организације, посебно у привредним друштвима, и дељење користи са модерним друштвом.

2.1 Појмовно одређење информационих система

Врло тешко је одредити, посебно данас, шта је најбитније познавати из области информационих система за менаџере и све друге кориснике. За све облике организовања предузећа, општине, школе, државе је неопходно добро познавати информационих система да би они могли да опстану и напредују. Информациони системи могу помоћи производним компанијама да своје пословање једноставно одвијају и на удаљеним подручјима, да понуде нове

производе и сервисе, да изнова обликују послове и радне токове, и да пронађу нове начине вођења послова.

Информација представља скуп података са одређеним значењем и представља поруку у било ком облику. Улога информације је да смањи неизвесност у оквиру неке критиче области, односно информација постаје корисна и вредна стављањем у одређени контекст [8].

Податак је у основи порука која се може и не мора искористити. Ако постоји и најмања вероватноћа да се порука једнозначно и тачно искористи, те представља неоспориву чињеницу, тада представља информацију. Својства објекта и њихових односа у простору и времену изражавамо подацима [24].

Информација је интерпретација податка - прочишћен, организован и обрађен податак у смисленом контексту. Информација је субјективног значења, у контексту примаоца, а знање се гради на основу нових информација које се надовезују на постојеће знање. Исти подаци могу бити различито интерпретирани од стране различитих људи у зависности о њиховом знању [5].

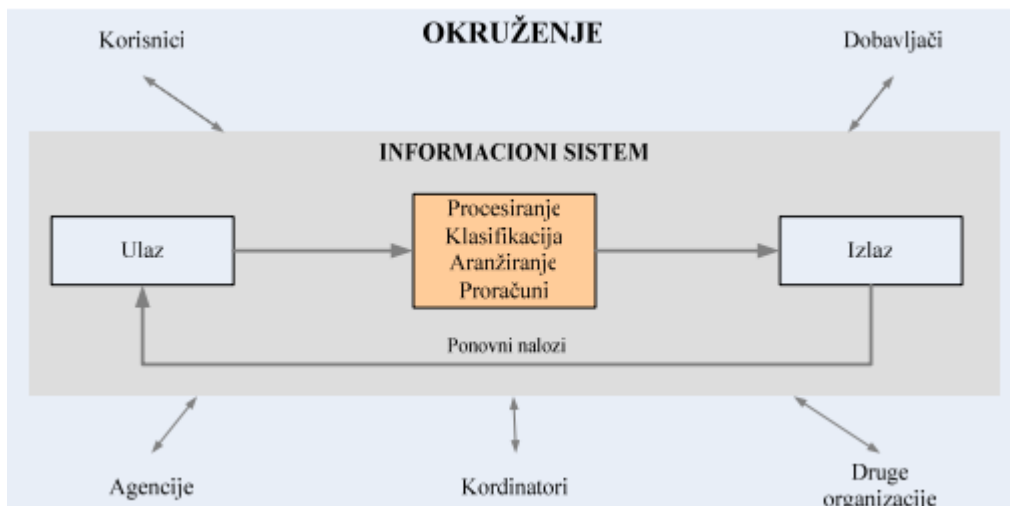
Кратак пример различитости информације и податка може бити следећи. Супермаркетски бројач за проверавање прикупља милионе података, као што су идентификациони број производа или цена сваког појединачно продатог производа. Овакви делови података могу бити тотализирани и анализирани да дају потпуну информацију, као што је укупни број боца детерџента за судове који је продат у одређеној продавници, које марке детерџента се најбрже продају у тој продавници или у том подручју или која је укупна свота потрошена на ту врсту детерџента у продавници или читавом региону.

Систем је уређени поредак међузависних компоненти повезаних заједно према неком плану за постизање одређеног циља.

Компоненте система јесу његови физички делови, улази и излази као и процеси, њихове конверзије и управљачки поступци (планирање, организација, управљање и надзор).

Елементи система:

- **Подсистеми:** компоненте које припадају систему;
- **Граница:** дефинише обим и домашај система;
- **Околина:** све што је ван граница система, али се још увек тиче система;
- **Улази (Input):** елементи који улазе у систем из околине;
- **Излази (Output):** елементи који напуштају систем;
- **Окружења:** везе између система и његове околине (заштита, филтери);
- **Ограничења:** унутрашњи и спољни фактори који одређују и ограничавају функционисање система.



Слика 1. Функције у информационом систему

Карактеристике система:

- **Организација** - структура и поредак, хијерархијске везе које одређују формалну комуникацију и управљачки ланац (нпр. војска, предузеће);
- **Интеракција** - начин на који поједине компоненте сарађују са другим компонентама (нпр. набавка с производњом, производња са продајом);
- **Међузависност** - један подсистем зависи о другом (улаз), да би могао функционисати;
- **Интегрисаност** - мера повезаности компоненти.

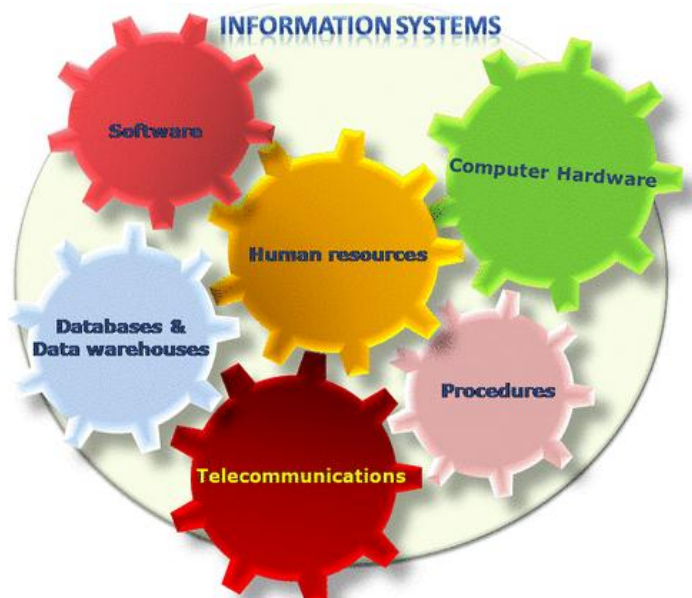
Информациони систем у техничком погледу може се дефинисати као скуп међусобно повезаних компоненти које скупљају, прерађују, бележе и дистрибуирају податке како би помогле доношење одлука, координацију и контролу у организацији. Информациони системи служе и као помоћ менаџерима и радницима за анализу разних проблема, као и веома корисна помоћ код израде нових производа. Такође информациони систем садржи неопходне информације о људима, местима и стварима унутар организације или спољног окружења.

Сваки информациони систем има три активности које су неопходне да корисник информационог система донесе одлуке, да контролише операције, анализира проблеме, и да креира нове производе или да даје услуге. Ове активности су улаз, обрада и излаз.

Улаз скупља необрађене податке унутар организације или из окружења. Обрада је процес која ове податке обрађује и доводи их у разумљиву форму. Излаз преноси процесиране тј. обрађене податке људима који ће их користити или активностима за које ће бити од користи.

2.2 Компоненте информационих система

Основне компоненте информационих система су хардвер и софтвер рачунара, базе података, системи рачунарских мрежа, људски ресурси и процедуре, односно методологије процесирања и преношења информација [4].



Слика 2. Компоненте информационог система

2.2.1 Хардвер

Данас, чак и најмања фирма, поседује или изнајмљује рачунаре. Обично се ради о персоналним рачунарима (PC). Веће организације користе више рачунарских система, почев од моћних радних станица, минирачунара па до ефикасно умрежених персоналних рачунара. Уједно са периферним уређајима, као што су улазни и излазни уређаји (монитори и штампачи, на пример), и рачунарске мреже, чине хардвер информационих система. У новије време, цене хардвера убрзано опадају, али брзина обраде и капацитет уређаја за чување података, се константно увећава.

2.2.2 Софтвер

Софтвер се обично дели на две шире схваћене класе: системски или оперативни софтвер и софтвер апликације. Основни системски софтвер се препознаје под именом оперативни систем. Оперативни систем управља хардвером, датотекама и другим ресурсима система, обезбеђујући систематско, конзистентно обављање задатака и контролу рачунара, најчешће преко графичког корисничког интерфејса. Софтвер апликације су програми

дизајнирани за обављање специјализованих задатака од којих се многи нуде на тржишту као пакети припремљени за коришћење одмах после инсталирања.

Апликациони софтвер обухвата програме који раде неки посао за корисника, као што су обрада текста, обрада података у електронским табелама и системи за управљање базама података

2.2.3 Базе података

Многи информациони системи су примарно управљачи база података. База података је колекција међусобно повезаних података, организованих на најпогоднији начин за коришћење, или вађење података по задатим критеријумима. Типичан пример база података у једном предузећу су подаци о радницима и каталози производа. Нарочиту вредност поседују базе о купцима, које се користе за дизајн и маркетинг нових производа. Било ко је икада купио нешто са кредитном картицом, директно у продавници, преко електронске поште, или на интернету, постаје део неке од многобројних база о купцима.

2.2.4 Рачунарске мреже

Рачунарске мреже се користе за повезивање, или умрежавање рачунарских система, као и за преношење свих врста информација и података. Жичне технологије користе коаксијални кабл и оптичка влакна, док бежичне мреже су засноване на преносу путем микро таласа и радио таласа. Могуће је подесити више различитих рачунарских конфигурација, зависно од потреба сваке организације. На основу мрежног покрића, могу се класификовати следеће мреже:

- Локалне мреже (LAN) повезују рачунаре одређеног места, као што су пословне зграде или академски кампус [6]. Локална мрежа је мрежа велике брзине преноса података која покрива релативно малу географску област. Обично повезује радне станице, персоналне рачунаре, штампаче, сервере и друге уређаје. Локална мрежа омогућава корисницима рачунара да деле приступ уређаја и апликација, размене фајлова између повезаних корисника, као и електронске поште;
- Градска мрежа (Metropolitan area network (MAN)) покрива већу географску област од локалне мреже, у распону од неколико блокова зграда до читавих градова. Географски опсег спада између мреже широког опсега и локалне мреже. Градска мрежа пружа интернет конекцију за локалну мрежу у градском региону, и повезивање са ширим мрежама као што је интернет;

- Мреже широког опсега (WAN⁷) повезују рачунаре који се налазе на различитим местима и често из различитих организација. Обухвата градове, земље користећи телефонске линије и сателитске везе.

Интернет је мрежа која повезује милионе рачунара са сваког континента. Преко умрежавања, корисници персоналних рачунара добијају приступ различитим ресурсима информација, великим базама података и људским ресурсима, могућност обављања истих послова у сарадњи са колегама, као и контакт са људима који деле иста професионална и приватна интересовања.

2.2.5 Људски ресурси и процедуре

Квалификовани стручњаци су витална компонента сваког информационог система. Техничко особље се састоји од развојних и менаџера операција, аналитичара и дизајнера система, програмера софтвера и администратора система. Као додатак, радници у једној организацији морају бити оспособљени за коришћење капацитета информационог система. Стотине милиона људи из целог света свесно или несвесно уче да користе информационе системе користећи интернет.

Процедуре за коришћење, управљање и одржавање информационих система су део њихове документације. На пример, неопходно је дефинисати процедуре за стартовање програма за наредбу исплата зарада, укључујући процедуре за тренутно покретање, ко има потребну дозволу да га покрене, као и ко ће имати приступ излазним информацијама.

2.3 Животни циклус развоја система

Назив животни циклус развоја зависи од контекста у коме је употребљен. Могу се издвојити следећи називи животног циклуса: животни циклус развоја информационих система, животни циклус развоја софтвера и животни циклус развоја апликација. Праћење животног циклуса обезбеђује конзистентан и стандардизован начин развоја информационих система. Сврха праћења животног циклуса развоја омогућава правилно планирање, извршавање и надзор развојног пројекта [7].

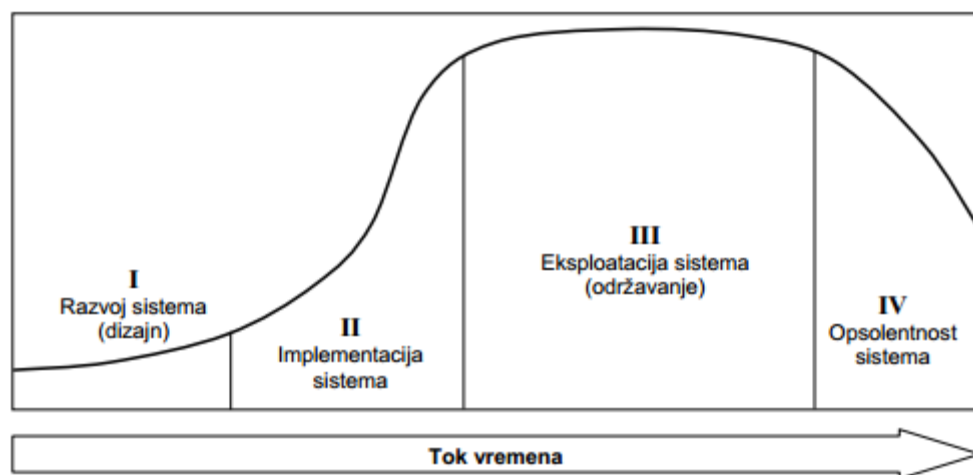
Животни циклус дефинише фазе и задатке (активности), које нужно треба обавити током развоја, без обзира на величину система који се гради. Свака поједина активност производи скуп резултата. Циклус осигурава "контролне тачке" за праћење напретка, процену постигнутих резултата и доношење одлука о даљим корацима. Пројекат пролази кроз фазе животног циклуса.

⁷ Wide Area Network је рачунарска мрежа која покрива шире подручје - града, региона или државе.

За животни циклус развоја софтвера могу се дефинисати следеће фазе и задаци:

- Потреба анализе и дизајна (Requirements Analysis & Specification),
- Концептуални/системски дизајн (Conceptual/System Design),
- Детаљни/програмски дизајн (Detailed/Program Design),
- Имплементација/кôдирање (Implementation/Coding),
- Појединачно и интегрално тестирање (Unit & Integration Testing),
- Системско тестирање (System Testing),
- Предаја система (System Delivery).

На слици 3. приказан је животни циклус информационих система, (Information Systems Life Cycle – ISLC) [8]. Тако животни циклус информационих система укључује фазу развоја система - дизајн, затим фазу имплементације система, фазу његовог рада и експлоатације и коначно, фазу застарелости система - опсолентност система. Системска проширења или надградња представљају главне модификације код неког постојећег система.



Слика 3. Животни циклус информационих система

Процедурални кораци у оквиру животног циклуса развоја система, постоје и дефинисани су и за темељне модификације и за комплетну замену система. Разлика која постоји између основних модификација система и минорних модификација се заснива на томе да под првим видом модификација подразумевамо додавање сасвим нових и дотад непостојећих функција, док под минорним модификацијама се подразумевају активности редовног одржавања.

Животни циклус информационих система, укључује следеће промене код постојећих информационих система, као што су: замена - нови систем, надградња - додавање нових функционалних перформанси и темељне модификације - замене неких делова система.

2.3.1 Фазе развоја информационог система

За животни циклус развоја информационог система (слика 4.) потребно је дефинисати много комплексније фазе.



Слика 4. Фазе развоја информационог система

Планирање развоја информационог система почиње снимањем стања (иницијална стратегија) [7]. Доприноси одређивању пословних циљева, идентификовању проблема и идеја, одређивању начина њиховог решавања, дефинисању захтева који се постављају пред систем. Садржи студију изводљивости, односно прегледну анализу проблемског подручја и одређивање (граница) пројекта. Планирање пројекта се састоји од израде плана рада, одређивања кадрова за пројекат, управљање и надзор пројекта. Пословни циљ је израда главног пројекта информатизације.

Анализа захтева је детаљна анализа којом се прецизирају границе пројекта и пословни захтеви. Врши се детаљна анализа постојећег система, проблема и пословних захтева. Спецификација захтева је детаљни опис захтева за информациони систем, обликован тако да га разумеју дизајнери. Представља модел будућег система.

Обликовање система, односно дизајн (моделирање) информаионог система, представља поглед пројектанта на будући информациони систем. Служи за доношење одлуке о томе како градити систем. Садржи дизајн потребних решења. Постоје решења која не треба дизајнирати. Детаљни дизајн представља разраду решења, односно израду технолошког модела информационог система (поглед извођача). Потребно је извршити дизајн архитектуре, интерфејса, складиштење података и програма, другим речима извршити техничку спецификацију система.

Израда (имплементација) подразумева уградњу базе података (BP), кóдирање процеса (функција) новог информационог система, а врши се након одабира рачунарске платформе. Тестирање је провера уграђених компоненти. Интеграција и провера система је удруживање делова и провера целине, да би се доказало да систем ради (да је исправно направљен), да ради оно што је захтевано (да је направљен прави систем који испуњава захтеве корисника).

Увођење у примену система представља преношење радних активности и конверзија података са старог на нови систем. Одржавање се састоји од измена система ради побољшања његових радних карактеристика (перформанси), побољшања или прилагођавања начина употребе. Одржавање подразумева и подршку добављача опреме, помоћ техничког особља корисницима информационог система у току његове употребе, као и израду плана одржавања.

Нови развојни циклус се спроводи након преиспитивања читавог система и констатације да су потребне веће измене услед промена у пословању или промена пословних циљева. Нови развојни циклус, најчешће, представља нови пројекат.

2.3.2 Циљеви и проблеми развоја информационих система

Основни циљ развоја информационог система је да се изгради систем који ради и који је поуздан, унутар задатих граница. То подразумева да се изгради систем који задовољава пословне циљеве, према захтевима корисника, у прихватљивом времену и по оправданој цени. Неки од проблема који се могу појавити приликом развоја информационог система су: прекорачење планираног времена и финансијских средстава, неиспуњавање захтева, односно неодговарајући систем, непоузданост, несигурност, нееластичност информационог система у примени, као и тешкоће у одржавању информационог система. Око 70% информационих система у свету се сматра неуспешним [7].

2.3.2.1 Узроци неуспеха пројекта информационих система

Разлози неуспешних пројекта информационих система су различити. Могу се издвојити следећи разлози: сложеност апликација, недостатак усмерености ка кориснику, занемаривање окружења организације, претерани оптимизам, изостанак праћења напретка и недостатак комуникације између корисника и извођача.

У нашем окружењу узроци неуспеха се, углавном, не истражују, а информације о неуспешним пројектима нерадо се објављују. Међу најчешћим узроцима може се претпоставити да су: лоша организација и вођење пројекта, ослонац на спољне пројектанте и саветнике, делегирано управљање пројектима, нереално планирање, формално извештавање о напретку, формални надзор над пројектом, као и потцењена улога сопствених стручњака. Не треба искључити и следеће узроке неуспеха: лоша израда пројекта, неодговарајућа анализа система, грешке у дизајну и контроли квалитета, употреба CASE алата и погрешно коришћење. Многи системи су пропали, или су били одбачени, јер су се извођачи трудили да направе лепа програмска решења, а нису разумели суштину пословног система и пословања.

