

UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO-SPRÁVNÍ

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ A INFORMATIKY

**Slabá místa při návrhu konkrétního
informačního systému
BAKALÁŘSKÁ PRÁCE**

AUTOR PRÁCE: Karel Štěpán

VEDOUCÍ PRÁCE: Ing. Hana Kopáčková PhD.

2008

UNIVERSITY OF PARDUBICE

**FACULTY OF EKONOMICS AND
ADMINISTRATION**

INSTITUTE OF SYSTEM ENGINEERING AND INFORMATICS

**Weak places in the design of a particular
information system
BACHELOR WORK**

AUTHOR: Karel Štěpán

SUPERVISOR: Ing. Hana Kopáčková PhD.

2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Karel ŠTĚPÁN**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Slabá místa při návrhu konkrétního informačního systému**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je popis postupu při návrhu konkrétního informačního systému a nalezení slabých míst, která mohou způsobit sníženou funkčnost systému.

Součástí práce je také návrh protiopatření.

Obsah bakalářské práce:

- 1) Vysvětlení základních pojmů
- 2) Charakteristika postupu návrhu informačního systému z teoretického hlediska
- 3) Popis postupu při konkrétním návrhu informačního systému
- 4) Nalezení slabých míst, která ovlivnila funkčnost
- 5) Návrh protiopatření

Rozsah grafických prací:


Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] ŘEPA, V. Analýza a návrh informačních systémů, Praha: Ekopress, s.r.o., 1999, ISBN 80-86119-13-0
- [2] VRANA, I., RICHTA, K. Zásady a postupy při zavádění podnikových informačních systémů, Praha: GRADA, 2004, ISBN 80-247-1103-6
- [3] KRÁL, J. Informační systémy, Veletiny: Science, 1998, ISBN 80-86083-00-4

Vedoucí bakalářské práce:


Ing. Hana Kopáčková, Ph.D.
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:


1. října 2007

Termín odevzdání bakalářské práce:

19. května 2008


prof. Ing. Jan Čapek, CSc.
děkan

L.S.


doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 30. října 2007

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Všechny literární prameny a informace použité v práci, jsou vypsány v seznamu použité literatury.

Seznámil jsem se s povinnostmi vycházející ze zákona č. 121/2000 Sb., autorský zákon, hlavně se skutečností, že Univerzita Pardubice může podle § 60 odst.1 autorského zákona uzavřít licenční smlouvu o užití této práce jako školního díla a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne _____

Karel Štěpán

Abstrakt

Úkolem této bakalářské práce je popis návrhu konkrétního informačního systému a nalezení slabých míst během samotného vývoje, která mohou zapříčinit nedostatečnou funkčnost systému. Navrhnout i dostatečná protiopatření, která by těmto problémům předcházela. V práci jsou popsány pravidla při stavbě informačního systému a jednotlivé etapy postupu vývoje IS/ STAG spolu s použitými nástroji.

Abstract

The task of this bachelor work is to describe the proposal of the particular informative system and find any weaknesses which could possibly come into existence during the development of the system and its usage. There is also the intention to suggest some counteractions which can anticipate those problems. There are rules and individual phases of development of IS/STAG described in this thesis altogether with all applicable instruments.

Obsah

Obsah	7
Seznam obrázků:	8
Seznam tabulek:.....	8
1. Úvod	9
2. Vývoj informačního systému:	10
2.1 Metodika = metodologie.....	10
2.2 Životní cyklus informačního systému	10
2.3 Životní cyklus IS versus životní cyklus projektu	12
Shrnutí smyslu metodiky tvorby IS	15
2.5 Metodika tvorby IS versus metody analýzy a návrhu IS	16
2.5.1 Obecné principy metodik tvorby IS	17
2.5.2 Typy životních cyklů vývoje IS	19
2.5.3 Metodiky vývoje a provozu IS můžeme členit do následujících skupin:..	22
2.5.4 Současné trendy ve vývoji metodik	22
3. Základní postup Vývoje IS/STAG	23
3.1 Principy metodiky	23
3.2 Definice vybraných základních pojmů.....	23
4. IS/STAG obecně	25
4.1 Historie IS/STAG	25
4.2 HW a SW IS/STAG.....	26
4.2.1 Vývojové oddělení.....	27
4.2.2 Oddělení serverů.....	28
4.2.3 Klientské stanice, tenký klient, partnerské VŠ	29
4.2.4 Case	29
4.2.5 Upgrade systému	29
4.2.6 Zabezpečení	30
5. Etapy vývoje IS/STAG.....	30
5.1 Informační strategie organizace	30
5.2 Úvodní studie systému	31
5.3 Globální analýza a návrh + Detailní analýza a návrh.....	33
5.4 Implementace	45
6. Slabá místa vývoje IS/ STAG a jejich protiopatření	49
7. Přílohy	53
Příloha 1: Funkční model procesu vývoje IS/STAG	53
Kontextový diagram vývoje IS/STAG	53
Příloha 2: Proces vývoje IS/STAG.....	54
Příloha 3: Kontextový diagram VŠ podle IS/STAG	55
Příloha 4: ERA model část 1	56
Příloha 4 ERA model část 2	57
8. Závěr.....	58
9. Zdroje:	59
Internet.....	59
Diplomové práce	60
Literatura	60
10. Seznam zkratk	61

Seznam obrázků:

Obrázek 1 Vztah mezi metodikou řízení projektu a metodikou vývoje IS [ŘEP1]	12
Obrázek 2 Vztah mezi pojmy [ŘEP1]	14
Obrázek 3 Model vývoje IS/IT [VOR1]	15
Obrázek 4 Základní vodopádový postup vývoje IS [ŘEP1]	19
Obrázek 5 Přírůstkový postup vývoje IS [REP1]	21
Obrázek 6 Historie IS/STAG[3]	26
Obrázek 7 Model IS/STAG[14]	32
Obrázek 8 Diagram hierarchické struktury [5]	33
Obrázek 9 FSD 1. úrovně [5]	34
Obrázek 10 FSD 2. úrovně Předměty [5]	34
Obrázek 11 FSD 2. úroveň Tvorba rozvrhu [5]	35
Obrázek 12 FSD 2. úrovně Předzáměr [5]	36
Obrázek 13 DFD subsystémy IS/STAG[5]	37
Obrázek 14 DFD 1. Předmět[5]	40
Obrázek 15 DFD 2. Rozvrh[5]	41
Obrázek 16 DFD Předzáměr[5]	42
Obrázek 17 DFD Předzáměr2[5]	42
Obrázek 18 Schéma studijních plánů [MICH1]	43
Obrázek 19 Formulář F2 Předmět (požadavky)[15]	46
Obrázek 20 Formulář F2 Předmět (literatura) [15]	46
Obrázek 21 Formulář F4 Vyučující [15]	47
Obrázek 22 Formulář F5 Individuální předzáměr[16]	48
Obrázek 23 Formulář F6 student (student) [18]	48
Obrázek 24 Formulář F6 student (referentka) [19]	49
Obrázek 25 Funkční model procesu vývoje IS/STAG [ŘEP1]	53
Obrázek 26 Proces vývoje IS/STAG [ŘEP1]	54
Obrázek 27 Kontextový diagram VŠ podle IS/STAG [ŘEP1]	55
Obrázek 28 ERA model part1	56
Obrázek 29 ERA model part 2	57

Seznam tabulek:

tabulka 1 Popis datových toků	37
-------------------------------------	----

1. Úvod

Informační systémy se v dnešní době staly naprosto nedílnou součástí našeho života. Setkáváme se s nimi v zaměstnání, ve školách, doma. Jejich historie se začala psát v podstatě ve stejnou dobu, jako historie samotné informační vědy. Od vzniku prvních nekonvečních informačních systémů po druhé světové válce uplynulo již mnoho desítek let a vývoj IS udělal obrovský skok kupředu.