2.3.2.2 Побољшања успешности информационих система

Да би се побољшала успешност информационог система потребно је:

- пројектовање информационог система;
- планирање, управљање извођењем, праћење напретка;
- укључивање корисника.

Корисник познаје пословни процес и одређује потребе, а информатичар упознаје пословање и израђује информациони систем. Важно је да у процесу израђивања учествује и руководство, да би се упознало са стварним могућностима и бенефиције увођења информационог система, нарочито јер доноси коначне (за пословни систем судбоносне) одлуке.

3 Модели развоја информационих система

Активност развоја софтвера пролази кроз неколико фаза. Различити модели развоја софтвера имају више или мање исте фазе, али је редослед ових фаза различит, што чини да се модели развоја софтвера разликује један од другог. Иако постоје неки модели, где је могућ повратак на претходну фазу, постоје неки модели где то није могуће. Слично у неким моделима, фазе тестирања почињу на крају фазе развоја, док је у неким моделима истовремени процес са развојем. У даљем тексту биће укратко описани неки од модела.

3.1 Водопад (секвенцијални) модел

Постоје многи приступи развоју софтвера који се користе током процеса развоја софтвера. Ови приступи се такође називају процес развоја модела софтвера. Сваки процес моделирања прати одређени животни циклус, како би се осигурао успех у процесу развоја софтвера.

Један такав приступ који се користи у развоју софтвера је „The Waterfall Model“ - Водопад модел. Водопад приступ је био први процес моделирања, који је уведен и широко праћен у софтверском инжењерству како би се осигурао успех пројекта. У водопад приступу, цео процес развоја софтвера је подељен у одвојене фазе. Фазе су [9]:

1. Захтев спецификација,
2. Дизајн софтвера,
3. Имплементација,
4. Тестирање и одржавање.

Фазе су усклађене на каскадни начин, тако да друга фаза почиње када се постигне дефинисани скуп циљева за прву фазу и по одобрењу, отуда и назив приступа „водопад“.



Слика 5. Фазе развоја водопад модела

3.1.1 Фазе водопад модела

Прикупљање захтева и анализа: сви могући захтеви који се развију наводе се у овој фази. Захтеви су скуп функција и ограничења које крајњи корисник очекује од система. Захтеви се прикупљају од корисника, и анализирају се за њихову ваљаност и могућност укључења. На крају, документ потребних спецификација је креиран који служи у сврху смерница за наредну фазу модела.

Дизајн система и софтвера: пре почетка фазе стварног кодирања, врло је важно да се разуме захтев крајњег корисника, а такође треба имати идеју о томе како крајњи производ треба да изгледа. Приликом дизајнирања проучене су неопходне спецификације из прве фазе, и дизајн система је спреман. Дизајн помаже у одређивању хардвера и системских захтева, а такође помаже у дефинисању укупне архитектуре система. Спецификације дизајна система служе као инпут за наредну фазу модела.

Имплементација и тестирање јединица: на пријему пројектне документације система, рад је подељен на модуле/јединице и кодирање почиње. Систем је први пут елаборирао у мале програме под називом јединице, које су интегрисане у следећој фази. Свака јединица се развија и тестира функционалност система, и означава се као јединица за тестирање. Јединица за тестирање углавном проверава да ли су модули/јединице задовољили спецификације система.

Интеграција и тестирање система: јединице су сада интегрисане тако да формирају комплетан систем у фази интеграције и тестирају се да провере да ли су сви модули/јединице координирани једни са другима, да ли се систем у

целини понаша како спецификације налажу. Након успешног тестирања, систем се испоручује купцу.

Инсталација и распоређивање: купац примењује софтвер у сопственом систему. Потребе о којима купац треба да брине је да његов систем испуњава минималне системске захтеве софтвера. Купац такође треба да води рачуна о свим конфигурацијама система и реконфигурацији на његовој страни посла. Када је софтвер исправно инсталиран, почеће комуникацију са дилерима на „need-to-know“⁸ основи, и пријавити сваки недостатак који се јавља.

Одржавање: ова фаза модела се практично никада не завршава. Генерално, проблеми са системом (који се нису нашли у току развојног циклуса) долазе након почетка практичне примене система, тако да су питања која се односе на систем решена након његовог распоређивања. Не уочавају се сви проблеми директно, али се појаве с времена на време и треба да буду решени, па се овај процес назива одржавање, иако је још увек прилично у фази тестирања.

3.1.1.1 Предности водопад модела

- Постоји јасна усаглашеност рада и контроле. То олакшава подешавање распореда задатака да буду завршени у одређеном року.
- Након завршетка процеса једне фазе, могућ је почетак друге фазе, због чега не постоји преклапање фаза, или производ не мора да иде кроз различите итеративне кораке.
- Много је лакше замислити овај модел пошто се процеси спроводе линеарно, трошкови ресурса се смањују у великој мери, што заузврат знатно помаже у смањењу трошкова пројекта.
- На крају, документација и тестирање се врши по завршетку сваке фазе, што помаже у одржавању квалитета пројекта.

3.1.1.2 Недостаци водопад модела

- Не примају се сви захтеви истовремено, добијање захтева од купца се може наставити и након прве фазе. То може негативно утицати на процес развоја система.
- Проблеми једне фазе никада нису у потпуности решени у истој фази. Заправо, многи проблеми настају након завршетка фазе. То доводи до лоше структурираног система, како нису сви проблеми решени непосредно.
- Пројекат није увек расподељен на флексибилан начин.
- Како су захтеви купца додати на листу, а нису сви услови испуњени, то доводи до развоја готово неупотребљивог система. Ови захтеви се затим испуњавају у новијој верзији система, што повећава трошкове развоја.

⁸ Описује играничења података која се сматрају врло осетљивим.

Иако постоји неколико недостатака модела, он наставља да буде најкоришћенији приступ у индустрији. Ово се може приписати чињеници да предности овог модела превазилазе своје недостатке у већини типова пројеката. И на крају, чак и ако тим нема много искуства, овај модел још увек савршено служи сврси.

Секвенцијални приступ, поред добрих особина, не даје увек праве ефекте када је у питању остварење претходно дефинисаних циљева [7]. Узроци су вишеструки, а неки од њих су следећи:

1. У секвенцијалном моделу примене методологије животног циклуса крајњи корисник није довољно укључен у процес развоја програмског производа;
2. Између почетка пројекта и првих оперативно применљивих резултата код корисника временски интервал је доста дуг;
3. Често се јавља потреба да се прецизни, методолошки комплексни пројектантски кораци изводе на основу недовољно прецизно идентификованих информационих захтева;
4. Уместо да то буде постепено, постоји потреба да се у развој информационог система одједном уложи значајна финансијска средства.

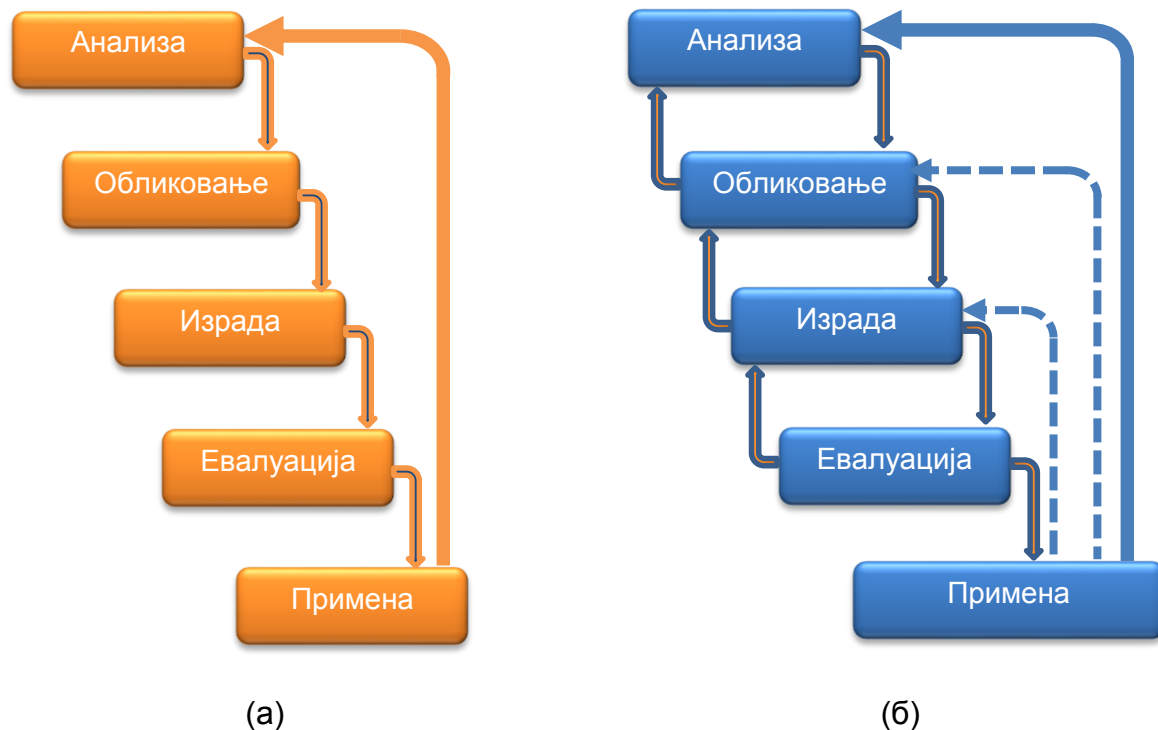
Узајамнио деловање прва три недостатка секвенцијалног приступа има за последицу да се скривене мане програмског производа и претходно неидентификовани кориснички захтеви често откривају тек у фази увођења апликације у употребу, што је јако касно јер се корекција грешака и одржавање компликују, а продужава се време потребно за добијање коначног решења апликације. У циљу избегавања негативних ефеката секвенцијалног приступа јавио се прототипски приступ, као и еволутивни, инкрементални, спирални, звездасти, "V" модел и други мање значајни модели.

Поред класичног модела, могу се издвојити следеће варијанте секвенцијалног (водопадног) модела: псеудоструктурни водопадни модел и радикални (структурни) развој.

Класични водопадни модел (слика 6 (а)) редоследно (секвенцијално) напредује из фазе у фазу. Нису дозвољене накнадне промене резултата претходних фаза. Прикладан је великим пројектима (инвестицијама), за добро дефинисано окружење, где постоје разрађене процедуре ручне обраде или рачунарски систем који треба унапредити. Недостаци овог модела су изражени у случају грешака или нових/промењених захтева, као и у потреби увођења према горе (bottom up⁹) модула, подсистема и система. Систем није употребљив док није у потпуности готов. Проблем представља и представа о производу на основу писане спецификације.

⁹ Наглашава кодирање и рано тестирање, које може да почне чим је први модул назначен.

Псеудоструктурни водопадни модел (слика 6 (б)) садржи повратну везу и могућност промене резултата претходних фаза. Овај модел развоја информационог система омогућава примену техника структурног програмирања.



Слика 6. Упоредни приказ класичног развоја (а), псеудоструктурираног и радикалног (б)

Радикални (структурни) развој (слика 6 (б)) омогућава да се активности различитих фаза могу обављати истовремено. Дозвољава коришћење речника података, програмских језика четврте генерације и генератора апликација. Прикладан је када се унапред не зна коначни изглед система.

3.2 Verification & Validation Model - „V“ модел

Савршен софтверски производ је направљен када се сваки корак прихвати са пуно обзирности, да се прави производ развије на прави начин [10]. Software verification & validation је један такав модел, који помаже дизајнерима система и тест инжењерима да потврде да је производ изграђен на прави начин током процеса развоја и побољшан квалитет софтверског производа.

Software verification & validation чини сигурним, да су одређена правила слеђена у току развоја софтвера и омогућава да софтвер који је направљен испуњава потребне спецификације. Ово смањује ризик повезан са било којим софтверским пројектом до одређеног нивоа помагајући у откривању и исправљању грешака, које се несвесно учине током процеса развоја.

3.2.1 Шта је верификација?

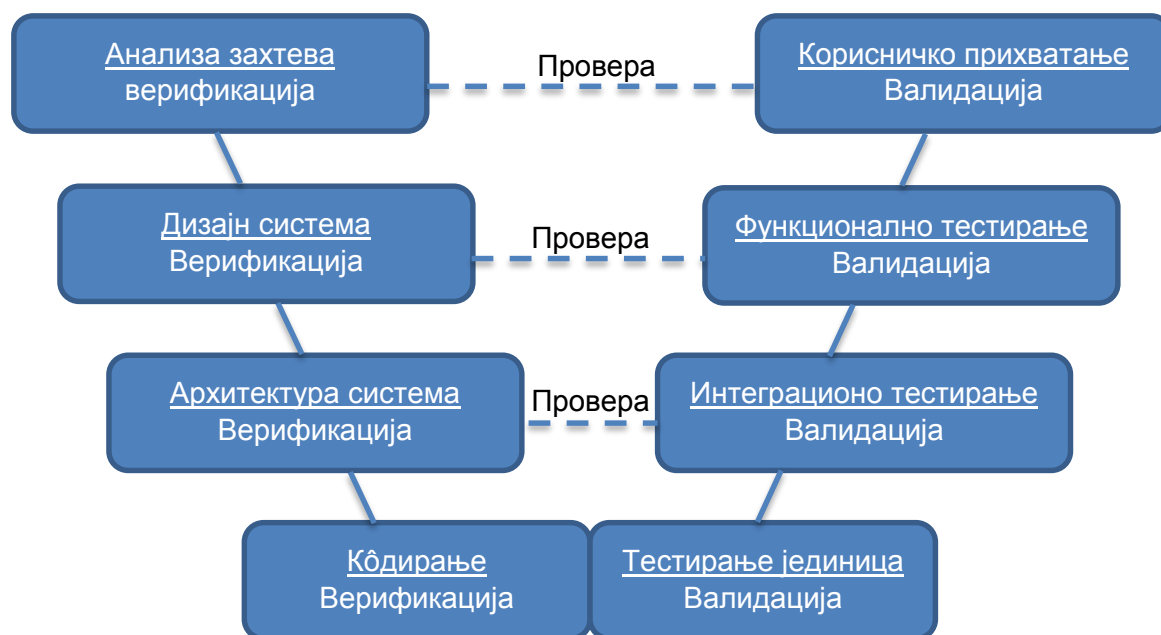
Стандардна верификациона дефиниција гласи: "Are we building the product RIGHT?" тј. верификација је процес који чини да се софтверски производ развија на прави начин. Софтвер би требало да потврди своје унапред дефинисане спецификације. Како развој производа пролази кроз различите фазе, анализа се врши како би се осигурало да су испуњене све потребне спецификације.

Методe и технике које се користе у верификацији и валидацији треба пажљиво пројектовати, од којих планирање почиње од самог почетка процеса развоја. Верификациони део целокупног модела долази пре валидације, који укључује анализу захтева, дизајн система, преглед система, архитектуру, кодирање.

3.2.2 Шта је валидација?

То је процес проналажења да ли је производ исправно развијен? То значи да, без обзира за шта је софтвер развијен, треба да извршава задатке које корисник очекује да уради. Софтверски производ треба функционално урадити задатке што треба, и треба да задовољи све функционалне захтеве постављене од стране корисника. То се врши у току или на крају процеса развоја, како би се утврдило да ли производ задовољава прописане услове.

Процеси валидације и верификације иду заједно, али видно процес провере ваљаности почиње након завршетка процеса верификације (када се заврши кодирање производа). Свака активност (као што су верификација тражених спецификација, верификација функционалности дизајна итд.) има своју одговарајућу валидациону активност (као што су функционална тестирања, код валидације/тестирање, проверу система/интеграције, итд.).



Слика 7. Фазе развоја 'V' модела

Често се каже да је 'V' модел проширење водопад модела. За разлику од водопад модела, активности у овом моделу су 'V' облика, са кôдирањем на доњем шпицу, тј на средишњим активностима. Свака фаза у циклусу развоја софтвера има одговарајућу фазу у циклусу тестирања, где је верификација једна страна развоја апликације, валидација, завршни процес, се састоји од активности тестирања [11]. Активности које спадају под верификационе су:

- **Анализа захтева:** Информације о предложеном систему се прикупљају од крајњег корисника, користећи припремљена документа потребних спецификација софтвера, како би се припремила основа софтвера;
- **Дизајн система:** Познат и као функционални дизајн, има за циљ да припреми функционални дизајн софтвера. У случају било које функционалности, није изводљиво исто као што је наговештено кориснику;
- **Архитектура система:** Када је дизајн система на месту, архитектура потребна за систем је направљена, која се често назива дизајн високог нивоа. Овде је то релациона мрежа база, табеле базе података итд. све на чему је рађено;
- **Кôдирање:** Након завршетка архитектуре систем почиње се са кôдирањем. За кôдирање, цео систем је подељен на мале модуле, који се касније интегришу и формира се цео систем.

Након завршетка верификационих активности, долазу друга страна 'V' модела, активности валидације. То су:

- **Тестирање јединица:** Ово је прва фаза валидације, где се мали модули који су развијени тестирају да ли функционални;
- **Интеграциона тестирања:** Када су модули спремни, интегришу се и затим тестирају. Тестирање помаже у одређивању грешака/недостатака у интерфејсу и интеракцији између два различита модула;
- **Функционално тестирање:** У овој фази се тестирају стварне спецификације против тражених спецификација система. Врши се испитивање из перспективе крајњег корисника. Такође, постоји шанса да су неке од функционалности уочљиве након потпуне интеграције система;
- **Корисничко прихватање:** Циљ овог испитивања је да се провери, да ли је систем у складу са траженом спецификацијом софтвера. То помаже у утврђивању да ли је софтвер развијен у складу са критеријумима прихватљивости и да ли систем треба да буде прихваћен или не. Такође, ова фаза тестирања се врши у симулираном реалном окружењу.