Obecně se uvádí, že IS musí splňovat tyto základní funkce. Musí svým uživatelům poskytnout dostatečnou funkcionalitu, ať už se jedná o zajištění evidenčních, analytických, transakčních, kontrolních, rozhodovacích nebo plánovacích činností. Dále musí zajistit odpovídající dostupnost informací, jejich aktualizaci a správnost, vyvarovat se duplicit a přinést očekávané ekonomické efekty. Proto by vývojové týmy měly postupovat při navrhování a vývoji informačního systému podle určitých metodik, protože metodiky mají zaručit právě spolehlivost a funkčnost systému. Díky nim se sníží výskyt chyb v IS na minimum. Přesto se stává, že konečný IS nesplňuje očekávání uživatelů a proto dochází k následným úpravám a změnám v systému.

Pro svoji bakalářskou práci jsem si vybral informační systém STAG, protože s ním přijdu do styku skoro každý den. S IS/ STAG jsem se poprvé v životě setkal až na vysoké škole, Univerzitě Pardubice. V začátcích svého studia jsem měl problémy s orientací a s funkcemi jednotlivých modulů systému, ale každodenním používáním a opakováním jsem se i já s ním naučil zacházet, stejně tak jako desetitisíce dalších studentů v České republice.

Další důvod, proč jsem si vybral za téma IS/ STAG je i to, že IS/STAG v dnešní době nalezneme na jedenácti vysokých školách v České republice. Materiály pro svoji bakalářskou práci jsem čerpal nejen z University Pardubice, ale i z ostatních vysokých škol.

Cílem práce je sestavit návrh a postup vývoje IS/STAG a nalézt jeho slabá místa a pokusit se je odstranit.

2. Vývoj informačního systému:

Na počátku 80 let nastal v oblasti vývoje informačních systémů významný posun. V dnešní době s dostatečným odstupem je vidět, že bylo dosaženo kritické kvality v metodách, technologiích a v organizaci práce, nutných při vývoj IS. Hlavním projevem tohoto historického mezníku byl zrod dnes jich dobře známých systémů na podporu vývoje IS, systémů CASE.

Tyto systémy zastávali původně funkci týkající se bezprostředně vývoje IS, v dnešní době se počet funkcí markantně rozrostl a mají obrovský vliv i na plánování, provoz, údržbu a rozvoj systému. O tom nás přesvědčují pokroky jejich dalšího vývoje – od podpory v samotné vývojové fázi životního cyklu IS, k podpoře jak strategických rozhodování na počátku vývoje, tak k podpoře operativních činností, týkající se provozu systému, jeho řízením, zpracováváním případných změn a následným rozvojem systému.

2.1 Metodika = metodologie

Metodika tvorby IS se běžně nazývá pozadí, které zadalo podnět k vytvoření a k řízení změn v dalším vývoji. Obsahuje jak celkový pohled na průběh vývoje vzniku, tak existenci IS ve formě takzvaného Životního cyklu systému, tak rozpracování jednotlivých fází životního cyklu a konečně příslušné metody, techniky a nástroje používané v jednotlivých činnostech vývoje, provozu a údržby IS.

V metodice tvorby IS jsou zahrnuty všechny etapy, přístupy, zásady, postupy, pravidla, dokumenty, řízení, metody, techniky, nástroje.

Tento souhrn se zabývá celým životním cyklem IS. Metodiky vývoje a provozu IS jsou především nápomocny všem, kteří se podílejí na vývoji IS, ať už se to týká dodavatelů IS nebo naopak zadavatelů IS. Přesně určují kdo, kdy, co a proč má dělat během vývoje a provozu IS.

2.2 Životní cyklus informačního systému

Jako základní kámen všech úvah v metodikách je brána představa o uspořádání a obsahu tzv. životního cyklu IS. Z této představy potom vychází a dále s ní pracuje veškerý obsah metodiky. Jednotlivé potřebné dokumenty, cíle a specifika řízení, metody, techniky a nástroje jsou metodikou připojeny k jednotlivým etapám, fázím a krokům životního cyklu. Podstata těchto základních jednotek životního cyklu IS je nastavena díky rozdílným v jejich

obsahu. Jejich náplň je jiná pro každou fázi, etapu a krok. Začátky a konce etap, fází a kroků vývoje IS jsou základními klíčovými body postupu, milníky, v nichž metodika určuje způsob řízení postupu vývoje IS, proto se na ní klade velký důraz.

U každé etapy by měla metodika stanovit:

- Cíl etapy
- Účel a obsah etapy
- Předpoklady zahájení etapy
- Kriteria pro ukončení etapy
- Klíčové dokumenty etapy
- Kritické faktory etapy
- Činnosti etapy
- Návaznosti etapy

Všechny etapy jsou rozděleny na různé činnosti (fáze, kroky) a ty by měly být metodikou popsány:

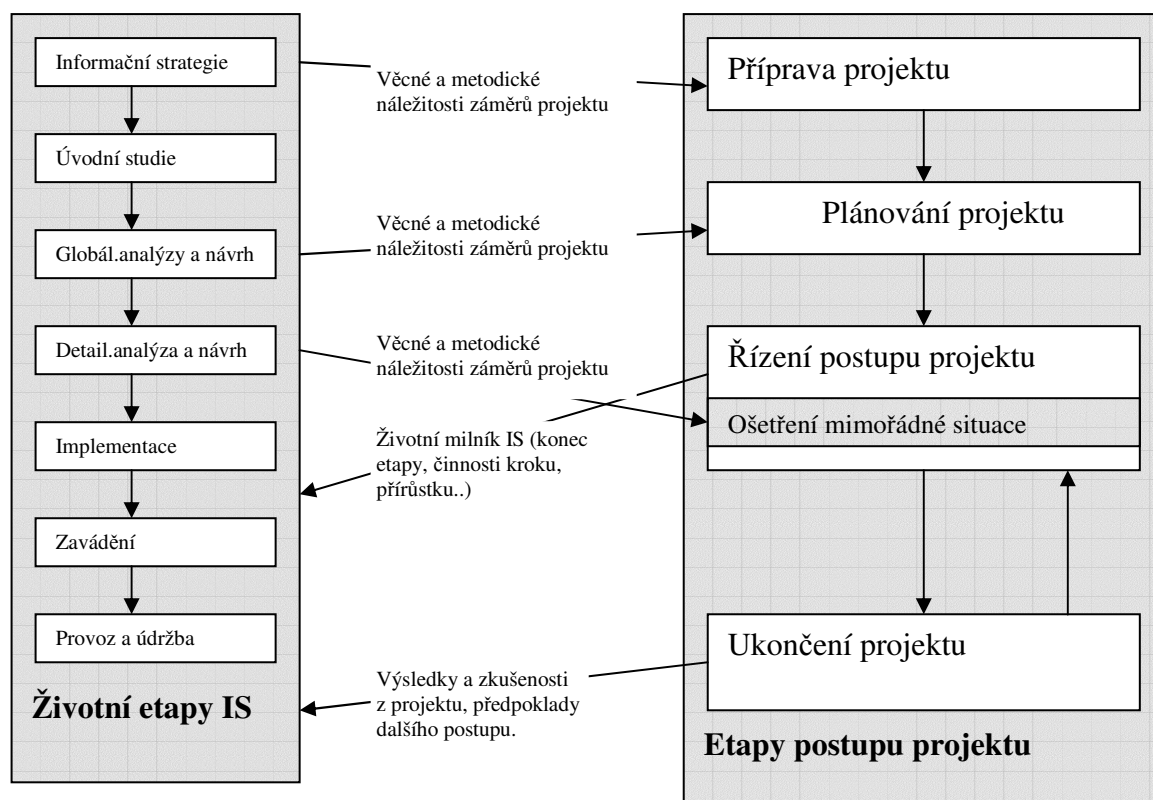
- Cíl činnosti
- Postup
- Vstupy
- Výstupy
- klíčové dokumenty
- ostatní produkty
- zúčastněné profese a odpovědnost
- doporučené techniky a nástroje

Vývoj IS, či jeho části je vždy projektem. Projekt, který má jasně definovatelný cíl, časově omezenou dobu a neopakovanou a s detailně stanoveným a omezeným rozpočtem a s přísnými požadavky na kvalitu. Tyto aspekty a podmínky je nutno v projektu a jeho postupech řídit.

Vlastní řízení projektu by se do jisté míry mohlo brát jako samostatný obor. Všechny projekty mají obecně stejný, základní životní cyklus:

- příprava a plánování projektu
- zahájení a operativní řízení postupu projektu
- ukončení projektu [ŘEP1]

2.3 Životní cyklus IS versus životní cyklus projektu



Obrázek 1 Vztah mezi metodikou řízení projektu a metodikou vývoje IS [ŘEP1]

Vztah mezi metodikou řízení projektu a metodikou vývoje IS lze porovnávat se vztahem mezi metodickou představou životního cyklu IS a metodickou představou průběhu jednoho projektu. Metodika vývoje IS se zabývá hlavně smyslem, obsahem a vzájemnými souvislostmi jednotlivých věcných činností, které jsou nutné v rámci vývoje IS provádět.

Pokud je obecný životní cyklus vývoje IS, sestavený metodikou vývoje IS a obecné schéma postupu je sestavené metodikou řízení projektu, vytvářejí se zde věci, které spolu musí platit současně, ale zároveň jsou zcela různé. Proto je vždy nutné rozlišovat mezi věcnými činnostmi projektu (metodikou vývoje IS stanovenými) a činnostmi jejich řízení (metodikou řízení projektů stanovenými).

Veškeré podstatný obsah životního cyklu IS je stanoven jeho metodikou vývoje. Přesný způsob naplnění života předepsaného metodikou záleží na konkrétních projektech, které jsou řízeny pravidly obecné metodiky řízení projektu. Jeden konkrétní projekt se týká zpravidla jedné části životního cyklu projektu.

Obě metodiky přicházejí do kontaktu už na samém začátku vlastní přípravy projektu. Hlavní úkol etapy příprava projektu je přednést věcný obsah projektu a zjistit, zdali je tento projekt proveditelný v daných podmínkách. K tomu slouží metodika vývoje IS, která zajistí smysluplnost a vztahy, které jsou potřeba k jednotlivým činnostem vývoje.

V etapě Plánování projektu se musí zpracovat konkrétní náplň projektu ať už věcnou, tak i časovou. Zde má metodika vývoje IS za úkol sestavit následné činnosti a jejich konkrétní požadavky na zdroje, parametry i spolu s možností alternativ posloupnosti činností a vzájemné náhrady zdrojů.

Účast metodiky vývoje IS, která je potřeba při plánování projektu, je potřeba i při samotném postupu projektu a to se týká i všech změn v plánu spojených s šetřením mimořádných situací. Jako řešení mimořádných situací bude vždy vznesena žádost na konkrétní změny v plánu projektu, které, aby se mohli uskutečnit, požadují dostatečné znalosti z oblasti metodiky vývoje IS.

Skončením projektu pokaždé dojde k uzavření určité části vývoje IS, etapy, skupiny etap, přírůstků a podobně. Z pohledu metodiky vývoje IS se jedná dosažení určitých cílů a milníků v životním cyklu IS. Metodika vývoje IS tyto milníky, jinak řečeno také místa, kde je ukončena jedna část vývoje IS, věcně definuje. Milníky zahrnují všechny konce etap životního cyklu IS a i místa uvnitř jednotlivých etap, která ukončují skupinu činností, fází, kroků atd., ze kterých lze vyčíst a vyjádřit nějaký konkrétní výstup. Metodika vývoje IS milníkům definuje jejich věcnou náplň, řídicí náležitosti a podmínky kvality, u kterých se musí zkontrolovat jejich dodržení a potřebu účasti příslušných rolí. V tomto bodě se setkává metodika vývoje IS s metodikou řízení projektů, která se řízením kvality zabývá obecně.

Etapa Ukončení projektu obsahuje konečné posouzení průběhu a veškerých výsledků projektu a má za úkol stanovit předpoklady pro další postup ve vývoji IS. Projektem získané zkušenosti jsou důležitým výstupem. Díky těmto zkušenostem vznikají pro další vývoj důležité úpravy ve vývoji IS a spolu s každým dokončeným projektem dochází ke zkvalitňování metodiky vývoje IS. [ŘEP1]

Jak bylo v předchozích odstavcích několikrát zmiňováno, metodiky jsou obsahově naplňovány jednotlivými:

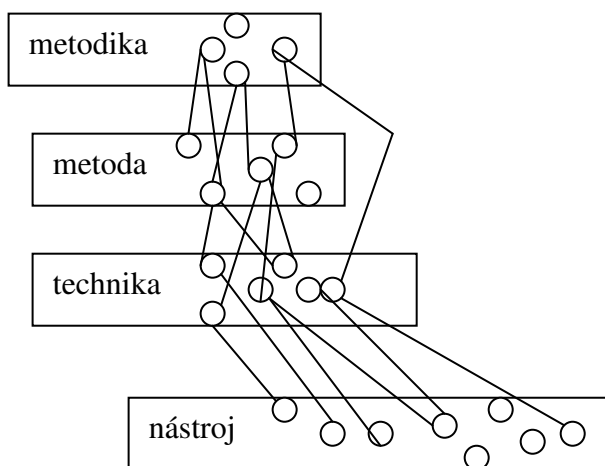
- a) Metodami
- b) Technikami
- c) Nástroji

a) Metodika tvorby IS: je doporučený souhrn etap, přístupů, zásad, postupů, pravidel, dokumentů, řízení, metod, technik a nástrojů pro tvůrce IS, který pokrývá celý životní cyklus IS. Určuje kdo, kdy, co a proč má dělat během vývoje a provozu IS. Metodika by se měla vztahovat na všechny prvky IS (pracovníky, organizační procedury, data,

SW, HW atd.), organizační vlivy IS, ekonomické otázky spojené s vývojem a provozem IS a doporučené dokumenty v případně způsob řízení v jednotlivých fázích životního cyklu IS.