3.3 Прототипски модел

Уз структурни приступ, прототипски приступ развоју програмског производа представља концепт који се може применити у оквиру методологије животног циклуса. Прототипски приступ постаје у пуној мери практично применљив (иако је идеја о прототипском приступу у софтверском инжењерству била присутна доста раније) тек појавом свеобухватних и квалитетних пројектантских и програмерских CASE (Computer Aided Software Engineering) производа, који су интегрисани са окружењем четврте генерације [7]. У зависности од намене, могу се уочити следеће три врсте прототипског модела:

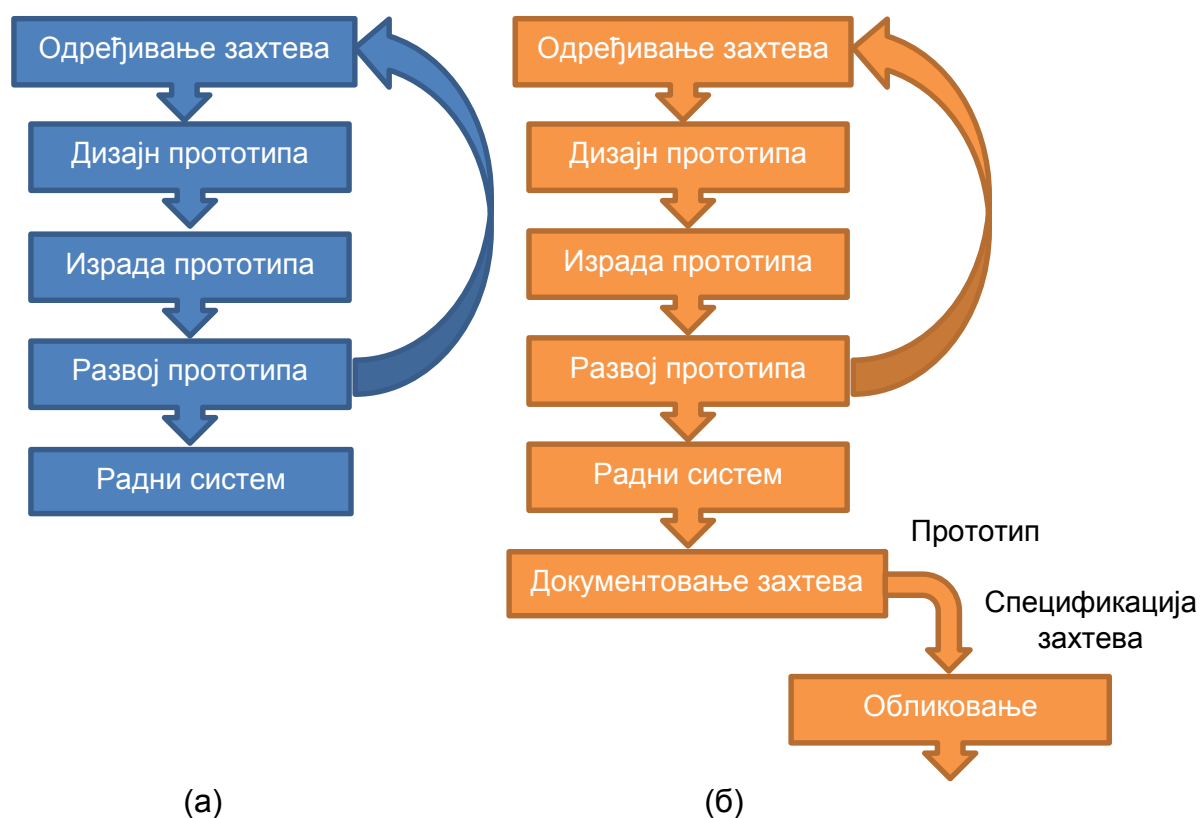
- *модел опонашања* (модел у природној величини), односно једноекрански или вишеекрански модел којим се приказује како ће изгледати део система (нпр. интерфејс);
- *истраживачки модел*, за истраживање делова система како би се проверила нека кључна подешавања (нпр. провера перформанси одређеног модула);
- *уградни модел*, тј. тражење различитих начина на којима се систем може изградити (нпр. који систем за управљање базама података, програмски језик).

Прототип може постепено, инкременталном дорадом, да постане део завршног информационог система. Прототипски развој подразумева итеративни приступ, обично коришћењем програмског језика четврте генерације (4GL). Радни модел даје се на увид кориснику и омогућава кориснику стварање слике о изгледу система. Корисник даје примедбе за поправку и побољшања, чиме се

стиче боља слика о захтевима корисника. Уједно се уклањају могућа изненађења на крају развоја.

Савремени софтверски алати омогућавају брзу израду прототипа. Функционални прототип доградњом може да постане радни систем (слика 8.(а)). Овакав приступ развоју информациоих система је прикладан за мале пројекте. Предности су у итерацијама промена (корисници се смеју предомислити), повећању креативности и брзини развоја. Недостаци су у томе што се "заборавља" да прототип није прави систем, могући неуспех замене прототипа радним системом, документација произлази из израде па постоји опасност да писане спецификације никад неће бити направљене, као и немогућност исправне процене и планирања ресурса.

Ограничени/структурни развој прототипа служи као средство за утврђивање захтева. Добија се нефункционални прототип (који се не може користити у обављању радних задатака корисника, а садржи изглед екранских форми, мени и извештај), чији се развој у одређеном тренутку прекида и следи фаза обликовања система (слика 8 (б)).



(а) (б)
Слика 8. Фазе брзог (а) и структурног (б) развоја прототипа

3.4 Еволутивни модел

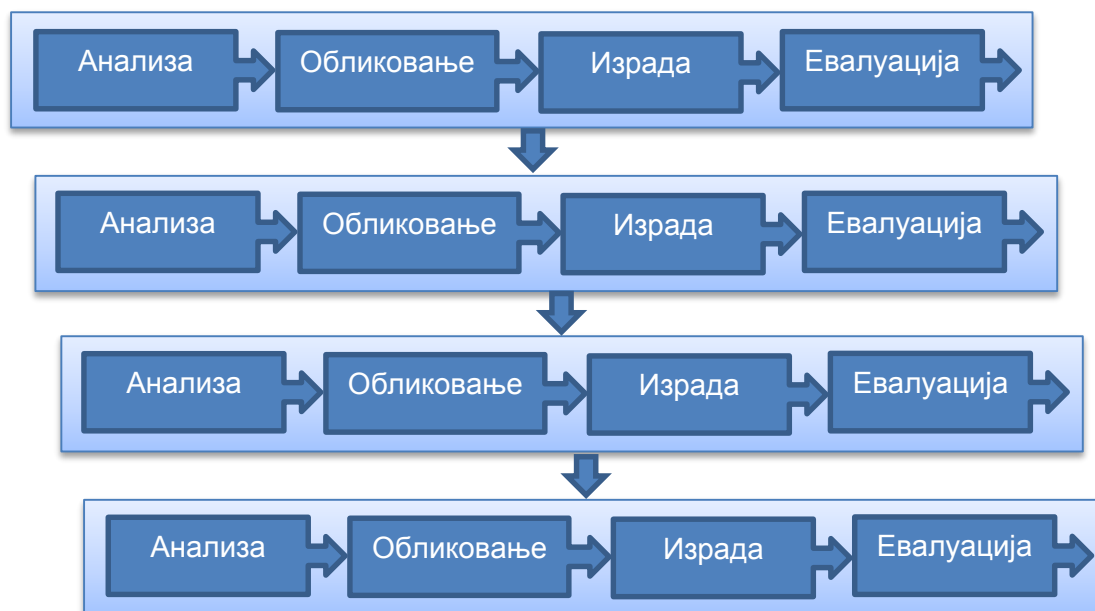
Примена информационих система мења поглед корисника, а његове потребе се мењају (расту) током примене [7]. Може се закључити да информациони систем расте са пословним системом кога подржава. Тежња за прилагођавањем система је присутна и у развоју информационог система. Један од основних принципа, на коме се заснива примена секвенцијалног модела методологије животног циклуса, је да реализација наредне фазе не започиње док се текућа фаза не заврши. Уочава се да управо примена овог принципа, код неких пројеката, може проузроковати негативне ефекте примене методологије животног циклуса.

Еволутивни модел примене методологије животног циклуса, насупрот секвенцијалном, предвиђа да је за одређене фазе животног циклуса програмског производа могуће да наредна фаза започне пре него што се претходна заврши, што доводи до одређеног степена паралелизма у реализацији тих фаза. Према томе, фазе животног циклуса почињу да се спроводе итеративно. До ове идеје се дошло на основу претпоставке да не треба одједном реализовати комплетну фазу и утрошити за то велику количину времена и новца, у ситуацији када се пројектантске активности изводе на основу недовољно прецизно идентификованих информационих захтева. Брз прелазак у наредну фазу треба да обезбеди бољу основу за каснији успешни завршетак претходне фазе.

Како је један од битних мотива за настанак еволутивног приступа проблем недовољно прецизно идентификованих информационих захтева, сматра се да његова практична примена има смисла уколико се он комбинује са прототипски приступом, као методом за тачно утврђивање информационих захтева. За утврђивање информационих захтева се претежно примењују одбачени прототипови.

Примењују се две варијанте еволутивног приступа. Према првој, након утврђивања прецизних информационих захтева, резултати концептуалног и имплементационог пројектовања база података се интегришу, а пројекти подсистема се усаглашавају са шемом интегрисане базе података. Другим речима, потпројекти се поново посматрају као целина и на њих се примењују, нешто измењени, кораци концептуалног и имплементационог пројектовања, као при примени секвенцијалног модела методологије животног циклуса. Ова варијанта еволутивног приступа комбинује многе добре стране секвенцијалног модела методологије животног циклуса и прототипског приступа, али не решава проблеме дугог временског интервала од почетка пројекта до појаве првих, оперативно применљивих резултата код корисника, и потребе улагања финансијских средстава одједном, а не постепено. Према другој варијанти, потпројекти се реализују међусобно независно и могу бити фазно померени у

времену. На тај начин се решавају проблеми дугог временског интервала од почетка пројекта до појаве првих резултата и потребе улагања одједном финансијских средстава. Овакав независан рад, међутим, може довести до нижег степена интегрисаности информационог система.



Слика 9. Могући еволутивни модел примене методологије животног циклуса

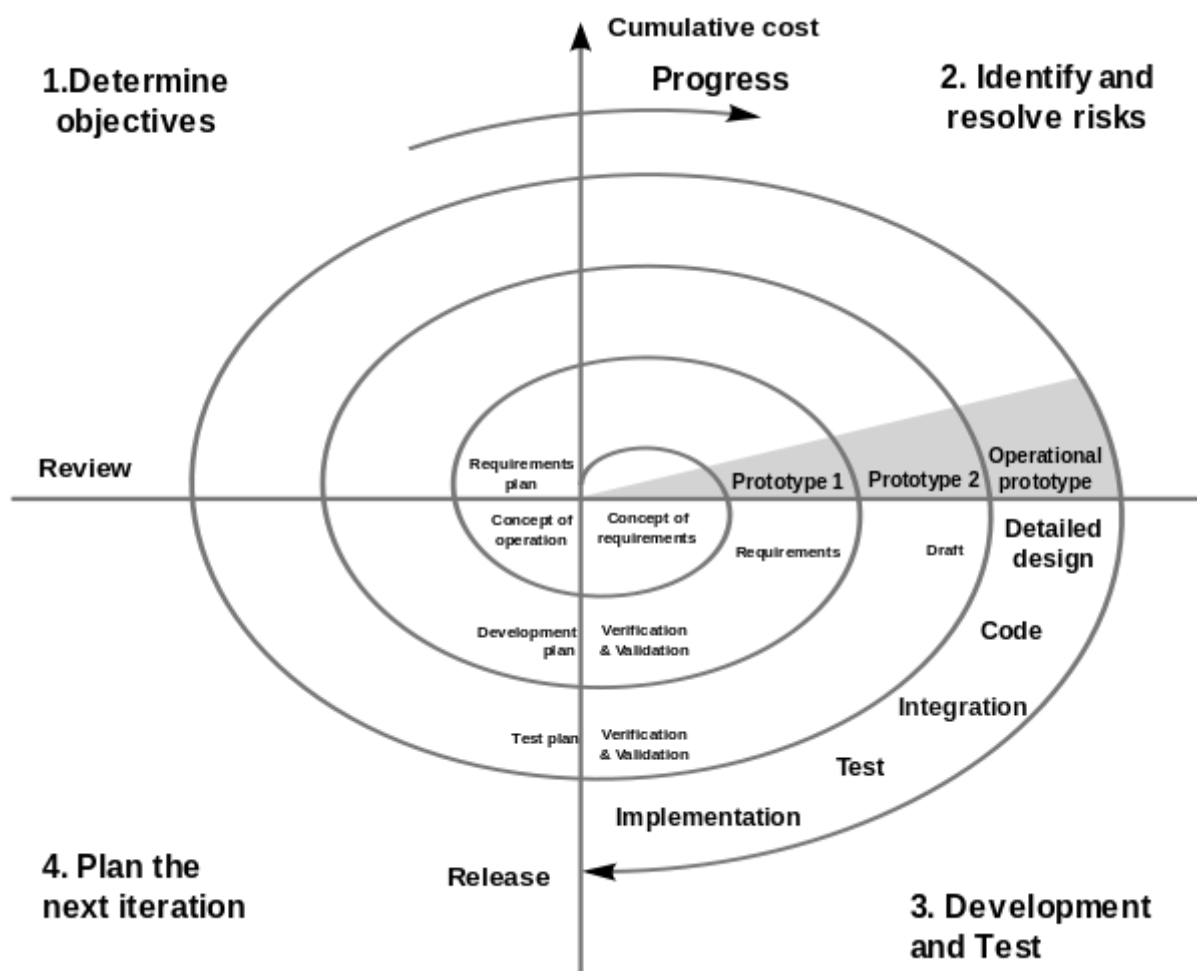
На слици 9. је приказан могући еволутивни модел примене методологије животног циклуса. Развој се врши у инкременталним корацима, довољно малим да се могу обављати прототипски. Сваки од низова развојних активности (анализа, обликовање, израда, евалуација) води оперативном производу који се испоручује и који генерише наредне захтеве.

3.5 Инкрементални модел

Као и у случају еволутивног модела, на почетку примене инкременталног модела, комплетно се спроводи фаза стратегије методологије животног циклуса [7]. Након тога, формирају се релативно мањи потпројекти са нижим степеном међусобне интеграције и утврде се следећи параметри потпројекта: циљеви, ресурси и рок предаје у употребу. Циљеви и ресурси су променљиви параметри, који се по потреби могу мењати у току самог пројекта, док је рок предаје у употребу програмског производа обавезно фиксни параметар. Потпројекти се реализују међусобно независно и могу бити фазно померени у времену. Инкрементални модел се може посматрати као посебна варијанта еволутивног модела животног циклуса.

3.6 Спирални модел - Нови приступ ка развоју софтвера

Водопад модел је један од најстаријих и најједноставнијих модела дизајнираних и праћени током процеса развоја софтвера [12]. Али модел има своје недостатке, као што је не постојећа правилна подела фаза у животном циклусу, грешке/проблеми везани за фазу нису решени током текуће фазе. Уместо тога сви ти проблеми везани за једну фазу спроводе се у наредној фази и потребни су да се реше у следећој фази, а за то је потребно више времена за наредну фазу како би се решили проблеми из претходне фазе. Фактор ризика је најважнији део, који утиче на стопу успешности развијеног софтвера и прати овај модел.



Слика 10. Фазе развоја спиралног модела

У циљу превазилажења мане водопад модела, било је неопходно да се развије нови модел развоја софтвера, који би могао да помогне у обезбеђивању успеха софтверског пројекта. Један такав модел је развијен који садржи уобичајене

методологије праћене на водопад моделу, али су уклоњени скоро сви могући фактори ризика познати на том моделу. Модел се назива "Спирални модел".

Постоје четири фазе у овом моделу које су: планирање, анализа ризика, инжењерство и евалуација. Ове четири фазе итеративно следе једна за другом у циљу отклањања свих проблема, са којима се суочава у водопад моделу. Понављање фаза помаже у разумевању проблема повезаних са фазом и решавању тих проблема када се иста фаза понови следећи пут, планирање и развој стратегије треба следити приликом пролаза кроз следеће фазе. Фазе су:

Determine objectives: У овој фази, циљеви, алтернативе и ограничења пројекта се одређују и документују.

1. Системски захтеви су дефинисани што је детаљније могуће укључујући артефакте за функционалност, перформансе, хардверски/софтверски интерфејс итд.;
2. Алтернативе као што су build vs. buy или могу ли се користити постојеће компоненте;
3. Ограничења као што су цене, распоред и намена интерфејса.

Циљеви и остале спецификације су фиксне како би се одлучило које стратегије/приступе пратити током животног циклуса пројекта.

Identify and resolve risks: Ова фаза је најважнији део спиралног модела. У овој фази, све могуће (и расположиве) алтернативе, које могу да помогну у развоју исплативости пројекта су анализирани и стратегијски се одлучује како да се користе. Ова фаза је посебно додата у циљу идентификовања и решавања свих могућих ризика у развоју пројекта. Ако ризици указују било какве неизвесности у захтевима, прототип може да се користи да настави са расположивим подацима и сазна могуће решење у циљу да се размотри о потенцијалним променама у захтевима.

Development and test: У овој фази, стварни развој пројекта се спроводи. Креира се прототип система из идејног пројекта праћењем уобичајеног образаца креирања и врши се преглед дизајна, кода и тестирање. Излаз ове фазе је прошао кроз све фазе итеративно у циљу добијања побољшања у исти пројекат.

Plan the next iteration:

1. Прегледа се први прототип за предности, слабости и ризике;
2. Стварају се услове за други прототип;
3. Дизајнира се други прототип;
 - i. Креира се пројектни план;
 - ii. Документује се конфигурација плана за управљање прототипом;
 - iii. Развија се тест план;
 - iv. Осмишљава се инсталациони план.

Процес напредује у спиралном погледу указујући да је праћењем итеративног поступка, постепено изграђен сложенији софтвер како се иде путем понављања кроз све четири фазе. Прва итерација у овом моделу се сматра најбитнијом, како у првој итерацији, скоро сви могући фактори ризика, ограничења, су идентификовани и у следећој итерацији, све познате стратегије се користе да би се комплетирао софтвер. Радикалне димензије показују еволуцију производа према комплетном систему.

Међутим, као и сваки систем, има своје предности и мане, спирални модел има превише својих предности и мана. Како је овај модел развијен да превазиђе недостатке водопад модела, праћења спиралног модела, високо квалификованих људи у области планирања, анализе ризика и ублажавање последица, развоја, односа са клијентима итд. су обавезна. Ово заједно са чињеницом да, процес треба да се понавља више пута, захтева више времена и некако је скуп задатак.

3.7 Поређење између водопад модела и спиралног модела

Развој софтвера на ширем нивоу обухвата све активности, између испољавања идеје за софтвер до одржавања поменутог софтвера [13]. Понекад, развој може бити упућен на актуелни задатак писања софтверског кода, и такође га одржавати током одређеног временског периода. Током времена, постоје различити модели развоја софтвера, који су коришћени за наведену сврху. Први од модела софтвера који се користи је водопад модел. Други модел, који се најчешће користи је спирални модел. Оба ова модела су међу најпопуларнијим моделима у области развоја софтвера. Старији од ова два модела је водопад модел. Често постоји водопад модел vs. спирални модел дебате, која се може чути у ходницима, када се предузима нови процес развоја софтвера. Разумевање разлике између ова два модела ће олакшати да се одлучи, који је прави модел који ће се користити за развој софтвера. Међутим, подједнако је важно да се разуме сваки од ових модела, што ће помоћи у бољем разумевању разлике између њих.

3.7.1 Водопад - waterfall модел

Водопад модел се често назива линеарни и секвенцијални модел, проток активности у овом моделу су прилично линеарна и секвенцијална као што име сугерише. У овом моделу, развој активности софтвера прелази на следећу фазу само ако су активности у текућој фази готове. Међутим, као што је случај са водопадом, не може се вратити на претходну фазу. Фазе овог модела су:

- Прикупљања и анализа захтева,

- Дизајнирање,
- Кодирање,
- Интеграција система,
- Тестирање и отклањање грешака,
- Достављање,
- Одржавање.

3.7.1.1 Предности

Најважнија предност оваког модела се заснива на томе да је неопходно минимално планирање за кораке који следе, како активности у свакој фази иду навише, изводљиво је да једна не садржи план за целу фазу. Постоји одређена количина дисциплине која се спроводи као једна, треба само гледати у једну фазу процеса у било ком тренутку. У другим моделима, често је тешко ухватити почетак и крај наведене фазе, што није случај са моделом водопада. Најважнија предности је да пројекат не пробија рок рапоредa. Број ресурса који раде на пројекту се не повећава сваког дана, као планирање истог на почетку саме фазе.

3.7.1.2 Мане

Као што постоје предности водопад модела, постоје и неки недостаци поменутог модела. Први и најважнији недостатак овог модела је немогућност прављења промене у систему, када су замрзнути системски захтеви. То није необично видети да се захтеви мењају у почетним фазама. Другим речима, не може се вратити на претходну фазу. То је мана, због чега је овај модел потиснут у полеђини. Исто се дешава да буде случај са хардвером и захтевом софтвера за софтвер. Ако се процес развоја софтвера продужава за дужи временски период, постоје шансе да хардвер и софтвер могућности изабрани за софтвер могу постати застарели на крају процеса развоја.

3.7.2 Спирални модел

Спирални модела уведен је, због недостатака код водопад и прототип модела софтверског инжењерства. То је комбинација поменута два модела развоја софтвера. Од имена модела, може се закључити да се активности развоја софтвера спровode попут спирале. Да би објаснили модел додатно, цео процес развоја софтвера је подељен на мале пројекте. Фазе спиралног модела су следеће:

- Фаза планиања;
- Анализирање ризика;
- Инжењеринг фаза;
- Кодирање и имплементација;

- Процена опстајања.

3.7.2.1 Предности

Недостатак водопад модела је предност спиралног модела. То је реалан модел, који се често користи у развоју великог софтвера. Постоји систематски приступ који се користи у спиралном моделу, који је интегрисан у оквиру итеративног. Ово помаже у обезбеђивању да нема проблема у софтверу. Промене у софтверу се могу обавити у било ком тренутку процеса развоја софтвера.

3.7.2.2 Мане

У спиралном моделу, важно је имати члана у тиму, који је стручњак у процени ризика. Без исправне процене ризика, постоје шансе да софтвер може бити потпуни неуспех. Клијент може да потроши много времена са развојним тимом да реше проблеме који су настали у софтверу. Ово може понекад бити као оптерећење за оба клијента, као и за развој софтвера. Ово такође доводи до учешћа купца током процеса развоја софтвера, што може да изазове да купац игнорише одређене ризике и изазове штету на пројекту.

3.7.3 Разлика између водопад модела и спиралног модела

Док у спиралном моделу, купац је свесан свих дешавања у развоју софтвера, у водопад моделу купац није умешан. То често доводи до ситуације, где софтвер није развијен у складу са потребама купца. У спиралном моделу, купац је укључен у процес развоја софтвера од речи „крени“. Ово помаже у стварању услова да софтвер задовољава потребе купца.

У водопад моделу, када се процес развоја помера на следећу фазу, не постоји корак назад. То често доводи до застоја, посебно у фази кодирања. Много пута се види да дизајн софтвера изгледа изводљиво на папиру, међутим, у фази имплементације може бити тешко да се кодира исти. Међутим, у спиралном моделу, пошто постоје различите итерација, прилично је лакше да се промени дизајн и учинити софтвер изводљивим.

У спиралном моделу могу се поновити различите фазе развоја софтвера, онолико пута колико је потребно, током целог процеса развоја. Ово такође помаже у претраживању уназад или ревизији процеса. Ипак, није могуће урадити исто у водопад моделу, који нема такве могућности.

Често људи имају забуну око водопад модела или спиралног модел због чињенице, да спирални модел изгледа сложеније. Може се приписати чињеници да постоји много итерација, које су једна од предности модела. Истовремено, често не постоји документација укључена у спиралном моделу,

што отежава праћење целокупног процеса. С друге стране, водопад модел има секвенцијално напредовање, заједно са јасном документацијом целокупног процеса. Тиме се обезбеђује да један од модела има бољи утицај над читавим процесом.

Из горње дискусије о спиралном моделу против водопад модела, јасно је да оба модела имају своје предности и недостатке. Док је неодлучност присутна између водопад и спиралног модела дебате, најбоље је да се процени софтвер који се развија и онда направи одговарајући избор. Величина пројекта и ургентност софтвера мора бити узета у обзир за исти. Истовремено, средства на располагању ће имати важну улогу у процесу развоја софтвера.

4 Алати за пројектовање информационих система (CASE алати)

Развој информационих технологија карактерише заостајање софтвера у односу на хардвер [14]. Поменути недостатак софтвера, који се често назива софтверска криза, настаје због ниске продуктивности и високих производних трошкова.

Решење софтверске кризе је у искоришћењу особина инжењера проверених у пракси, и то, пре свега, методичности и оперативне дисциплине. Као резултат настаје софтверски инжењеринг који у себи садржи систематизоване и координиране активности потребне при пројектовању, имплементацији, експлоатацији и одржавању софтверских производа.

Даљи развој софтверских система на данашњем нивоу могућности рачунара и очекивања корисника, захтева високостручан рад и програмирање за своју реализацију. Пошто је ручно развијање софтвера од најнижег нивоа скупо и дуготрајно и са не увек предвидивим резултатима, постоји потреба да се развој софтвера олакша, због чега је, пре више од двадесет година, настало софтверско инжењерство као дисциплина.

Непостојање одговарајућих софтверских алата за подршку развоја софтверских производа, као и комплексност задатака и техника које се у методологији животног циклуса и у структурном приступу примењују, представља мотив за појаву CASE алата [7]. Скраћеница CASE је акроним назива на енглеском језику Computer Aided Software Engineering или Computer Aided System Engineering (програмско/системско инжењерство подржано рачунаром), док је скраћеница CAISE акроним назива Computer Aided Information System Engineering (информационо/системско инжењерство подржано рачунаром).

Под CASE алатом се подразумева било који програмски производ намењен за подршку, или аутоматизацију, барем једног задатка у оквиру животног циклуса другог програмског производа, или је намењен за комплетну подршку пројектовања и реализације другог програмског производа. Основни циљеви примене CASE алата су: обезбеђење задовољавајућег квалитета пројекта и пројектне документације, обезбеђење задовољавајућег квалитета самог програмског производа, повећање продуктивности пројектаната и програмера, скраћење времена пројектовања и реализације програмског производа, обезбеђење једноставног и јефтиног одржавања програмског производа.

Софтверски инжењеринг је строга примена инжењеринга, научних и математичких принципа и метода у економичној производњи квалитетног софтвера [15]. Као посебна дисциплина, која се бави економско-технолошким аспектима израде софтвера, као и свака друга технологија, може се посматрати са два аспекта:

- Конвенционална (ручно програмирање, дефинисање и документовање проблема и одржавање система) и
- Неконвенционална (CASE).

Технике и кораци су слични а суштинске разлике огледају се у начину контроле, израде и одржавања производа (софтвера). Када говоримо о CASE технологијама (или програмским алатима за развој апликација) онда морамо нагласити да према захтевима и потребама, само програмирање се врши софтверски.

4.1 Дефиниције CASE алата

CASE алати служе за аутоматизацију софтверског инжењерства и самим тим представљају основни алат којим се служи пројектант информационог система [14].

Прве дефиниције CASE алата су подразумевале да представљају системе чији су циљеви да дефинишу, интегришу и аутоматизују што је могуће више фаза у развоју софтвера. CASE алати омогућавају да развијање софтвера постане више инжењерска делатност, а мање индивидуална уметност и умеће.

Зависно од тога које фазе пројектовања и имплементације CASE алати покривају, они се деле на CASE алате на вишем и CASE алате на нижем нивоу. CASE алати на вишем нивоу покривају прве фазе у производњи софтвера (анализу система и пројектовање), а CASE алати на нижем нивоу пружају помоћ у фазама програмирања.

CASE систем представља алат који служи као помоћ пројектанту информационог система. Од ефикасности овог алата може да зависи квалитет готовог производа (информационог система), тако да је пројектанту веома важно да одабере прави алат који ће га замењивати у већини мануелних послова везаних за пројектовање.

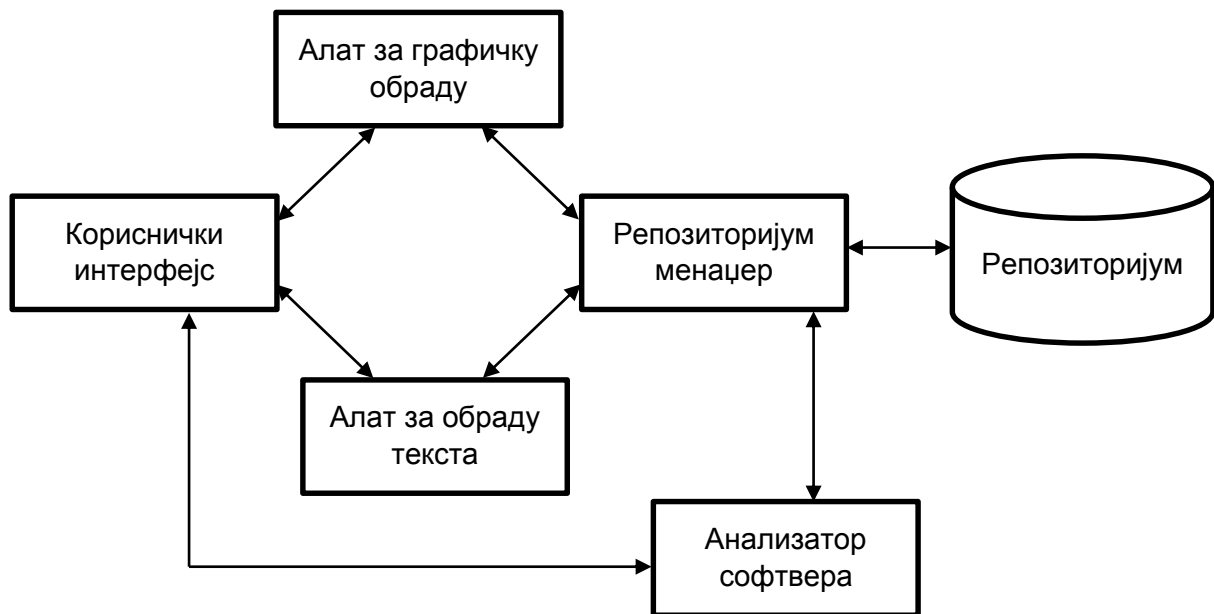
Од ефикасности CASE алата може зависити квалитет готовог софтверског производа. Успешним коришћењем правилно одабраног CASE алата може се:

- минимизирати време и труд (коштање) развоја софтвера;
- вишеструко повећати продуктивност у писању софтвера;
- подићи ниво квалитета;

- повећати поузданост;
- стандардизовати произведени софтвер.

4.2 Елементи CASE алата

Типично CASE окружење укључује: репозиторијум, алате за графичко представљање, софтвер за дефинисање текста, софтверски интерфејс према репозиторијуму, софтвер за процену и интерфејс према кориснику као што је приказано на слици 11. [14].



Слика 11. CASE архитектура

Репозиторијум је активан речник података који подржава дефинисање различитих типова објеката и њихових веза. Алати за графичку обраду омогућавају развијање разних типова дијаграма, као и оцену комплетности дијаграма на основу унапред дефинисаних правила.

Алат за обраду текста омогућава дефинисање назива, садржаја и детаља појединих ставки у репозиторијуму. Кориснички интерфејс је интерпретер који одређује форму података коју треба узети (графика или текст). Анализатор софтвера је експертни део CASE алата и он анализира улазе како у дијаграм тако и у репозиторијум, утврђујући њихову лексичку комплетност (исправност назива варијабле), а и проверавајући компатибилност са осталим објектима у апликацији. Кориснички интерфејс обезбеђује интерактивну и off-line обраду преко екрана и извештаја.

Идеални CASE треба да омогући потпуну аутоматизацију комплетног животног циклуса пројекта информационог система, обухватајући почетну иницијативу,

ниво анализе, рад на одржавању софтверског производа све до његовог повлачења. Такав CASE постаје жижа за све послове софтверског инжењерства, па се рад на развоју софтвера концентрише на логички аспект пројектовања. Управо због тога идеални CASE треба да омогући: израду архитектуре процеса организације, планирање и праћење пројекта, групни рад на развоју софтвера, апликативно и ручно дефинисање процедура, нормализацију података, генерисање шеме базе података, писање кода у кориснички одабраном програмском језику, аутоматско тестирање генерисаног кода према спецификацији апликације и експертну процену савршености производа, као и начин корекције. Идеални CASE треба већ у репозиторијуму да препозна компоненте које су за поновну анализу, дизајн и кодирање.

По правилу 40-20-40, које важи за развој програмског система, 40% пројектног времена отпада на анализу и моделирање, 20% је одвојено за програмирање, а преосталих 40% за тестирање [14]. Тренд добављача је да се елиминише кодирање, што чини само једну петину времена потребну за развој софтверског производа. Садашњи тренутак захтева постојање CASE алата који ће покрити процес тестирања, а самим тим и скратити време израде софтверског производа.

Будући CASE биће у могућности да у остатку 40% пројектног времена идентификује делове апликације који могу бити поново искоришћени [14]. Из наведеног се може претпоставити и пут развоја CASE алата, а то је ефикасније покривање времена за анализу и дизајн, као и времена потребног за тестирање.

4.2.1 Врсте CASE алата

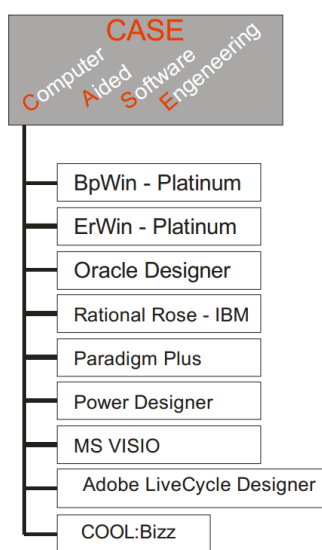
- Интегрисани алати за подршку свих фаза развоја [16],
- Алати за анализу и пројектовање (Upper CASE),
 - Алати за израду дијаграма,
 - Генератори форми и извештаја,
 - Алати за анализу,
- Алати за имплементацију и одржавање (Lower CASE),
 - Генератори кода,
- Алати за подршку (Cross Life-cycle CASE).

4.3 Могућност CASE алата за моделирање пословних података

Моделирање пословних података и генерисање базе података неког пословног система, која ће садржати све пословне податке важне за функционисање

пословног система мора бити интегралан и свеобухватан поступак [15]. На основу података доносе се одлуке у пословању и због тога је важно да сви потребни подаци буду обухваћени у бази података, а односи који постоје у реалном свету морају на исти начин бити забележени и у логици повезивања њених објеката. Израда логичког релацијског модела података и аутоматско генерисање релационе базе података спроводе се помоћу савремених CASE алата. Ти алати омогућавају издвајање детаљног логичког релацијског модела података као независног «производа», што пословном систему отвара могућност контроле развоја и примене пословних апликација.

У раду ће бити дан кратак преглед неколико CASE алата са аспекта могућности израде логичког и физичког модела података, и аутоматског генерисања физичке базе података из развијеног модела.

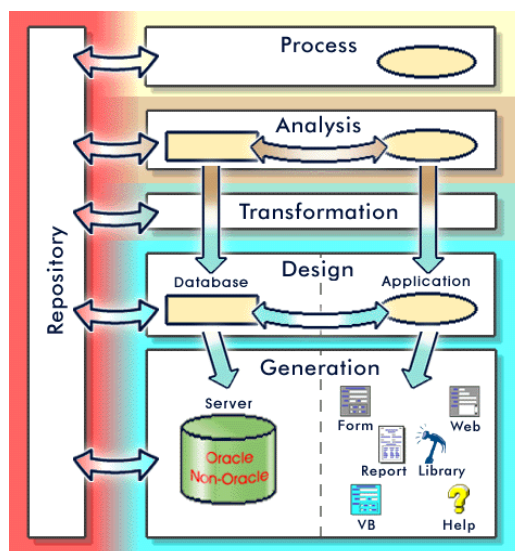


Слика 12. CASE алати

4.3.1 ORACLE Designer

Одлика ORACLE Designer алата је да су интегрисани и да подржавају све фазе у развоју система [14]. Подржавају концепт водопад модела, као и спиралног модела, развоја информационог система. Сви алати за развијање дијаграма развијени су у C++ тако да је омогућено коришћење MS-Windows библиотеке. Концепција ORACLE CASE окружење се базира на јаким радним станицама у Windows амбијенту и скупу алата (моделера, дијаграмера итд.) преко којих се приступа зајдничком речнику. Сваки податак се у ову базу уноси само једанпут, а онда се, по потреби, референцира у свим случајевима када је то потребно.

Карактеристика речника је да је вишекориснички и да се његова конзистентност стално прати процедурама за одржавање референцијалног интегритета и алатима за консолидацију.