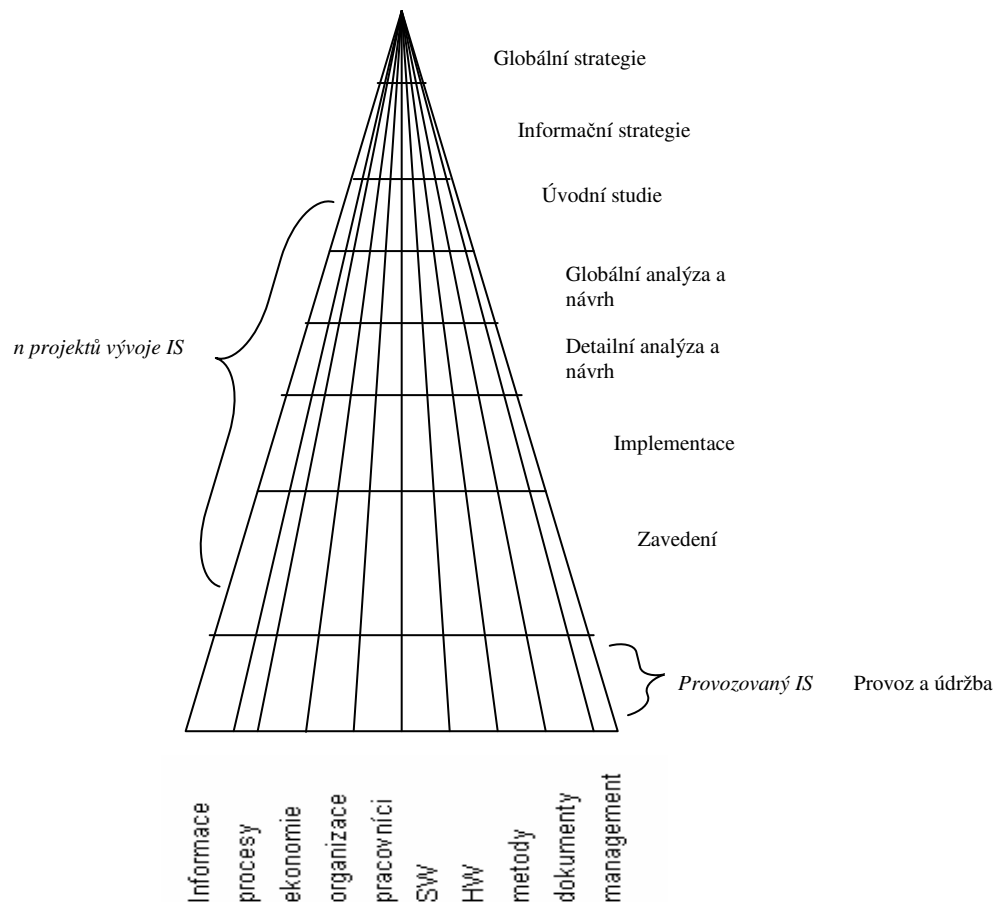
- b) **Metoda:** určuje, co je třeba dělat v určité fázi nebo činnosti vývoje či provozu IS. Metoda je vždy spojena s určitým přístupem, jako je funkční, datový nebo například objektový přístup. S přihlédnutím k této charakteristice řeší každá metoda postup činností v určité části procesu vývoje systému, nebo pouze z některého úhlu pohledu na systém
- c) **Technika:** určuje, jak se dobrat požadovaného výsledku. Zpravidla určuje přesný postup jednotlivých činností, způsob použití nástrojů, varianty rozhodnutí v určitých situacích a co z nich vyplývá, vymezuje obor své působnosti atd. Na rozdíl od metody je přesnější v závěrech a omezenější v okruhu použití.
- d) **Nástroj:** je prostředkem k uskutečnění určité činnosti v procesu vývoje a provozu IS a prostředkem k vyjádření výsledků této činnosti. Nástroj je často svázán s konkrétní technikou. Nástroje vždy formalizují vyjádření, proto je možné a žádoucí, aby byly v maximální míře automatizovány. [ŘEP1]

Není pravda, že by nějaké jednotlivé metody patřily určitým konkrétním metodikám nebo, že ke každé metodě, že je připravena určitá technika. Vztahy mezi metodami, technikami a nástroji, i jejich příslušností k metodikám, mohou být rozmanité, jak je znázorněno na obrázku 2.



Obrázek 2 Vztah mezi pojmy [ŘEP1]

Metodiky se odkazují na příslušné metody v relevantních místech životního cyklu IS, přičemž některé metody jsou více specifické, jiné mají univerzálnější charakter. Obdobně i technika může patřit k určité metodě, nebo je společnou pro řadu různých metod. Technika buď vyžaduje specifický nástroj, který je s ní pevně svázán a bez ní nemá smysl, nebo používá obecněji použitelný nástroj. Některé nástroje jsou natolik universální, že je naprostý nesmysl je vázat k nějakým konkrétním technikám.



Obrázek 3 Model vývoje IS/IT [VOŘ1]

Shrnutí smyslu metodiky tvorby IS

Z obecného hlediska patří mezi základní a zároveň nejdůležitější přínosy zavedení metodiky při vývoji a tvorbě IS především zvýšení kvality vyvíjeného IS a tím pádem i zvýšení jeho prestiže a konkurenceschopnosti na trhu. Tohoto základního přínosu zaváděné metodiky tvorby programových systémů je dosahováno tím, že metodika umožní:

- Dosáhnout vyšší produktivity a spolupráce projektových týmů, umožněné zejména jasným definováním jednotlivých etap životního cyklu vývoje IS činností v rámci těchto etap a jejich návazností atd.

- Vytvořit přesně definovaný komunikační standard pro každý typ funkčních míst a to jak tvůrců, tak i uživatelů systému.
- Na jednotlivé etapy životního cyklu vývoje IS zajistit dostatečnou specializaci všech zúčastněných ať už to jsou projektové týmy nebo jednotliví pracovníci uvnitř týmu. Získat větší nezávislost vyvíjených nebo již vytvořených IS na konkrétních řešitelích. Dosáhnout větší pružnosti vytvářených IS, aby se daly naplňovat potřeby uživatelů. Předložit kvalitní a hlavně aktuální dokumentaci vyvíjeného IS. Dosáhnout rychlejší a efektivnější údržby dokončených IS. Vymežit kritéria kvality vytvářených IS pro každou etapu životního cyklu vývoje IS.
- Efektivně a kvalitně řídit, plánovat a kontrolovat práci na projektech. Snížit rizika vyčerpávání jak finančních, technických, tak i personálních zdrojů. Tedy věnovat zvýšené úsilí počátečním etapám životního cyklu vývoje IS, kde se právě rozhoduje o realizovatelnosti a nerealizovatelnosti daného IS.
- Získat možnosti přenositelnosti vytvořených IS do jiných implementačních prostředí a to hlavně za pomoci oddělení čistě obsahových charakteristik systému od charakteristik, daných použitou technologií a implementačním prostředím.
- Získat přesně definovaná kritéria a východiska pro výběr a použití prostředků CASE, vzhledem k metodikou určené množině metod, technik, nástrojů a jejich vzájemných vazeb.
- Jasně určená metodika, která je dodržovaná jednotlivými projekčními týmy, patří k základním sjednocujícím a řídicím nástrojům tvorby, používání a rozvoje IS.
- Metodika má za úkol hlavně návod postupů týkající se tvorby IS. Definuje co, kdo, kdy má dělat a s jakou odpovědností v procesu tvorby IS.
- Doplněním o kritéria a dohody patří do kapitoly zavádění metodiky, kdy dojde k její realizaci v konkrétním prostředí pilotního projektu. Výběr a rozvoj metodiky začíná již v etapě jejího zavádění, tedy během realizace pilotního projektu. Když už dojde ke zkonkrétnění metodiky v projektu a prostředí vzniká technologie tvorby IS. Ale současně není možné přestat pracovat s metodikou na úrovni obecnosti, kvůli možnosti promítání změn v implementačním a technologickém prostředí vyvíjených systémů a současně změn v pravidlech, přístupech a metodách. [ŘEP1]

2.5 Metodika tvorby IS versus metody analýzy a návrhu IS

Během vlastního vývoje metod a technik analýzy a návrhu IS, se dospělo postupně k tomu, že hlavně technické pojetí vývoje IS už nestačí a je třeba ho rozšířit o další aspekty

vývoje IS. Vznikla tak univerzálnější metodika vývoje IS, která v sobě zahrnuje veškeré aspekty, tedy etapy, přístupy, zásady, postupy, pravidla, dokumenty, řízení, metody, techniky a nástroje, které pokrývají celý životní cyklus IS.

V tomto pojetí je metodika tvorby IS souhrnem. S metodikami vývoje a provozu IS pracují hlavně členové vývojového týmu. Metodika by se měla zabývat všemi prvky IS, ekonomickými otázkami týkající se vývoje a provozu IS a příhodnými dokumenty a případně způsobem řízení v jednotlivých fázích životního cyklu IS.

2.5.1 Obecné principy metodik tvorby IS

Bez ohledu na různé možné účelové varianty, by měla každá metodika tvorby IS vycházet vždy z následujících základních principů:

1. Orientace na cíle a problémy. Základem při tvorbě IS je hledání požadavků a problémů v organizaci a teprve z toho sestavování cílů IS. Uskutečnění tohoto pravidla je hlavním účelem počátečních etap vývoje IS.
2. Účast zadavatele projektu. Vedení organizace, pro kterou je IS vyvíjen, se musí dostatečně aktivně zapojovat do vývoje IS, musí dostávat průběžně informace o problémech a tyto problémy řešit po vzájemné dohodě s vedením vývoje IS, protože odpovědnost za obsah IS leží na straně zadavatele. Bez dodržení tohoto principu nelze předpokládat úspěšnou realizaci IS.
3. Klíčové dokumenty a jejich schvalování. Výsledky činností během vývoje jsou zaznamenány písemně do dokumentů. U Některé z těchto dokumentů nesou funkci takzvaných klíčových dokumentů, u nich je nutné, aby byly schváleny jak vedením vývoje, tak i vedením ze strany zadavatele. Chybí-li toto schválení a potvrzení správnosti dokumentu, není možné ve vývoji systému dál pokračovat. Díky tomuto principu dochází k zapojování vedení organizace, popsané v bodě 2.
4. Zapojení uživatele do návrhu. Stejně jako vedení organizace i zástupci budoucích uživatelů jsou členy projekčních týmů a zastávají funkci rozhodovací a tím pádem se podílejí na spoluodpovědnosti za výsledek.
5. Modelování a abstrakce, princip tří architektur systému. Vytváření tří úrovní modelů systému vede k jasnému oddělení podstaty systému od omezení, která jsou do návrhu přidávána v důsledku zvolené technologie a implementačního prostředí. Uskutečnění tohoto principu je zajištěna prostřednictvím používaných metod návrhu IS, podporovaných v maximální míře automatizovanými nástroji, a vede ke snížení rizika nepříznivých dopadů změn na návrh systému.

6. Ověřování a testování návrhu během celého vývoje. Po ukončení jakékoliv činnosti se důsledně ověřuje, zdali výsledky uskutečněné činnosti splňují cíle, které si organizace a IS a uživatelé systému zadali a jestli jsou tyto výsledky správné po formální i logické stránce. K testování a ověřování systému dochází již od počátečních etap vývoje. Tento způsob ověřování se uskutečňuje díky řídicí komisy projektu a dále také konceptem a průběžným upřesňování schvalovacího postupu.
7. V každé etapě probíhá analýza i návrh. V každé etapě se analyzují požadavky na systém a zachází se jen do takových detailů, aby bylo možné pomocí této analýzy navrhnout systém tak, aby mohla být zahájena následující etapa. To zapříčiní, že nedochází ke zbytečnému zabíhání do podrobností už na začátku vývoje a eliminuje se tak riziko pozdějších změn požadavků na systém. Tyto požadavky se v dalších etapách pouze zpodrobňují. Pokud dojde ke změnám v nadřazeném požadavku nebo se přidá další požadavek, musí se vývojový tým vždy vrátit k předcházející etapě, kde se požadavek poprvé projeví jako změna a z ní dále plynou další změny týkající se všech etap následujících. Teprve poté lze pokračovat v dalším návrhu.
8. Vývoj probíhá z hlediska všech úhlů pohledu na systém. V každé etapě je nutné analyzovat a navrhovat na odpovídající úrovni podrobnosti:
 - Data
 - Funkce
 - Organizace
 - Sociální a psychologická hlediska
 - Technologie
 - Ekonomická hlediska

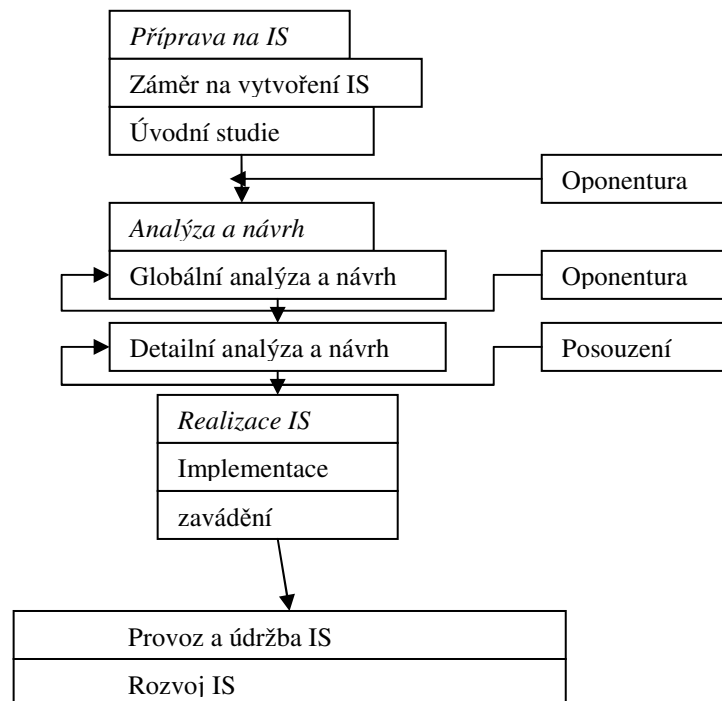
Prověření návrhu z hlediska všech těchto úhlů pohledu je jedním z měřítek úplnosti návrhu.

9. Otevřenost metodiky. Metodika tvorby IS se staví hlavně na běžně používaných a ověřených metodách a technikách vývoje IS a přesto by měla být schopna absorbovat veškeré nové poznání v tomto odvětví. To umožňuje hlavně široké využití dostupných podpůrných počítačových prostředků vývoje IS. Současně to ale znamená, že každá metodika by měla být schopná do určité míry spolupracovat s většinou standardních metodik vývoje IS.[ŘEP1]

2.5.2 Typy životních cyklů vývoje IS

K tomu, aby metodika splňovala výše zmíněné základní principy, především taky aby pokrývala veškeré aspekty vývoje IS a aby byla otevřená a nezávislá, musí hlavně určit základní životní cyklus vývoje IS.

Životní cyklus ohraničuje základní etapy vývoje IS a jejich obsah. Zpravidla však nelze uskutečnit vývoj IS postupně po jednotlivých etapách, jedná se o takzvaný vodopádový způsob. Tento typ je charakterizován následovně. Využívá ruční programování (bez nástrojů typu CASE), metodologie strukturovaného programování, řízení projektu pomocí CPM, PERT, pracuje s databázovými systémy, společně používá on-line a batch zpracování, aplikace jsou vyvíjené pro centrální počítač, programování se účastní programátoři profesionálové, používají se SW nástroje jako generátory sestav, existuje systém schvalování dílčích výsledků i systému jako celku, uživatel se účastní pouze při definování požadavků a při zavádění. [20]



Obrázek 4 Základní vodopádový postup vývoje IS [ŘEP1]

Používanějším postupem vývoje IS místo vodopádového postupu je přírůstkový postup. Přírůstkem se chápe dokončená a relativně uzavřená část systému, kterou lze samostatně navrhnout, implementovat a uvést do provozu přičemž funkčnost jiných přírůstků, které byly dokončeny dříve, zůstává zachována.

System se potom vyvíjí postupně, jeden přírůstek za druhým. Každý takovýto přírůstek vlastně žije ve svém životním cyklu a tyto různé přírůstky se vzájemně prolínají. Například části, které jsou pro vývoj důležité, a nebo jejich uskutečnění je snadné, či to jsou jasně neměnné části systému, tak ty jsou obvykle rychle uvedeny do podoby implementace a jsou zavedeny do provozu, zatímco jiné části systému jsou například jenom ve fázi úvodní studie, či globální analýzy. Proto se dá také lépe vypořádat s postupnými změnami a navyšováním uživatelských požadavků, než při celkovém jednorázovém vývoji, kde chybí taková ta flexibilita okolo změn návrhu. A z praxe, z uvedení přírůstku do provozu se získá spousta poznatků a zkušeností, ze kterých se pak dále vychází.