Слика 13. Карактеристике ORACLE Designer-a

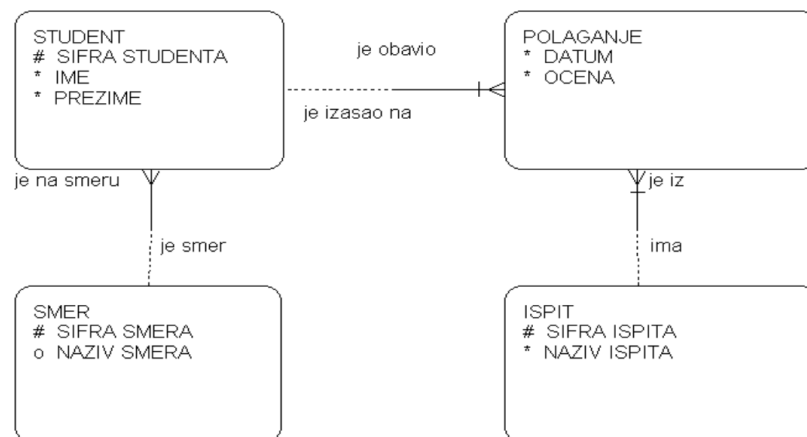
4.3.2 Entity Relationship Diagrammer (ERD)

Део Oracle пакета Designer/2000, који служи за моделирање података, подржава следеће концепте: атрибут, ентитет, специјализацију и генерализацију (подтипови и надтипови), везе [14].

При креирању ентитета ERD захтева само његово име, а сви остали детаљи везани за ентитет, као и дефинисање његових атрибута, могу се касније обавити кроз опцију едитовања ентитета. По жељи се атрибути могу приказивати у оквиру ентитета на дијаграму и тада постоји конвенција да се: испред атрибута који улазе у састав примарног кључа ставља знак „#“, испред атрибута који су обавезни „*“ и испред опционалних атрибута „о“, што доприноси прегледности дијаграма.

Слика 14. ERD - Екранска форма дефинисања ентитета

При моделирању веза међу ентитетима задају се њихова опционалност (обавезност везе се представља пуном линијом, а необавезност везе испрекиданом линијом са сваке стране везе понаособ) и њихова кардиналност (веза са горњом границом кардиналност 1 се представља једном тачком додира линије, која представља везу одговарајућег ентитета, док се веза са горњом границом кардиналност М представља разгранатом линијом са три додирне тачке). Посебне врсте веза које су подржане овим CASE алатом су: преносиве и непреносиве везе (transferable и non-transferable), а односе се на могућност превезивања 01:00 појаве ентитета са другом појавом референтног ентитета; спуштени кључеви, када се кроз овакву везу у кључ зависног ентитета укључује комплетан кључ ентитета од кога егзистенцијално зависи; рекурзивне везе (позива саму себе опет и опет да би се завршио неки задатак), које моделирају везу 1 појаве ентитета са другом појавом истог ентитета; ексклузивне везе, које моделирају ситуацију у којој може у једном тренутку постојати само једна од две или више веза које полазе од једног ентитета.



Слика 15. ERD пример модела „објекти и везе“

Дефинисање домена за ERD се врши у Repository Object Navigator-у, алату за директан приступ речнику података. Домени дефинисани на тај начин су доступни и осталим модулима ORACLE-овог интегрисаног CASE-а, а у ERD-у се користе за дефинисање атрибута.

4.3.3 BPwin (Business Process Windows)

BPwin алат за моделирање и анализу сложених пословних процеса је заснован на концептима стандарда IDEF0 (Icam DEFinition for Function Modeling)¹⁰ [14]. Због лакше идентификације процеса и њихове оптимизације, као и одбацивања сувишних процеса, настала је методологија моделирања пословних процеса.

¹⁰ Је функција моделирања за описивање производне функције, која нуди функционални програмски језик за анализу, развој, реинжењеринг и интеграцију информационих система, пословних процеса или софтверских инжењеринг анализа.

Моделирањем процеса се добија уређена структура са јасно дефинисаним правилима који се процеси одвијају и на који начин.

BPwin омогућава ригорозно тестирање конзистентности дијаграма. Динамички објекти који воде кроз модел спречавају грешке које најчешће настају при креирању модела.

Креирање документације из BPwin-ових модела процеса је изведено на исти начин као и у свим алатима фамилије *LogicWorks*-а. Постоји потпуна слобода у дефинисању сета података из модела активности, као и у дефинисању изгледа документације.

BPwin омогућава рад у графичком окружењу, уз пуну подршку рада мишем, тако да је едитовање модела података изузетно лако; дијаграми који се добијају дају потпуни приказ модела. Декомпозиција процеса се врши директно на дијаграму, а кретање по нивоима декомпозиције подразумева пребацивање са дијаграма на дијаграм. При раду на једном моделу отворени су сви дијаграми из тог модела.

4.3.3.1 IDEF0

IDEF0, акроним од (Icam DEFinition for Function Modeling, где је 'ICAM' акроним за Integrated Computer Aided Manufacturing) је методологија функције моделирања за описивање производне функције, коју нуди функционални језик моделирања за анализу, развој, реинжењеринг и интеграцију информационих система; пословних процеса [17].

IDEF0 је део IDEF¹¹ породице језика за моделирање у области софтверског инжењерства, и изграђен је на функционалном језику за моделирање Structured Analysis and Design Technique (SADT¹²).

4.3.3.1.1 IDEF0 приступ

IDEF0 може да се користи за моделирање разноврсних аутоматских и не-аутоматизованих система. За нове системе, може се прво користити за дефинисање захтева и задавање функција, а затим да се пројектује решење које ће задовољити захтеве и изводити функције. За постојеће системе, IDEF0 се може користити за анализу функција које систем обавља и за снимање механизма (средстава) по којима се они обављају. Резултат примене IDEF0 на систем је модел који се састоји од хијерархијског низа дијаграма, текстови који су азбучно поређани.

¹¹ Односи се на породице језика за моделирање у области система и софтверског инжењерства.

¹² Је методологија софтверског и системског инжењерства за описивање система као хијерархије функција.

4.3.4 ERwin

ERwin алат за моделирање података је израђен на концептима који су постављени IDEF1X (Integration DEFinition for Information Modeling)¹³ стандардом [14].

ERwin/ERX је CASE алат намењен моделирању података ER (Entity Relationship) методом, која је подржана IDEF1X (Integration DEFinition for information modeling) методом, а корисник може и опционо користити и IE (Information Engineering) методу. Моделирање података обухвата два аспекта: логичко дефинисање модела и физички дизајн базе. Коришћење овог алата подразумева истовремени развој логичког модела података и физичко дефинисање базе, што значи да није потребно никакво додатно пребацивање модела из логичког нивоа у физички и обрнуто.

У било ком тренутку дефинисања логичког нивоа базе, пре него што се почне са физичком дефиницијом базе података, потребно је одабрати циљни сервер. То је DBMS¹⁴ на коме ће модел бити реализован. Избор циљног сервера није дефинитивна ствар, то значи да се циљни сервер може мењати. Erwin инсистира на независности модела од DBMS платформе.

Сервер FRE (Forward and Reverse Engineering) омогућава живу везу ERwin-овог модела и базе података. Конекција се врши преко ODBC (Open Database Connectivity)¹⁵ driver-a¹⁶. На основу дефинитивног физичког модела података може се креирати база на одабраној DBMS платформи. Erwin поседује и модул за инверзни инжењеринг, који чита системске каталоге постојеће базе података, па на основу њих аутоматски креира ER дијаграм по методологији IDEF1X стандарда.

Оба смера се свде на синхронизацију модела података са базом података и поређење модела са стањем базе. Осим са базом података, Erwin може да пореди модел са SQL шемом или другим Erwin моделима. Модел може селективно да се синхронизује са променама у бази података у оба смера. Омогућена је постепена надоградња базе података из ERwin-овог модела података, уз контролу физичких параметара базе података. При свакој синхронизацији са базом, Erwin пружа детаљан извештај о свим променама насталим синхронизацијом. Исти модел података се може користити за

¹³ Је језик за моделирање података за развој семантичких модела података.

¹⁴ Database Management System или само DBMS је систем за управљање базом података. То је софтверско-хардверски пакет који омогућава да база података буде доступна свим корисницима.

¹⁵ Је стандардни програмски посреднички језик application programming interface (API) за приступ системима за управљање базама података (DBMS).

¹⁶ ODBC остварује DBMS независност помоћу ODBC driver-a као преносни слој између апликације и DBMS.

генерисање више база података са различитим DBMS платформама, или за конвертовање апликација са једне DBMS платформе на другу.

4.3.4.1 IDEF1X

Integration DEFinition for Information Modeling (IDEF1X) је програмски језик за развој семантичких модела података [18]. IDEF1X се користи за производњу графичких информационих модела који представљају структуру и семантику информација унутар окружења/система. IDEF1X дозвољава изградњу семантичких модела података који могу послужити као подршка управљању података као ресурс, интеграцији информационих система, као и изградњи рачунарских база података. Овај стандард је део групе IDEF језика за моделирање у области софтверског инжењерства.

4.3.4.1.1 Преглед

IDEF1X је техника моделирања података на стандардан, доследан и предвидив начин, како би се управљало подацима као ресурс. Може се користити у пројектима који захтевају стандардан начин дефинисања и анализирања ресурс података у оквиру организације. Такви пројекти обухватају уграђивање технике моделирања података у методологију управљање подацима као ресурс, интеграцију информационих система, или дизајнирање рачунарских база података. Примарни циљ IDEF1X стандарда је да обезбеди:

- Средства за потпуно разумевање и анализирање ресурс података организације;
- Уобичајени начин представљања и комуницирања комплексних података;
- Технику за представљање свеобухватних података потребних за покретање предузећа;
- Средства за дефинисање апликације-независан приказ података који може бити проверен од стране корисника и трансформисан у физички дизајн базе података;
- Техника за добијање интегрисане дефиниције података из постојећих извора података.

Главни циљ IDEF1X је да подржи интеграцију. Приступ интеграцији се фокусира на снимање, управљање, и коришћење једне семантичке дефиниције извора података која се назива концептуална шема - „Conceptual schema¹⁷“. Концептуална шема пружа јединствену интегрисану дефиницију података у оквиру предузећа која није базирана ни према било којој појединачној апликацији и независна је од података који се чувају или користе. Примарни циљ концептуалне шеме је да обезбеди конзистентну дефиницију значења међусобних односа између података који се могу користити за интеграцију,

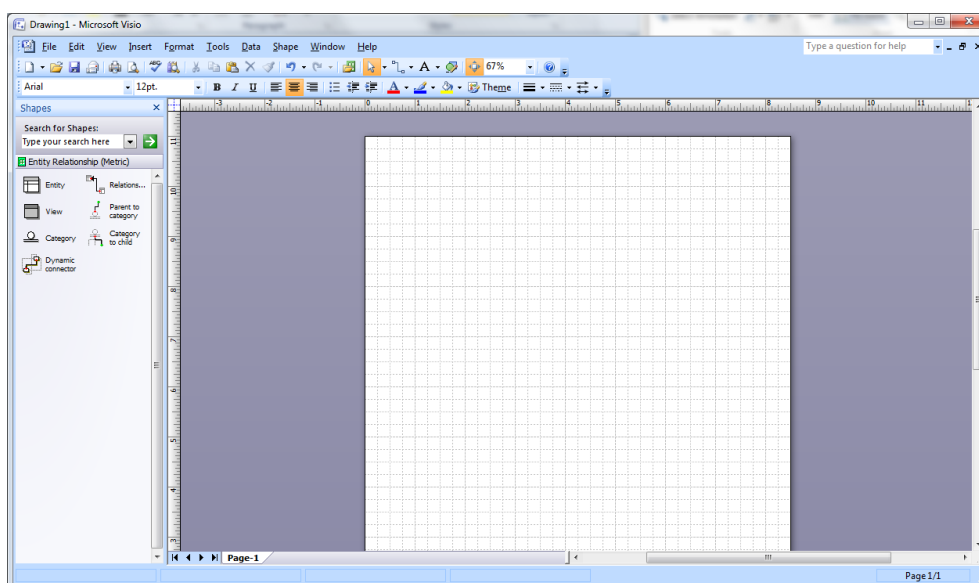
¹⁷ Висок ниво описа потреба пословних информација.

дељење, и управљање интегритетом података. Концептуална шема има три важне карактеристике:

- У складу са инфраструктуром пословања и односи се на све области примене;
- Проширива, тако да нове податке може дефинисати без мењања претходно дефинисаних података;
- Прилагодљива потребама корисника и различитом складиштењу података.

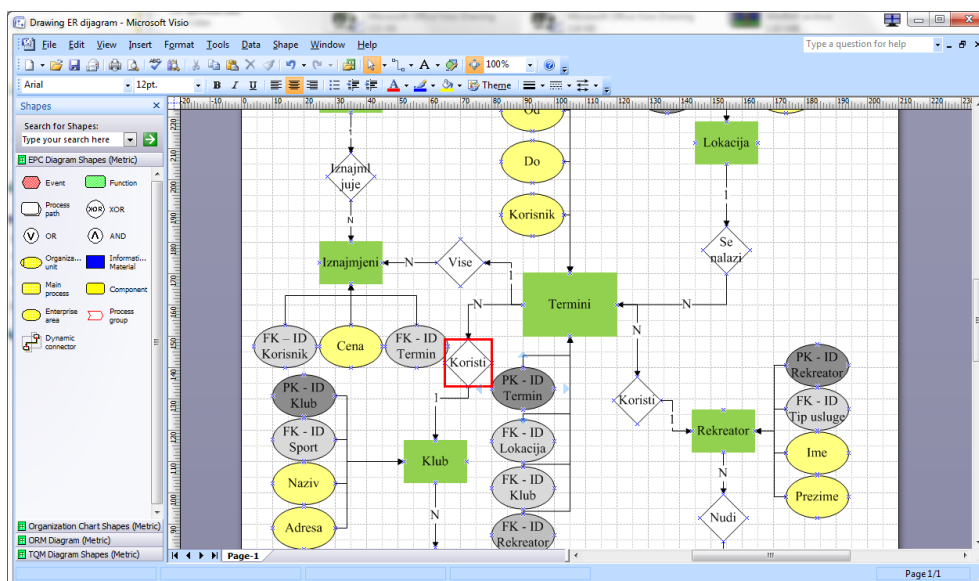
4.3.5 MS visio

Visio је CASE алат, у склопу MS office пакета, који у својој типичној инсталацији пружа могућности израде различитих модела [15]. Сви дијаграми припадају једној од 16 категорија, или типова модела. Кориснички оријентисани интерфејс, једноставно коришћење симбола за цртање и примери различитих модела омогућавају коришћење алата и особама које поседују мању количину знања о пројектовању информационих система. На слици 16. приказан је изглед почетног интерфејса за израду модела.



Слика 16. Кориснички интерфејс алата Visio

Израда модела података могућа је у категорији за израду дијаграма под називом Database цртањем Database Model Diagram-а. Дијаграм садржи стандардне опције за израду ентитета, атрибута ентитета, кључних атрибута ентитета и различитих врста веза. На слици 17. приказан је пример Entity Relationship Model (ERA) модела израђеног у Visio алату.



Слика 17. ER модел у Visio-у

Visio не пружа могућност израде репозиторија различитих модела, што значи да нема нити хоризонталне нити вертикалне повезаности модела. Пренос модела у друге програме омогућен је само у сврху презентације (word, excel, powerpoint) што значи да помоћу овог CASE алата није могуће нити аутоматско генерисање базе података ни обрнути поступак аутоматског генерисања модела података из базе података.

4.4 Развојно окружење информационих система - Microsoft Access

Microsoft Access, такође познат као Microsoft Office Access, је систем управљања базама података који комбинује релациони формат Microsoft Jet Database Engine¹⁸ са корисничким интерфејсом и алате за развој софтвера [19]. Садржан је у Microsoft Office пакету апликација, укључујући професионална и виша издања или продаја одвојено.

Microsoft Access складишти податке на основу приступа Access Jet Database Engine. Такође је могуће увозити и директно повезивати складиштене податке у другим апликацијама и базама података.

Програмери софтвера и база података могу да користе Microsoft Access за развијање апликација, а напредни корисници могу да користе за развијање софтверских апликација. Као и у другим Office апликацијама, Access је подржан

¹⁸ Је формат базе података по коме је изграђено неколико Microsoft производа.

од стране Visual Basic-a¹⁹ за апликације, објектно оријентисаног програмског језика који може да подржи различите објекте укључујући DAO (Data Access Objects), ActiveX Data Objects, и многе друге ActiveX Data Objects (ADO)²⁰ компоненте. Визуелни објекти који се користе у обрасцима и извештајима излажу своје методе и својства у програмском окружењу VBA (Visual Basic for Applications), и модули VBA кода се могу прогласити и назвати функцијама Windows оперативног система.

4.4.1 Примена

Поред коришћења своје датотеке за складиштење података, Microsoft Access се такође може користити као 'front-end'²¹ програм, док други производи делују као 'back-end'²² табеле, као што је Microsoft SQL Server и не Microsoft-ови програми Oracle²³ и Sybase²⁴ [19]. Више back-end извора може да користи Microsoft Access Jet Database (**accdb** и **mdb** формати). Исто тако, неке апликације као што су Visual Basic, ASP.NET²⁵, или Visual Studio .NET могу користити формат Microsoft Access базе података за своје табеле и упите. Microsoft Access такође може бити део сложенијег решења, где може да се интегрише са другим технологијама као што су Microsoft Excel, Microsoft Outlook, Microsoft Word, Microsoft PowerPoint и ActiveX Controls²⁶.

Access табеле подржавају различите типове поља, индекса и референцијалних интегритета, укључујући каскадне исправке и брисање. Access такође укључује интерфејс упита, обрасце за приказ унетих података и извештаје за штампу.

Задаци који се понављају могу да се аутоматизују преко макроа са опцијама point-and-click²⁷. Такође је лако поставити базе података на мрежу и омогућити вишекорисничко коришћење и ажурирање података без ремећења међусобних

¹⁹ Или VBA је примена Microsoft's-овог језика за програмирање Visual Basic 6 и његовог интегрисаног развојног окружења.

²⁰ Је софтвер креиран од стране Microsoft-а који прилагођава своју претходну Component Object Model (COM - бинарни интерфејс стандард за софтверске компоненте) и Object Linking and Embedding (OLE - је технологија која омогућава уграђивање и повезивање са документима и другим објектима) технологију за садржај преузет са мреже.

²¹ Прикупљање уноса у различитим облицима од корисника и обрада истих да се ускладе са спецификацијама да 'back-end' може користити.

²² Користи обрађене уносе од стране 'front-end'-а.

²³ Систем за управљање базама података.

²⁴ Софтвер за управљање, анализу и мобилисање информација користећи релационе базе података, аналитику и решење складиштење података.

²⁵ Је серверска страна интернет апликација намењена за развој динамичких интернет страница.

²⁶ Је софтвер који прилагођава своје раније технологије, (COM) и (OLE), за садржај преузет са мреже.