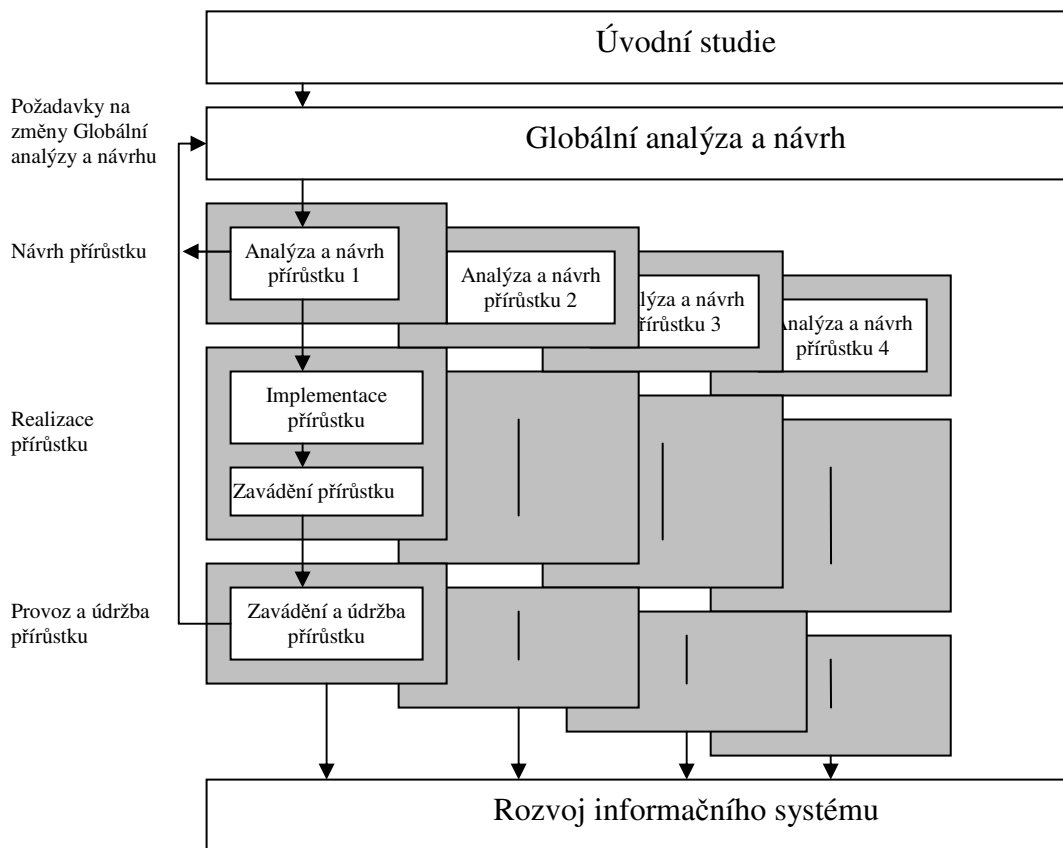
I když jsou zřejmé klady, přínosy a nutnost přírůstkového postupu vývoje IS, tak je také důležité a obtížné správně rozdělit samotné přírůstky. Každá technologie si určuje vlastní postup a pořadí následujících činností:

1. identifikace základních jednotek fungování,
2. stanovení těch úseků systému, které nejsou obsáhlé a náročné a mohou být pro uživatele viditelný užitek už v krátké době,
3. definování částí systému od nejvýznamnějšího po méně významné z pohledu podniku, firmy, organizace,
4. určení priorit přírůstků podle priorit požadavků, které naplňují,
5. určení pořadí přírůstků podle toho, jaké kladou nároky na řízení rizika, za předpokladu, že nejrizikovější se navrhnou a realizují nejdříve,
6. určení pořadí přírůstků podle pořadí konceptu dat a realizace datové základny tak, aby na řadu přišly nejprve ty přírůstky, které vytvářejí entity před těmi, které s entitami pracují,
7. určení pořadí přírůstků podle optimální posloupnosti nahrazování používaného systému systémem novým,
8. určení pořadí přírůstků podle vhodného pořadí dodávek jednotlivých částí systému

Při určování vhodného pořadí dodávek se rozhoduje podle následujících faktorů:

- priority věcné oblasti s ohledem na rychlou návratnost investic nebo realizaci přínosů pro věcnou oblast,
- rychlé nahrazení stávajících systémů, které neefektivně podporují věcné činnosti,

- případný požadavek minimalizovat rozsah rozhraní a přemostění, které je pro implementaci požadovaných přírůstků potřeba vytvořit,
- případný záměr implementovat nejmenší možné přírůstky v co nejkratší době,
- optimální sekvence vývoje vzhledem k datům (nejprve systémy, které vytvářejí entity, pak až systémy které používají minimalizaci požadovaného rozhraní, přemostění a převodů dat),
- případný požadavek snížení počtu potenciálně paralelních přírůstků minimalizovat dodatečné náklady spojené s řízením, rizikem a složitostí,
- případné rozhodnutí předcházet syndromu budoucího šoku, kdy je nový přírůstek dodán pře tím, než je předcházející přírůstek v organizaci plně integrován a přijat,
- možné externí vlivy (hardware, software),
- možné změny ve vývoji věcné oblasti, které mohou mít vliv na realizovatelnost nebo rozsah přírůstku,
- snaha udržet zainteresovanost, zájem a účast uživatele. [ŘEP1]



Obrázek 5 Přírůstkový postup vývoje IS [ŘEP1]

2.5.3 Metodiky vývoje a provozu IS můžeme členit do následujících skupin:

- a. mezinárodní: využití a funkci mají stejnou jako státem podporované IS, ale používají je hlavně nadstátní a státní úředníci
 - Euromethod (evropské společenství)
 - Software Life-Cycles Process-IS
- b. firemní: patří do vlastnictví firem, škol, institutů atd. hlavní zájem je především specifický postup vývoje IS a tomu příslušné nástroje a pravidla.
 - ORACLE CASE
 - RAP
 - IEM
- c. státem podporované: státem kontrolované, pro státní úředníky, je zde vidět snaha o vzájemné sjednocení vyvíjených a vzájemně souvisejících systémů, aby v konečném výsledku tvořily systém
 - SDM
 - V-MODEL

2.5.4 Současné trendy ve vývoji metodik

Metodika vývoje IS se musí průběžně aktualizovat a obnovovat, aby zachytila změny v IT a v obecných přístupech k chápání analýzy a návrhu IS.

- Nahrazování vodopádového postupu tvorby IS postupy iterativními (přírůstkovým): Podporuje přírůstkový vývoj IS a rychlý vývoj jednotlivých aplikací. Cyklicky se zde neopakují jednotlivé fáze vývoje (analýza, návrh, implementace atd.)
- Pronikání objektových charakteristik: Vnoření objektových metod, technik a nástrojů do základního postupu. Přejít od strukturovaného přístupu k analýze a návrhu IS k objektovému přístupu. Objektový přístup propojuje datový a funkční přístup tím, že definuje nad množinu dat množin operací, které s těmi daty souvisejí. Objektový přístup přináší přehlednost systému, jak pro tvůrce, tak pro uživatele, ale hlavně zpřístupní čím dál složitější aplikace i naprostým počítačovým laikům.
- Globalizace pojetí analýzy: Analýza dostává obecnější podobu, nespojuje se jen se smyslem vývoje IS, ale díky Business Process Reengineering se objevuje analýza

procesů, která tvoří základ jak ke specifikaci a následnému vývoji IS, tak i k dalším činnostem organizace, ať už personální, řídicí, organizační, technologické atd.

- Posun od tzv. hard k soft metodám: IS se navrhuje se zohledněním na lidskou složku a na efektivitu užití IS.
- Zohlednění postupu implementace typového aplikačního software: Software instalovaný v podnicích většinou vzniká implementací typového aplikačního programového vybavení (TASW), které je upravuje za pomoci parametrů a úprav specifickým podmínkám konkrétního podniku. [ŘEP1]

3. Základní postup Vývoje IS/STAG

3.1 Principy metodiky

- Orientace na cíle a problémy
- Nutná podpora ze strany vedení.
- Modelování a abstrakce, princip tří architektur
- Zapojení uživatele do návrhu.
- Klíčové dokumenty a jejich schvalování
- Ověřování a testování návrhu během celého vývoje
- V každé etapě probíhá analýza i návrh
- Vývoj probíhá z hlediska všech úhlů pohledu na systém
 - Data
 - Funkce
 - Organizace
 - Sociální a psychologická hlediska
 - Technologie
 - Ekonomická hlediska
- Výběr vhodné podmnožiny činností v jednotlivých etapách
- Úprava metodiky pro různé typy projektů
- otevřenost metodiky dalšímu rozvoji

3.2 Definice vybraných základních pojmů

- Metodika: doporučený souhrn přístupů, zásad, etap, postupů, pravidel, dokumentů, řízení, technik, nástrojů pro tvůrce IS/STAG, který pokrývá celý

životní cyklus IS/STAG. Metodika tedy určuje kdy, kdo a co má dělat během vývoje a provozu IS/STAG

- **Metoda:** určuje, co je třeba dělat v určité fázi postupu projektu. Metoda je vždy zatížena určitým přístupem, jako je funkční přístup nebo přístup datového modelování a analýzy, anebo například objektový přístup. S přihlédnutím k této charakteristice řeší každá metoda postup činností v určité uzavřené části procesu vývoje systému.
- **Technika:** určuje jak se dobrat požadovanému výsledku. Zpravidla určuje přesný postup jednotlivých činností, způsob použití nástroje, varianty rozhodnutí v určitých situacích a co z nich vyplývá, vymezuje obor své působnosti atd. na rozdíl od mety je mnohem přesnější a v závěrech omezenější v okruhu použití.
- **Nástroj:** je jednak prostředkem vyjádření výsledku činnosti, prováděné určitou technikou, jednak prostředkem umožňujícím tuto techniku použít. Většina technik požaduje, aby nástroj byl grafický. Nástroje vždy formalizují vyjádření, proto je možné a žádoucí, aby byly v maximální míře automatizovaně podporovány.
- **Konceptuální úroveň:** vyjadřuje pohled na systém v pojmech uživatele, jaké funkce musí systém provádět a jaká data musí systém uchovávat nezávisle na prostředí, ve kterém bude systém realizován, vyjadřuje podstatu systému. Modely systému na konceptuální úrovni musí být objektivně pravdivé.
- **Technologická úroveň:** vyjadřuje pohled na systém z hlediska technologického prostředí, ve kterém bude systém realizován. Technologické modely musí obsahově vyplývat z konceptuálních modelů.
- **Implementační úroveň:** systém vyjádřený ve zvoleném implementačním prostředí. Modely na implementační úrovni musí obsahově vyplývat z technologických modelů.
- **Organizace:** podnik, státní instituce, společnost apod. a nebo i část takovýchto jednotek, pro kterou se IS vyvíjí.
- **Informační systém:** systém v organizaci, který poskytuje informace nutné pro plnění cílů a záměrů organizace.
- **Subsystem:** část systému, kterou lze realizovat jako celek.
- **Alternativa řešení IS/STAG:** náčrt možného řešení systému. Alternativy se navrhují až tehdy, kdy je jasné, co musí systém dělat, dále určuje, jak by mohl

byť systém realizován. Každá alternativa musí být promyšlena, navržena a odhadnuta z hlediska nákladů a musí být založena na potřebách uživatelů i na znalostech omezení systému.

- Alternativu lze rozdělit do třech hlavních skupin:
 - Z pohledu technického vybavení
 - Z pohledu programového vybavení
 - Z pohledu provozního
- Varianta: možná realizace jedné alternativy. Například lokální zpracování dat a přenos k centralizovanému zpracování na disketách nebo na magnetických páskách.[ŘEP1]

4. IS/STAG obecně

Systém IS/STAG je informační systém pro evidenci studijní agendy vysoké školy nebo univerzity. Je doplněn o moduly absolvent evidující absolventskou agendu a přijímačky evidující výsledky přijímacího řízení. V testování je modul evaluace umožňující evidovat studentské hodnocení výuky. Systém je v současnosti používán na 9 univerzitách a vysokých školách v České republice. Systém se umístil na třetím místě v mezinárodní soutěži Eunis Elite Award v roce 2001 pořádané sdružením pro informační systémy na evropských univerzitách EUNIS.[1]

Informační systém studijní agendy IS/STAG představuje aplikaci, která dnes zabezpečuje veškeré evidence týkající se studenta, studijních programů, studijních oborů, studijních plánů, předmětů, rozvrhových akcí, zápisů studentů na předměty, zápisů studentů na rozvrhové akce, zápisů studentů na zkoušky, studentů — uchazečů (přijímací řízení), studentů — absolventů (vystavování závěrečných dokumentů), matriky studentů, evaluace (sledování kvality) studia.[2]

4.1 Historie IS/STAG

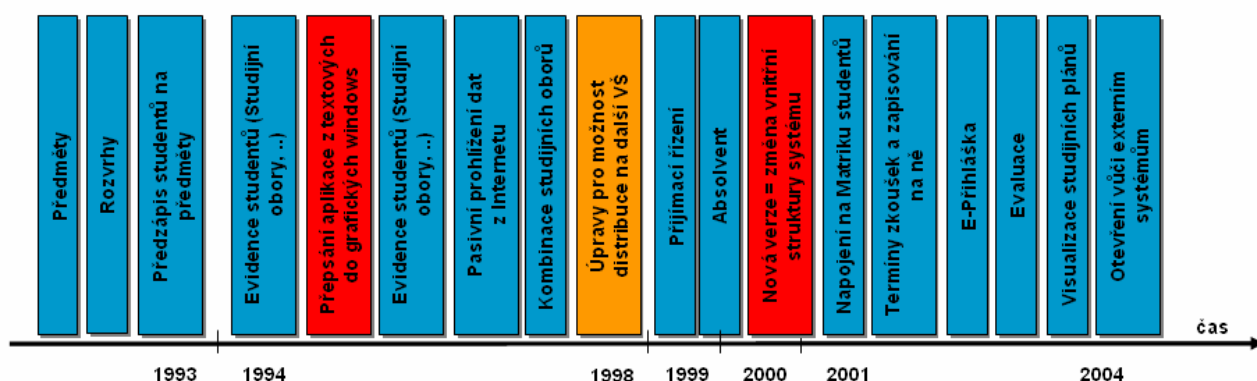
IS/ STAG se poprvé objevil v roce 1993 na Západočeské univerzitě v Plzni, kdy vznikla jeho první forma. Stalo se to kvůli přechodu z normálního systému na systém kreditní, který potřeboval počítačovou podporu systému. V raných dobách IS/ STAG si mohli studenti tvořit rozvrhy, nalézt informace o předmětech a provádět předzápis předmětů, pasivně si prohlížet data z internetu.

V roce 1998 nastala možnost distribuce IS/STAG na další vysoké školy, které o něj jevily zájem. Zájem ze strany škol byl veliký, protože do té doby pracovaly pouze s IS, které byly značně omezené, zastaralé a nesplňovaly všechny požadavky, které si školy žádaly. Mezi školy, které po roce 1998 přešly IS/STAG patří: Univerzita Pardubice, Technická univerzita Liberec, Technická univerzita Jana Evangelisty Turkyň, Západočeská univerzita, Akademie výtvarných umění Praha, Jihočeská univerzita, Veterinární a farmaceutická univerzita, Univerzita Tomáše Bati, Ostravská univerzita, Slezská univerzita, Univerzita Palackého.