²⁷ Су акције корисника рачунара који померањем показивача до одређене локације на екрану (показујући), а затим кликом миша активирају одређени задатак.

послова. Подаци су закључани на нивоу записа који се знатно разликује од Excel-а који закључава читаву табелу.

Постоје темплејти базе података у оквиру програма као и темплејти који се могу преузети са Microsoft-овог сајта. Ове опције су доступне при покретању Access-а и омогућавају корисницима да побољшају базе података са унапред дефинисаним табелама, упитима, обрасцима, извештајима и макроима. Темплејти базе података подржавају VBA кôд, али Microsoft темплејти не садрже VBA кôд.

Програмери могу да креирају решење користећи програмски језик VBA, који је сличан Visual Basic-у 6.0 (VB6) и користи се кроз Microsoft Office програме као што су Excel, Word, Outlook и PowerPoint. Већина VB6 кôда се може користити у VBA. Напредни корисници и програмери могу да прошире основе решења крајњег корисника на професионално решење са напредном аутоматизацијом, валидацијом података, уочавању грешке, и вишекорисничку подршку.

4.4.2 Карактеристике

Корисници могу да креирају табеле, упите, обрасце и извештаје, и повезати их заједно са макроима [19]. Напредни корисници могу да користе VBA за писање корисних решења са напредном контролом података и контролом корисника. Access такође има функција креирање извешаја која може да ради са било којим извором података над којим Access има приступ.

Оригинални концепт Access-а је да крајњи корисници могу имати приступ подацима из било ког извора. Остале карактеристике укључују увоз и извоз података у многим форматима, укључујући: Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server, Oracle, ODBC итд. Такође има могућност повезивања са подацима у постојећој локацији и користити их за прегледање, прављење упита, уређивање, и извештавање. Ово омогућава промену постојећих података обезбеђујући да Access користи најновије податке. Такође може обављати хетерогена придружења између сетова података складиштена на различитим платформама. Access се често користи од стране људи за преузимање података из базе података на нивоу предузећа за манипулацију, анализу, и локално извештавање.

Ту је и формат Jet Database који може да садржи и апликацију и податке у једном фајлу. То чини Access веома погодним за дистрибуцију целе апликације ка другом кориснику, који може да је користи у неповезаним срединама.

Једна од предности Access-а из перспективе програмера је његова релативна компатибилност са SQL-ом²⁸ (Structured Query Language) - упити се могу видети као SQL извештаји, а SQL извештаји се могу користити директно у Macros²⁹ и VBA Modules³⁰ за манипулацију Access табела. VBA могу бити укључене у упитима.

Access нуди параметризоване упите. Ови упити и Access табеле могу бити навођене из других програма као што су VB6³¹ и . NET³² кроз DAO³³ или ADO.

Десктоп издања система Microsoft SQL Server могу да се користи са Access-ом као алтернатива за Jet Database Engine. Ова подршка је почела са MSDE (Microsoft SQL Server Desktop Engine, Microsoft Data Engine или Microsoft Desktop Engine)³⁴, претходном верзијом Microsoft SQL Server 2000 и наставља са SQL Server Express верзијама, SQL Server 2005 and 2008.

4.4.3 Microsoft Access Runtime - извршавање

Microsoft нуди бесплатне верзије Runtime у Microsoft Access-у, који омогућавају кориснику покретање десктоп Access апликације без потребе куповине или инсталирања пуне верзије Microsoft Access-а [19]. Ово омогућава програмерима да креирају Access базе података које се могу слободно дистрибуирати на неограниченом броју крајњих корисника. Runtime омогућава корисницима да прегледају, мењају и бришу податке, заједно са текућим упитима, обрасцима, извештајима, макроима и кодом VBA модула.

Runtime не дозвољава корисницима да мењају дизајн Microsoft Access објеката или кода. Runtime верзије су сличне њима одговарајућим пуним верзијама програма Access-а и обично су компатибилне са претходним верзијама. На пример, Access Runtime 2010 омогућава кориснику да покрене Access апликацију направљену верзијом 2010, као и 2007, до верзије 2000.

²⁸ Је програмски језик наменски дизајниран за управљање подацима у релационим систему база података.

²⁹ Скраћено „macroinstruction“, у рачунарству је правило или образац који одређује како би одређени улазни низ требао бити означен за замену излазне секвенце према дефинисаној процедури.

³⁰ Техника дизајнирања софтвера која наглашава посебне опције у независним модулима тако да свака омогућава извршавање жељене функције.

³¹ Програмски језик треће генерације и интегрисано развојно окружење корпорације Microsoft за свој програмски модел Component Object Model (COM-бинарни стандардни интерфејс за софтверске компоненте).

³² Софтвер развијен од стране Microsoft-а који ради првенствено на оперативном систему Microsoft Windows. Укључује велику библиотеку Framework Class Library (FCL-колекција за виšekратну употребу класа, интерфејса и вредности типова), и обезбеђује интероперабилност језика (сваки језик може користити код написан на другим језицима) у више програмских језика.

³³ Jet Data Access Objects, је превазиђен општи програмски интерфејс за приступ бази података

³⁴ Систем за управљање базама података.

4.4.4 Развој

Access складишти све табеле база података, упите, обрасце, извештаје, макрое и модуле у Access Jet database као један фајл [19].

За развој упита, Access нуди "Query Design", графички кориснички интерфејс који омогућава корисницима да изграде упите без познавања структурног језика упита. У дизајну упита корисници могу приказати извор података упита (који може бити табела или упит) и селектовати поље које желе вратити тако што ће изабрати и превући у мрежу. Придруживање се може подесити селектовањем жељеног поља у табели и превлачењем у другу табелу. Access омогућава корисницима преглед и манипулацију над SQL кодом по жељи. Свака Access табела, укључујући и повезане табеле из различитих извора података, може да се користи у упитима.

Access подржава креирање "pass through queries". Ови исечци SQL кода се могу бавити спољним извором података коришћењем ODBC (Open Database Connectivity)³⁵ конекција на локалном рачунару. ODBC конекција омогућава корисницима да комуницирају са подацима ускладиштеним ван Access програма без употребе повезане табеле или Jet-а. Корисници креирају „pass through queries“ упите коришћењем SQL синтаксе подржан од стране спољног извора података.

Приликом израде извештаја (у "Design View-у") допуне или измене контрола проузрукују извршења повезаних упита у позадини, а дизајнер је приморан да чека завршење процеса пре него што буде у могућности да направи још неку измену. Ова функција се не може искључити.

Не-програмери могу да користе макро функцију за аутоматизовање једноставних задатака кроз серију падајућим селекцијама. Макрои омогућавају корисницима лако повезивање команди, као што је покретање упита, увоз или извоз података, отварање и затварање форми, преглед и штампање извештаја итд. Макрои подржавају основну логику (IF услови) и могућност да позову и друге макрое. Макрои могу такође садржати под-макрое које су слични подпрограмима.

У складу са осталим производима из Microsoft Office пакета, други програмски језик који се користи у Access-у је VBA. VBA је сличан језику Visual Basic 6.0 (VB6) и код може бити смештен у модуле, класе, форме и извештаје. Да бисте креирали богатији, ефикаснији и одржив готов производ са добрим руковањем грешака, већина професионалних Access апликација развијено је коришћењем

³⁵ Је стандардни програмски посреднички језик за приступ системима за управљање базама података.

VBA програмског језика уместо макроа, осим где је интернет распоређивање услов.

4.4.5 Функције и употреба Access-а

Access је алат за управљање базама података који се може наћи у склопу пакета Microsoft Office [20]. Постоји потреба за алатом за управљање базама података као што је Access за све корпоративне и приватне организације и сл. где је велики број података са којим се рукује. Управљање базом података је процес манипулисања меморисаних података према потреби. Базе података су дизајниране да понуде организовани механизам за складиштење, управљање, и преузимање информација. У суштини управљање базама података укључује функције складиштење података и проналажење. У неким случајевима подаци у бази података могу се уређивати и манипулисати у складу са захтевима. Access је савршен алат да урадите све ове врсте функција.

4.4.5.1 Предности Access-а

Access има веома једноставан оперативни поступак у односу на друге алате за управљање базама података који су доступни. Такође, прилично је јефтинији и може да се инсталира на већини рачунара. Још једна предност је, што обезбеђује одличан графички кориснички интерфејс, док већина алата за управљање базама података пружају console operating interface (CUI)³⁶ интерфејс. Ова функција ствара одлично пријатељско окружење. Креирање табела за управљање базама података се лако постиже. Могућа је директна интеракција између креираних табела и едитовања истих. Креирање табела се врши директно куцајући име, имовину итд., то доводи до лакшег руковања, док други софтвери имају сложене синтаксе за креирање ентитета. За коришћење Access-а није потребно високо програмерско знање.

4.4.5.2 Посебне особине Access-а

Креирање табела се врши једноставно дизајнирањем или опцијом „wizard“-чаробњак. Подаци унети и сачувани у Access-у се могу лако извести или пренети у друге табеле или статички софтвер за даљу анализу. Као део Microsoft Office Suite-а, Access омогућава једноставну интеграцију са Word-ом и Excel-ом. Сви подаци сачувани у Access-у могу се лако користити кроз Microsoft Office апликације. На пример, имена и адресе које су ускладиштене у Access бази података могу да се увозе и користе у Word-у директно.

Права предност Access-а јесте његова релациона способност која омогућава корисницима да повезују табеле, коришћењем поља заједничка за обе табеле,

³⁶ Је средство за интеракцију са рачунарским програмом где корисник издаје команде на програму у облику узастопних редова текста (команда линија).

чиме се омогућава корисницима да добију податке из једне табеле који се односе на другу табелу. Подаци из засебних табела могу лако бити спојени да се произведе нова датотека за даљу анализу. Следећа функција под називом „Forms“ пружа могућност креирања корисничког интерфејса за унос података. „Report“ функција у Access-у је једна од најбољих на располагању. Сачувани подаци могу бити дати у одговарајућем формату погодни за штампање или у друге сврхе. Други аспект осигурања валидних података јесте уверење да нема дуплираних података. Развојно окружење које подржава Access обезбеђује брз развој апликације.

У принципу корисници са мало знања програмирања могу ефикасно користити Access. Постоје корисни менији попут File, Edit итд., да пружају корисницима веома ефикасну употребу доброг управљања базама података. Нема контролног прозора, нити сложених синтакси, и нема сложених упита за Access. То су заједничке карактеристике већине алата за управљање базама података. Тако је Access веома ефикасан управљачки алат за базе података без икаквих сметњи или нејасноћа приликом управљања истим. Како долази са Microsoft Office пакетом, врло је компатибилан за свакодневне потребе у канцеларији. Функције као што су унос података, складиштење и проналажење могу се лако урадити.

Самим тим се може рећи да је Access поуздан и веома ефикасан алат за управљање базама података.

4.4.6 Начин рада Access-а

Као и већина система за управљање базама података данас присутних на тржишту Access има неке функције овог апликационог генератора што увелико доприноси једноставнијем и бржем стварању апликација [21]. Писање кода за целу апликацију, ако се користи овај систем за управљање базама података, полако постаје прошлост. За захтевније апликације и даље остаје могућност дописивања кода ако је потребно (нпр. за размену података са другим Windows апликацијама).

Основа сваке апликације у Access-у је база података. Она може садржати следеће објекте:

- Табеле,
- Упите,
- Форме,
- Извештаје,
- Макрое,
- Програмске модуле.

Табела је део базе података у којој се чувају кориснички подаци. Како Access управља релацијским базама података између табела је могуће успоставити релационе односе.

Упит приказује податке из табеле по неком задатом критеријуму (услову). Упита постоји више врста па се поједине врсте користе према потреби. Тако је податке могуће приказати у облику у каквом се приказују и у табели (табеларни облик), могуће је приказати целу табелу или пак само њен део, могуће је поља груписати и приказивати тотале, приказивати податке из више табела у једном упиту итд. Посебно су интересантни упити који користе параметре које корисник сам задаје и на тај начин одређује које податке жели видети из табеле.

Форма се користи за манипулацију подацима из табеле или упита (унос, брисање и промене података). Могуће је и штампање форме на штампачу. Она може да садржи податке из више табела или упита, могућа је комбинација форма-подформа (истовремени приказ две форме), а могуће је имати више форми унутар једне форме. Да би се олакшало и убрзало креирање форме постоји тзв. wizard за креирање форме. Он нам нуди избор различитих облика форми (једноствна, форма-подформа итд.) са пет различитих приказа поља табеле или упита (увучено, уоквирено итд.). Након креирања форме на тај начин тој се форми по потреби додају разни облици контроле (нпр. за излаз из форме, избор једне од понуђених могућности итд.).

Форма може садржати и тзв. OLE објекат (објекат из других Windows апликација са којима Access може размењивати податке, нпр. слика из CorelDraw-а). Могуће је вршити и разна израчунавања унутар форме. Да би то направили брзо и тачно брине се тзв. wizard за креирање израза преко којег можемо изабрати било коју функцију коју Access нуди уз одговарајућу синтаксу што смањује могуће грешке на минимум.

Извештај се користи ако се подаци из табеле или упита желе одштампати на штампачу. Један извештај може да садржи податке из више табела или упита. Постоји више врста извештаја. Тако је могуће направити комбинацију извештај-подизвештај (истовремено два различита облика извештаја у једном извештају), а извештај може садржати и форму. Као и код форме за брзо креирање извештаја постоји тзв. wizard за креирање извештаја. Могуће је користити и wizard за креирање извештаја ако су нам те могућности потребне (нпр. за додавање текућег датум, броја странице итд.).

Макро наредбе користимо да бисмо аутоматизовали поједине задатке у обради података. Креирају се преко дешавања над објектима које нуди Access (дешавање је нпр. затварање неког прозора, извршење неке контроле у форми или извештају итд.). Помоћу макроа могуће је све објекте неке базе података спојити у једну заједничку апликацију. Такође, могуће је контролисати рад са другим објектима (отварање, затварање објеката и др.), а форме и извештаје је

могуће контролисати по деловима (секцијама). Треба напоменути да се макрои не креирају помоћу Access Basic-а, него слично као и већ описани објекти.

Програмски модули садрже функције, потпрограме или појединачне инструкције написане у Access Basic-у, програмском језику за програмирање унутар Access-а. По својој синтакси овај програмски језик је врло сличан Visual Basic-у, међутим подржава и неке посебне могућности (рад са објектима - од креирања, промене, до брисања појединог објекта, а преко њега се обавља и размена података са другим апликацијама (Dynamic Data Exchange (DDE)³⁷).

³⁷ Начин комуникације међу процесима.

5 Опис реалног система и апликације

5.1 Надлежности и послови реалног система

Спортски центар Трстеник основан је ради обезбеђивања права утврђених Законом односно задовољавања потреба грађана у области физичке културе, и то Одлуком Скупштине Општине Трстеник, дана 13.07.2005. године, а регистрован је у Привредном суду у Краљеву решењем бр. фи 84/06 дана 13.12.2006 [22].

Установа за спорт и физичку културу Спортски центар Трстеник је организована као јавна служба – установа, јединствена економско пословна целина, чији је циљ пружање услуга корисницима у делатностима за које је регистрована.

Делатности су:

- делатност спортских арена и стадиона,
- делатност марина,
- остале спортске активности,
- третмани за побољшање физичког стања и расположења,
- приређивање сајмова,
- остале услуге - рекламе и пропаганде,
- изнајмљивање спортске опреме,
- остали превоз путника у друмском саобраћају,
- ресторани,
- кантине,
- кампови,
- остала трговина на мало у специјализованим продавницама,
- уређења и одржавање паркова, зелених и рекреационих површина.

5.1.1 Организациона структура

Јавно предузеће Спортски центар Трстеник је самостална организација [22].
Органи Установе су:

- Директор,
- Управни одбор,
- Надзорни одбор.

У оквиру Установе за спорт и физичку културу СЦТ, формиране су Организационе јединице и то:

- Сектор заједничких послова,

- Креативни сектор,
- Сектор за хигијену,
- Сектор за техничко одржавање,
- Сектор за одржавање отворених терена.

Организационим јединицама руководе координатори служби. Координатори за свој рад директно одговарају Директору.

Спортски центар Трстеник има право коришћења објекта Спортске хале у којој се пружају услуге спортског центра Трстеник.

5.1.2 Спортска хала

Спортска хала је годинама једина спортска дворана у којој могу да се такмиче и да тренирају клубови из Трстеника [22]. Хала, површине 2822 m² изграђена је давне 1980.године. У оквиру хале налазе се велика сала, мала сала, теретана, део за стони тенис, бифе, куглана, свлачионице и канцеларије.

Корисници спортске хале су сви клубови на територији општине Трстеник и школе које организују такмичења, као и радне организације у рекреативним терминима.

Спортска хала представља најважнији спортски објект у оквиру спортског центра. Велика сала је универзалног карактера. Пружа одличне услове за спортске и културне догађаје. Тренинзи и такмичења одвијају се на предвиђеној температури и под одличним осветљењем. Гледаоци користе северну и јужну, знатно пространију трибину. Капацитет је 2200 посетилаца. Поред спортских такмичења велика сала се често претвара и у идеални концертни простор.

Мала сала у Хали поседује одличне услове за тренинге боксера и стонотенисера.

Теретана је опремљена потребном опремом и апаратуром која задовољава свакодневне потребе спортских клубова, спортских организација, као и физичких лица која рекреативно долазе.

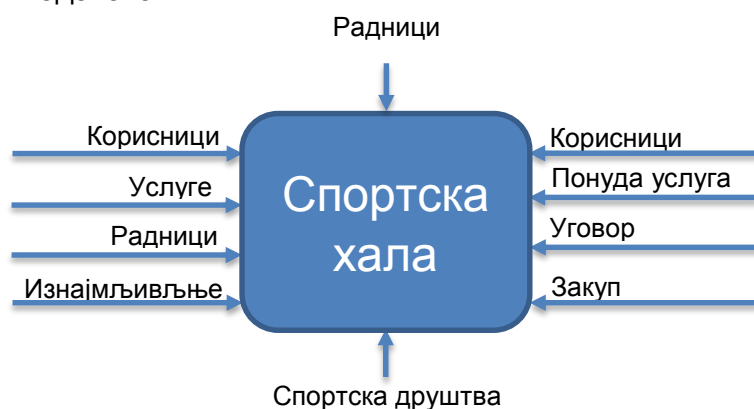
Спортисти и рекреативци имају на располагању довољан број свлачионица са свим потребним условима.

Основне услуге јавног предузећа Спортски центар Трстеник су издавање термина за тренинге и рекреацију клубовима, предузећима и физичким лицима у спортској хали Спортског центра Трстеник, као и пружање подршке у организацији спортских, културних и забавних садржаја [22].

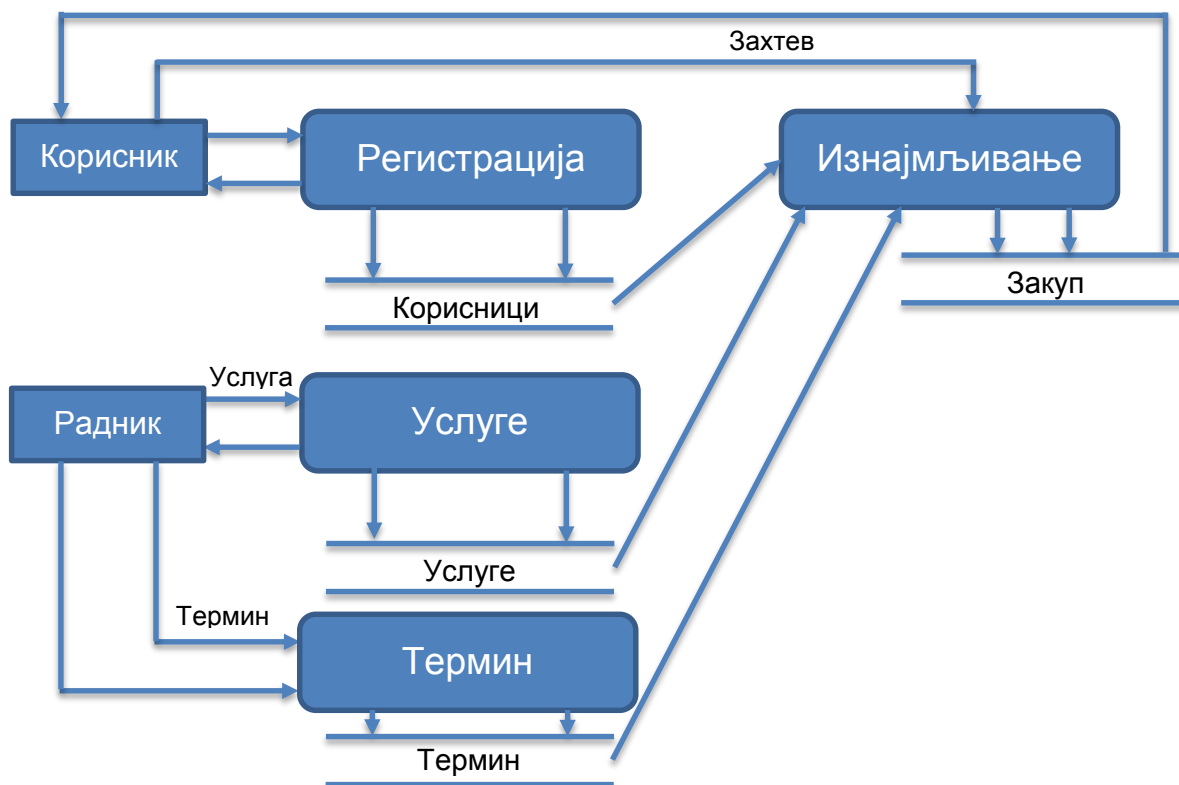
5.1.4 Структурна систем анализа пословног система и његова декомпозиција

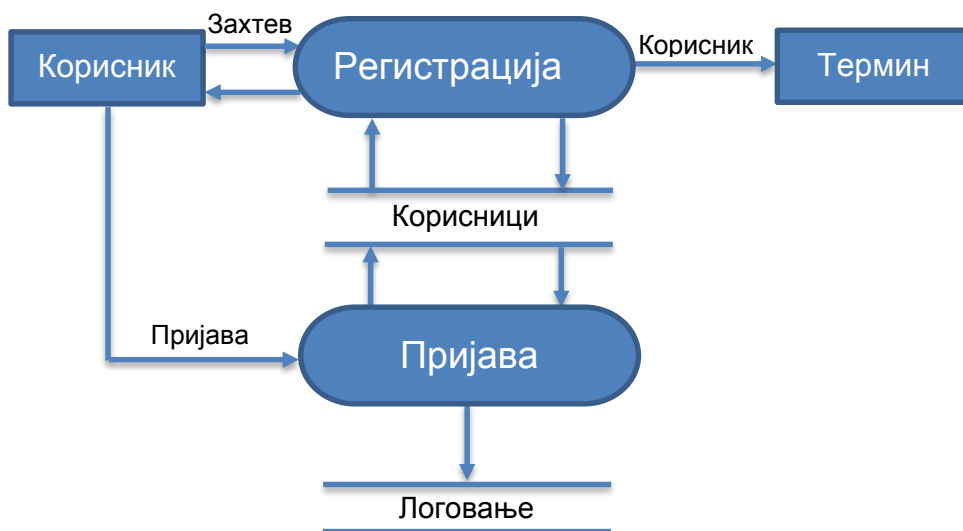
SSA је методологија за моделирање процеса организационог или технолошког система. Велики број CASE алата за пројектовања користи ову методологију. SSA методологија користи графички приказ за пројектовање информационог система. Разлози због којих је оваква методологија боља од текстуалне су: графички опис је ближи човеку; уклања се редуванса (понављање које је неопходно код текстуалног описа); текстуални опис се веома тешко преправља; тражење појмова у документацији је споро; итд. Основна два средства SSA су:

- Дијаграм тока података (DTP) и
- Речник података.

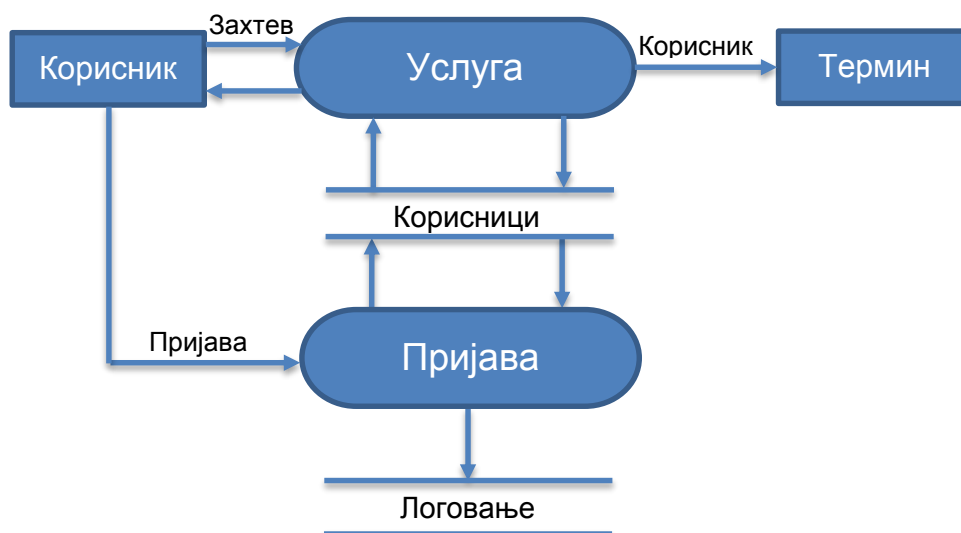


Слика 18. Дијаграм контекста информационог система Спортске хале

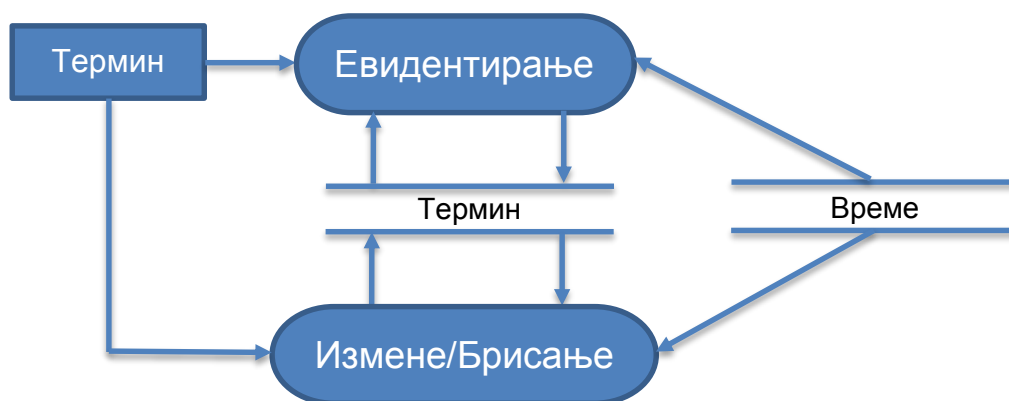




Слика 19. Декомпозиција процеса *Регистрација корисника*



Слика 20. Декомпозиција процеса *Евидентирање услуга*

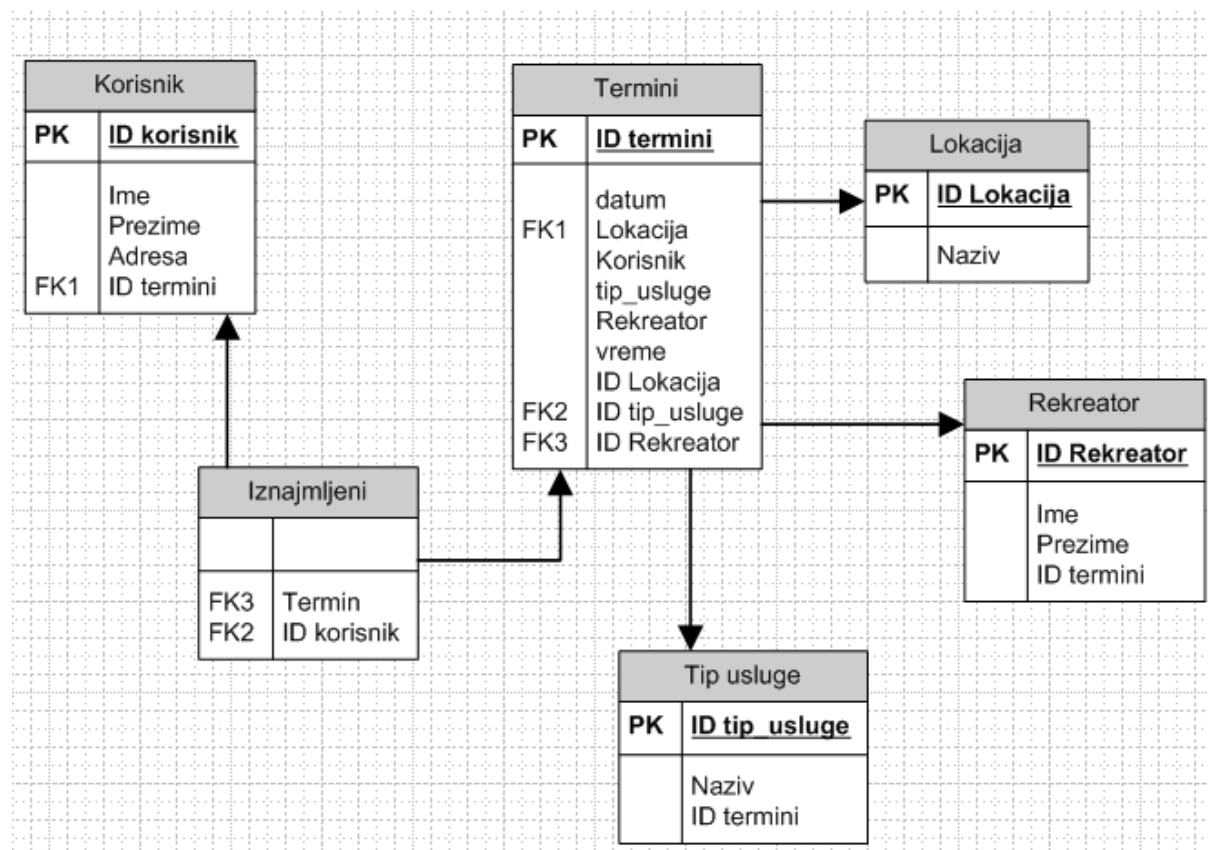


Слика 21. Декомпозиција порцеса *Евидентирање термина*



Слика 22. Декомпозиција порцеса *Изнајмљивање*

5.1.5 ER дијаграм



5.2 Речник података спортске хале

Корисник

(ID korisnik [PK]; Ime; Prezime; Adresa)

Изнајмљени

(ID Korisnik [FK]; ID Termin [FK]; Cena)

Термини

(ID Termin [PK]; datum; Lokacija [FK]; Korisnik; ID Rekreator [FK]; vreme)

Тип услуге

(ID Tip usluge [PK]; Naziv; Rekreator)

Рекреатор

(ID Rekreator [PK]; Ime; Prezime; ID Tip usluge [FK])

Локација

(ID Lokacija [PK]; Naziv)

5.3 Опис апликације

Информационе технологије или информационе и комуникационе технологије (ICT) играју значајну улогу у многим областима савременог друштва а самим тим и у спорту [23]. Информационе технологије створиле су велике промене и отворила нове могућности. Информациона технологија укључујући и интернет има огроман потенцијал да обогати животе људи. Ове технологије доприносе развоју и дистрибуцији информација, едукације, знања, искуства и идеја широм света. Информационе технологије такође помажу у смислу повећања ефикасности и брзине обављања циљева и задатака на радном месту и код куће.

У свим сегментима спортске индустрије информационе технологије су препознате као користан алат који омогућава јединствену могућност за побољшање и олакшавање рада спортске организације и њене администрације. Појам информационих технологија у спорту односи се на све ресурсе, процесе, поступке и системе за прикупљање података који могу бити корисни спортој организацији у испуњавању својих циљева и задатака.

У опису који следи, биће приказан један вид информационе технологију кроз информациони систем Спортског центра Трстенник. Информациони систем приказује предности које се стичу имплементацијом информационог система у један спортски објект.

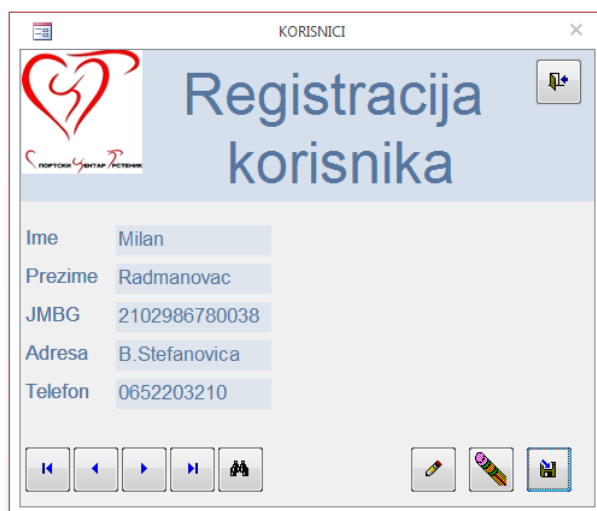
Покретањем апликације отвара се почетни прозор, слика 23.



Слика 23. Почетна форма

На прозору има увид у тренутно време и датум, што је потребно приликом избора слободног термина и датума одржавања. Пре него што се приступи резервацији термина, потребно је да у бази података постоји корисник који

жели да резервише термин. Уколико корисник није заведен у базу података, приступа се завођењу новог корисника, »Регистрација корисника«. Отвара се нова форма, слика 24.



Слика 24. Форма за регистровање корисника

На приказаном прозору је могуће извршити унос новог корисника кликом на



. Пре завођења новог корисника потребно је унети личне податаке о кориснику. По уносу података може се извршити процесирање и завођење



новог корисника кликом на

Поред уноса новог корисника могуће је извршити брисање непотребних података, корисника, са којима неће бити сарадње убудуће. Процес брисања



врши се кликом на



корисник који се брише. Такође је омогућен преглед заведених корисника путем навигационих тастера, као и претрага корисника (



) ради утврђивања постојања података одређеног корисника. У горњем десном углу, као и на другим формама, налази се иконица за излаз из форме,



Уколико је корисник који треба да резервише термин већ заведен, онда се приступа подформи »Претрага и резервација термина«, слика 25.

TERMINI

Pretraga i rezervacija termina

Unesi datum

Izaberi lokaciju

Pretraži

Upiši termin

Слика 25. Форма за претрагу и резервацију термина

По отварању радног прозора подформе, може се приступити прикупљању информација потребних за претрагу термина, датум и тражени објект. По задавању жељеног датума када је потребна услуга издавања и траженог објекта, иде се на »Претражи« и приказује се листа резервисаних термина за тај дан у траженом објекту, слика 26.

TERMINI

Pretraga i rezervacija termina

Unesi datum 02-Sep-14

Izaberi lokaciju teretana

Pretraži

Upiši termin

	datum	Lokacija	vreme	
+	02-Sep-14	teretana	08-09	
+	02-Sep-14	teretana	09-10	
+	02-Sep-14	teretana	10-11	
+	02-Sep-14	teretana	12-13	
+	02-Sep-14	teretana	14-15	
*				

Слика 26. Преглед заузетих термина

Уколико слободни термини ниси одговарајући, приступа се претрази за други датум (слободни термини су преостали термини који нису приказани као заузети у табели, слика 26.). Уколико је неки од слободних термина одговарајући, приступа се процесу резервације, »Упиши термин«, слика 27.

Слика 27. Форма за резервацију термина

Сада је потребно унети потребне податке како би се завршио процес резервације, слика 28.

	datum	Lokacija	vreme
+	02-Sep-14	teretana	08-09
+	02-Sep-14	teretana	09-10
+	02-Sep-14	teretana	10-11
+	02-Sep-14	teretana	12-13
+	02-Sep-14	teretana	14-15
*			

Слика 28. Форма са унетим подацима

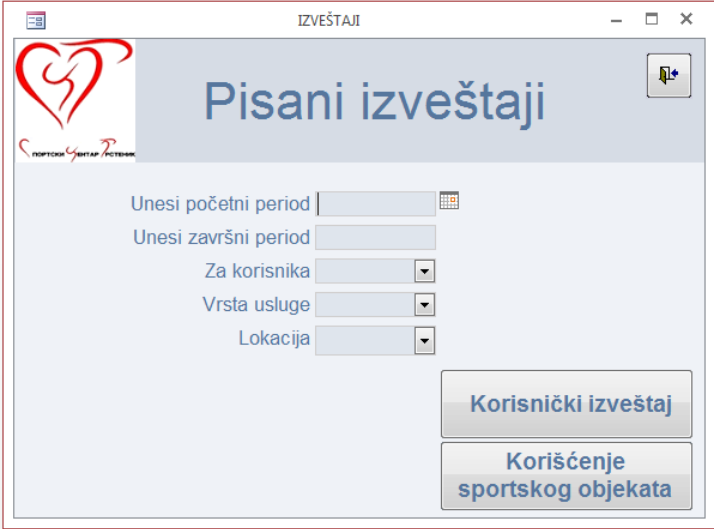
Када се унесу потребни подаци потребно је завршити процес резервације кликом на . Сада након поновне претраге термина за исти датум и локацију списак резервисаних термина биће проширен, слика 29.

	datum	Lokacija	vreme
+	02-Sep-14	teretana	08-09
+	02-Sep-14	teretana	09-10
+	02-Sep-14	teretana	10-11
+	02-Sep-14	teretana	12-13
+	02-Sep-14	teretana	14-15
+	02-Sep-14	teretana	20-21
*			

Слика 29. Резервисани термини

Поред могућности регистрације и брисања корисника и претраге и резервације термина, могуће је и штампање извештаја. Садржаји извештаја могу бити писани и графички. У наставку биће приказане могуће садржине извештаја.

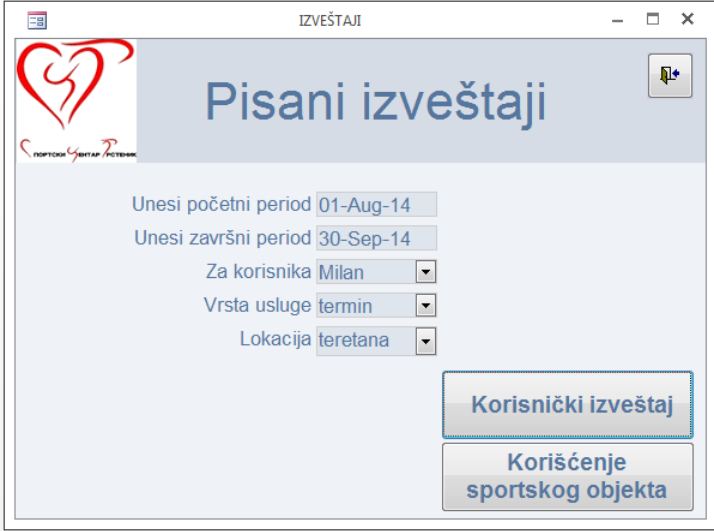
Приступа се подформи *»Писани извештаји«*, слика 30.



Слика 30. Форма за штампање писаних извештаја

Као што се види на радном прозору форме, потребни су одређени параметри на основу којих ће бити изведени одређени подаци из базе података који ће бити приказани на извештају.

Писани извештаји се односе на корисника појединачно, као и на коришћење изабраног спортског објекта. За кориснички извештај потребно је задати период за који се односи извештај, за ког корисника, која услуга је пружена и локација одржавања, слика 31.



Слика 31. Форма писаних извештаја са унетим подацима

По задавању потребних података иде се на *»Кориснички извештај«*, извештај изгледа као на слици 32.



Korišćenje usluga sportskog centra			
Korisnik		Milan	Radmanovac
Za period od		01-Aug-14	Do 30-Sep-14
Datum	Lokacija	Usluga	Vreme
18-Aug-14	teretana	termin	09-10
25-Aug-14	teretana	termin	09-10
28-Aug-14	teretana	termin	09-10
28-Aug-14	teretana	termin	20-21
01-Sep-14	teretana	termin	08-09
02-Sep-14	teretana	termin	08-09
03-Sep-14	teretana	termin	10-11
12-Sep-14	teretana	termin	14-15

Слика 32. Кориснички извештај

Када је у питању извештај који се односи на коришћење спортског објекта – локације, осим периода и локације други подаци нису потребни и клик на *»Коришћење спортског објекта«*. Извештај изгледа као на слици 33.



Korišćenje sportskih objekata

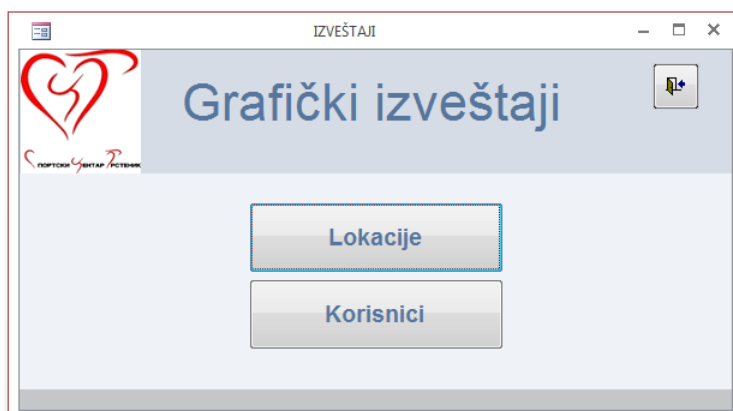
Objekat **teretana**

Za period od **01-Aug-14** Do **30-Sep-14**

Ime	Prezime	Datum	Vreme
Jovan	Ivovic	18-Aug-14	08-09
Milan	Radmanovac	18-Aug-14	09-10
Rade	Petrovic	18-Aug-14	10-11
Mila	Vlajkovic	18-Aug-14	11-12
Milan	Radmanovac	25-Aug-14	09-10
Jovan	Ivovic	25-Aug-14	10-11
Rade	Petrovic	25-Aug-14	14-15
Mila	Vlajkovic	25-Aug-14	15-16
Rade	Petrovic	28-Aug-14	08-09
Milan	Radmanovac	28-Aug-14	09-10
Jovan	Ivovic	28-Aug-14	10-11
Rade	Petrovic	28-Aug-14	12-13
Mila	Vlajkovic	28-Aug-14	14-15
Dragan	Stanojevic	28-Aug-14	16-17
Milan	Radmanovac	28-Aug-14	20-21
Milan	Radmanovac	01-Sep-14	08-09
Rade	Petrovic	01-Sep-14	10-11
Dragan	Stanojevic	01-Sep-14	12-13
Milan	Radmanovac	02-Sep-14	08-09
Mila	Vlajkovic	02-Sep-14	09-10
Mila	Radmanovac	02-Sep-14	10-11
Jovan	Ivovic	02-Sep-14	12-13
Mila	Vlajkovic	02-Sep-14	14-15
Mila	Vlajkovic	02-Sep-14	20-21
Milan	Radmanovac	03-Sep-14	10-11

Слика 33. Извештај коришћења спортског објекта

Поред писаних извештаја, информациони систем омогућује приказ графичких извештаја. На почетној форми активира се подформа *»Графички извештаји«* и отвара се форма, слика 34.



Слика 34. Форма за избор графичких извештаја

Као што се види на форми, има две врсте графичких извештаја. Разликују се у информацијама које се приказују, а тичу се корисника или локација.

Приступа се форми за приказ графичких извештаја који се односе на локације *»Локације«*. Отвара се форма, слика 35.

The image shows a web application window titled 'IZVEŠTAJI PROSTORIJA'. On the left side, there is the same logo as in the previous image. The main heading of the form is 'Izveštaj korišćenja prostorija'. Below the heading, there are three input fields: 'Unesi početni period' with a calendar icon, 'Unesi završni period', and 'Lokacija' which is a dropdown menu. At the bottom right, there are two rectangular buttons with rounded corners. The top button is labeled 'Za prostoriju' and the bottom button is labeled 'Sve prostorije'. The window has standard OS controls in the top right corner.

Слика 35. Форма за штампање графичких извештаја просторија

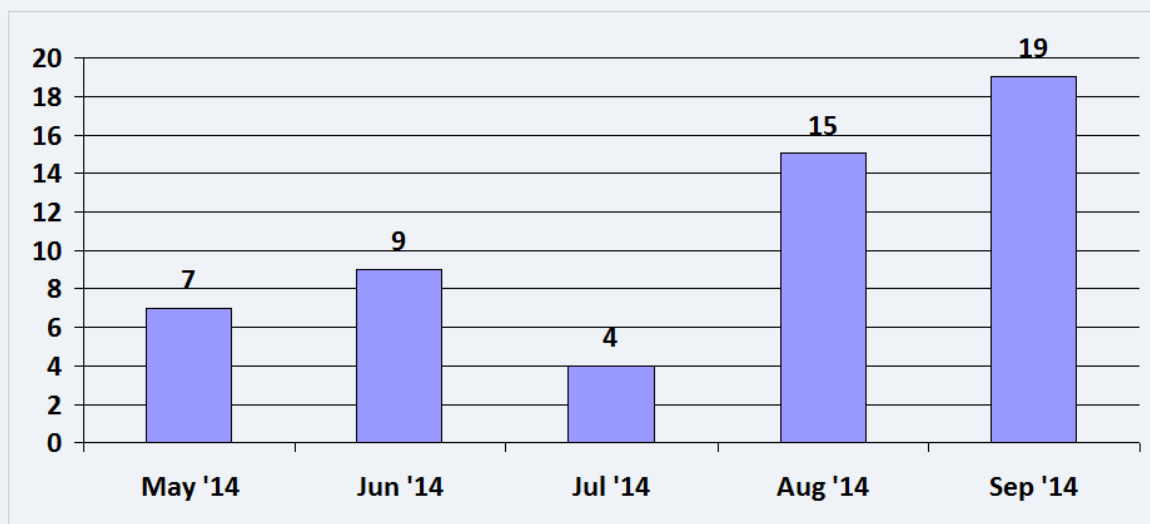
Потребно је унети период за који је потребан извештај, и изабрати локацију на коју ће се односити графички приказ. По уносу података, активирањем *»За локацију«* приказаће се извештај, слика 36.



Grafički prikaz korišćenja prostorije

Prostorija **teretana**

Za period od **01-May-14** Do **30-Sep-14**



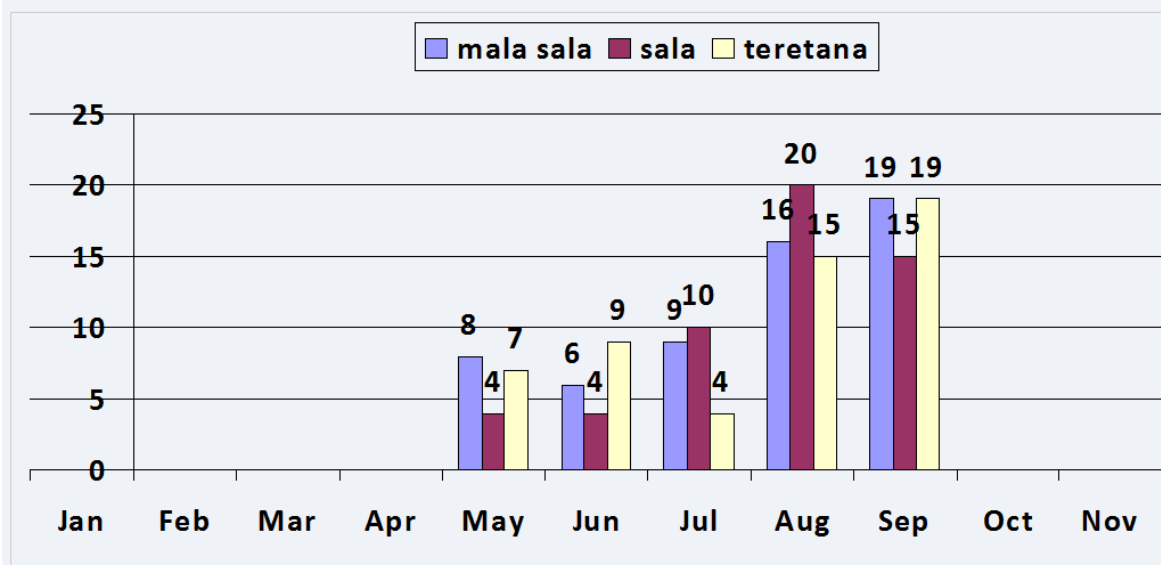
Слика 36. Извештај коришћења просторије

Уколико је потребно графички приказати коришћење свих локација, није потребно изабрати локацију, већ по задатом периоду приступити процесирању извештаја, »Све локације«. Извештај ће бити као на слици 37.



Grafički prikaz korišćenja svih prostorija

Za period od **01-May-14** Do **30-Sep-14**



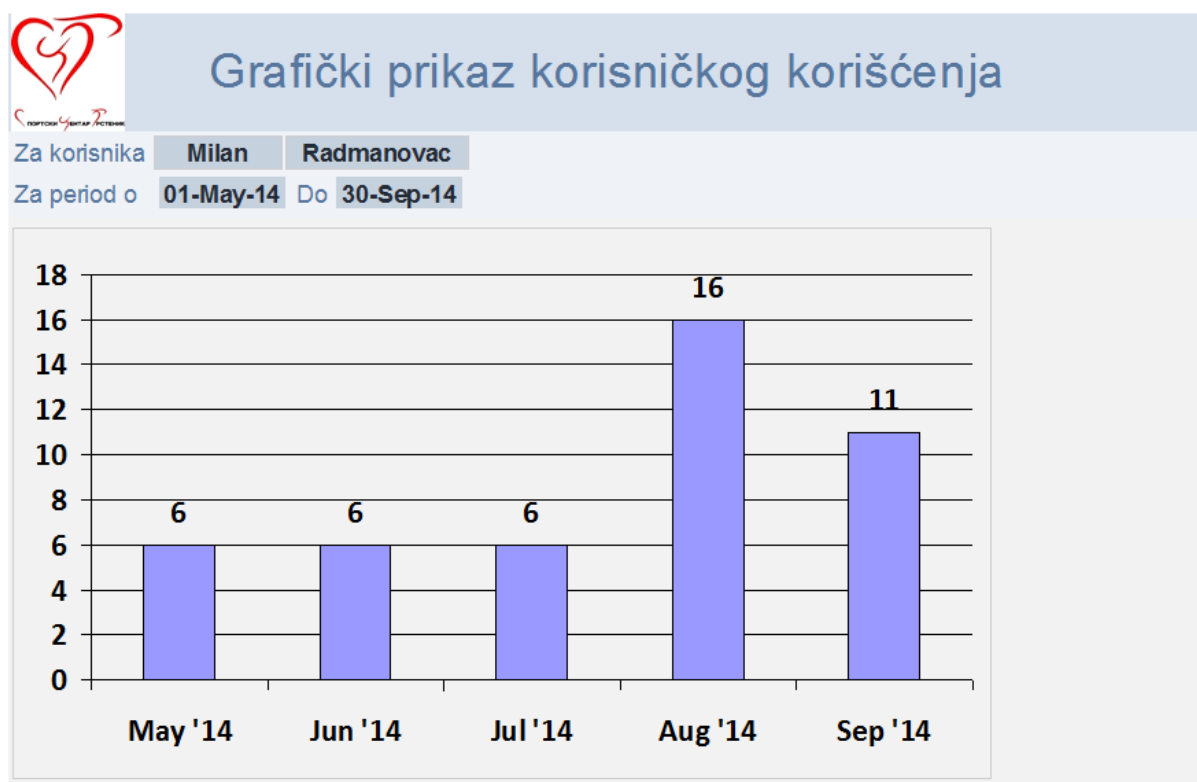
Слика 37. Извештај коришћења свих просторија

Сада се приступа форми за приказ графичких извештаја који се односе на кориснике, слика 34. »Корисници«. Отвара се форма, слика 38.



Слика 38. Форма за штампање графичких извештаја корисника

Потребно је унети период за који је потребан извештај, и изабрати корисника на кога ће се односити графички приказ. По уносу података, активирањем »Прикажи« приказати се извештај, слика 39.



Слика 39. Извештај корисничког коришћења

6 Закључак

У овом раду је приказан реални систем спортског центра. Рад се састоји из два дела - функционалног софтверског решења и документације везане за информационе технологије и информационе системе, животно циклус развоја система, моделе развоја информационих система, развојно окружење информационих система.

Развој информационог система обухвата целовиту и коректну израду решења за помоћ пословању увођењем информационо - комуникационе технологије, која ће на најбољи начин подржати пословне процесе и омогућити унапређење пословне технологије организације за коју се информациони систем развија.

У пројектовању и развоју информационог система неке организације могу се идентификовати три фазе развоја. Прва фаза обухвата израду плана информационог система. Друга фаза подразумева моделирање и генерисање базе података с развојем пословне апликације. Трећа фаза обухвата имплементацију пословне апликације и стављање новог информационог система у функцију.

Главни проблем који може бити решен имплементацијом информационог система јесте уклањање силне папирологије у предузећу која је потребна приликом размене добара са корисницима. Предузеће, као и корисници, добијају на времену приликом сарадње јер систем знатно брже информише о услугама које нуди предузеће. Самим тим се побољшава комуникација са корисницима приликом резервисања услуга спортског центра и стварају повољни услови за даљу сарадњу. Постојање корисника се лако и брзо проверава, као и завођење новог корисника. Поред регуларних извештаја могуће је графичким путем представљати извештаје који ће сликовито дати увид у коришћењу спортских услуга.

7 Литература

- [1]. <http://itinovacije.wordpress.com/2013/03/26/40/>
- [2]. http://sr.wikipedia.org/wiki/Informaciona_tehnologija
- [3]. **Арсовски З.(2000)** „Информациони системи“, Крагујевац.
- [4]. http://en.wikipedia.org/wiki/Information_systems
- [5]. <http://itrevizija.ba/wp-content/materijal/razno/PojamInformacionihSistemija.pdf>
- [6]. <http://is4biz.weebly.com/components-of-information-system.html>
- [7]. **Полишчук Е. Ј.(2007)** „Пројектовање информационих система“, Подгорица.
- [8]. Скрипта „Пројектовање информационих система и база података“.
- [9]. <http://www.buzzle.com/editorials/1-5-2005-63768.asp>
- [10]. <http://www.buzzle.com/editorials/4-5-2005-68117.asp>
- [11]. <http://www.buzzle.com/articles/waterfall-model-vs-v-model.html>
- [12]. <http://www.buzzle.com/editorials/1-13-2005-64082.asp>
- [13]. <http://www.buzzle.com/articles/comparison-between-waterfall-model-and-spiral-model.html>
- [14]. **Вељовић А.** „Развој информационих система и база података“.
- [15]. **Милошевић Б.(2013)** „CASE алати“, Висока техничка школа струковних студија, Ниш.
- [16]. **Мишковаћ В.(2011/2012)** „Основне функције и структура CASE производа“, Факултет за пословну информатику.
- [17]. <http://en.wikipedia.org/wiki/IDEF0>
- [18]. <http://en.wikipedia.org/wiki/IDEF1X>
- [19]. http://en.wikipedia.org/wiki/Microsoft_Access
- [20]. <http://bhashaindia.com/Patrons/HowTo/pages/Access.aspx>
- [21]. <http://tecajevi.freesevers.com/acnacin.htm>
- [22]. <http://www.sctrstenik.org.rs/index.html>
- [23]. http://www.savremenisport.com/Mediji_Informacione_tehnologije_i_sport.html
- [24]. <http://hr.wikipedia.org/wiki/Podatak>
- [25]. <http://ultimatesdlc.com/spiral-model/>