V roce 2000 díky změnám ve VŠ zákoně došlo k vytvoření nové verze IS/STAG2. Zákon č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách) zavedl na vysokých školách matriku studentů. Jedním z údajů, jež musí matrika studentů obsahovat, je i rodné číslo, které se stává jedním ze základních identifikačních klíčů studenta v matrice. Z tohoto důvodu je nutné, aby i studenti-cizinci požádali o přidělení rodného čísla, zejména studenti-cizinci, na jejichž studium je vysoké škole přidělována státní dotace, tj. všichni studenti studující v českém jazyce (včetně stipendistů). Přidělování rodných čísel je upraveno vyhláškou Federálního statistického úřadu č. 55/1976 Sb., o rodném čísle, ve znění pozdějších předpisů.[6]

Od schválení nového zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů jsou po studentech, učitelích a jiných pracovnících vysoké školy vyžadovány osobní údaje. Proto se musí dbát na větší bezpečnost při manipulaci s nimi. Každý uživatel se dostane jen k záznamům, na které má oprávnění.

V roce 2005 přichází verze pro outsourcing.



Obrázek 6 Historie IS/STAG[3]

4.2 HW a SW IS/STAG

Systém byl vyvinut i realizován v prostředí Oracle. Přístup do databáze je možný pomocí vzdáleně (lokálně) nainstalovaných klientů. Pro práci klienta (v současné době 32 - bitového) jsou používána běžná PC provozovaná pod OS Windows NT v síťovém provozu. Je

vyřešena automatická aktualizace nových verzí lokálních instalací u všech uživatelů. V daném prostředí bylo dosaženo špičky až 550 současně pracujících uživatelů (v plně interaktivním režimu práce, s odezvami do 1 sekundy).[4]

HW a SW potřeby IS/STAG se dají rozdělit podle okruhů využití, zdali se jedná o vývojovou stanici, správce serverů nebo se jedná o cizí školu užívající IS/STAG nebo o namátkového i pravidelného uživatele, který může pocházet nejčastěji z řad studentů nebo zájemců o studium.

4.2.1 Vývojové oddělení

Nachází se na Západočeské univerzitě v Plzni. Sídlí přímo v univerzitě v Centru informatizace a výpočetní techniky pod vedením prorektora pro Informační Technologie, v oddělení Projekce informačního systému, které má na starost Ing. Petr Jiroušek.

Toto oddělení ve vztahu k IS/STAG lze rozdělit ještě na vývojovou stanici a vývojový server. Vývojová stanice pracuje s programy od firmy Quest Software a Oracle: ToadSoft, Oracle Developer, Oracle Designer, Oracle Klient 6i.

ToadSoft je využíván k usnadnění práce při vytváření databází, při vývoji aplikací a zjednodušuje správu úkolů. V ToadSoft vývojáři a správci databází používají nástroje pro správu databází Oracle. ToadSoft funguje rychle, jednoduše se snahou zjednodušovat, zautomatizovat a zkvalitňovat vývoj aplikací a databází. V grafickém prostředí se pracuje zcela intuitivně. Program obsahuje všechny důležité nástroje pro Oracle.[8]

Oracle Developer se používá ve vývoji pro tvorbu formulářů a zpráv. Formuláře zajišťuje Oracle Forms a zprávy Oracle Reports. Standardní formuláře slouží k prezentaci, zobrazení a upravování dat v aplikacích. V grafickém uživatelském rozhraní se pracuje s různými prvky, jako jsou tlačítka, menu, rolovací lišty. Všechny tyto komponenty mohou být použity ve formulářích. Formuláře se tvoří formou tvoření nebo formou dotazování. Je zde minimalizována potřeba zdlouhavých operací, jako je třeba tvorba dynamických SQL.[9] Oracle Reports umožňuje provozovateli poskytnout okamžitý přístup k informacím na všech úrovních, ať už jsou uvnitř nebo vně organizace. Je to nástroj používaný pro vývoj, zpráv a údajů v databázi. Oracle Reports se skládá z Oracle Reports Developer a Oracle Application Server Reports Service. [10]

Oracle Designer používá se poté, co je vygenerován kód z Oracle Developer. Slouží jako podpora pro řízení procesů, modelování a analýzu systémů, designu a softwaru. Nabízí dále úložiště dat a je úzce svázán s formulářema z Oracle Developer. Pomocí tohoto nástroje se vyvíjí aplikace.[11]

Jako operační systém se ve vývojové stanici instaluje Windows NT a minimální hardwarová konfigurace se rovná PC PIII nebo vyšší s RAM 256MB.

Vývojový server naopak používá operační systém LINUX a stejnou hardwarovou konfiguraci. Dále pracuje se vzory formulářů, reportů, helpy a aktualizacími skripty, vyvinutými ve vývojové stanici a navíc pracuje s TeX, což je typografický software, který se užívá hlavně k psaní vědeckých, zvláště pak matematických textů, kvůli tomu, že má velice kvalitní sazbu, velikou výhodou je, že tento software lze dále přizpůsobovat svojí potřebě a programovat.

4.2.2 Oddělení serverů

Další oddělení pracující na vývoji a správě IS/STAG na Západočeské univerzitě v Plzni pod Centrem informatizace a výpočetní techniky patří Laboratoře počítačových systémů vedené Ing. Lubošem Kejzlarem. Toto oddělení spravuje a pokračuje ve vývoji serverů použitých při užívání IS/STAG. Terminal Services server, Databázový server a síť WEBnet .

Terminal Services server spravuje procesy spojené s provozem IS/STAG. Je napojen na operační systém Windows NT Terminal Server. HW musí splňovat minimální podmínky pro spuštění OS, kterými jsou Procesor Pentium nebo vyšší, RAM 32MB a grafika VGA nebo lepší. Terminal Service server pracuje s formuláři, reporty, TeX, za které odpovídá vývojové oddělení.

Databázový server pracuje na OS Linux s HW konfigurací procesor P4² 1,7 GHz, RAM 2 GB. Zpracovává data pomocí Oracle Db 9i, která do něj vloží uživatelé. Oracle Db 9i poskytuje výkonnou, spolehlivou, bezpečnou správu dat vysokých objemů. Patří mezi OLTP (on-line transaction processing), tedy je to třída programů, které usnadňují a řídí transakční procesy náročných aplikací a datových skladů.[12] K tomuto serveru se připojují uživatelé pomocí mod_perl - modulu pro webový server Apache. Uživatelům stačí mít připojení k internetu, libovolný OS a libovolný HW. Mohou tak pracovat s moduly a funkcemi, které jsou jim vymezené. Do databázového serveru jsou vloženy procedury k fungování systému.

Síť WEBnet pracuje čistě v internetovém rozhraní. Prezентuje formuláře, reporty a helpy. V tomto rozhraní se provádí také zálohování v této frekvenci 7*1 den + 4*1 týden + 3*1 měsíc. Díky tomu se zamezí případné ztrátě dat.

4.2.3 Klientské stanice, tenký klient, partnerské VŠ

Klientská stanice má nainstalovanou aplikaci IS/STAG spolu s Oracle Klient 6i, jako operační systém používá Orion NT a v HW konfiguraci potřebuje alespoň procesor PII a vyšší a RAM 64MB.

Tenký klient ten může mít libovolný OS a libovolnou HW konfiguraci. Záleží jen na uživateli. Tenký klient je vlastně koncové zařízení, které buď ve formě softwaru, nebo specializovaného jednoúčelového hardwaru slouží k zobrazování aplikace, která je instalována, zpřístupněná, spravovaná a vykonávaná výhradně na centrálním serveru, takže kromě vstupu z myši, klávesnice a obrazovky klienta se žádná další data nepřenášejí. Každý z připojených uživatelů má vlastní uživatelské prostředí. Server přijímá a zpracovává všechny stisky kláves a kliknutí myši od připojeného klienta a směruje výstup obrazovky k příslušnému klientovi. Vše včetně dat a hesel je kryptováno. Klientská zařízení, tak mají okamžitý přístup k firemním aplikacím přes server, bez nutnosti instalace těchto aplikací a údržby na každém jednotlivém koncovém zařízení a to kdykoliv, kdekoliv, odkudkoliv a z jakéhokoliv koncového zařízení atd.[13]

Partnerské vysoké školy pracují na stejné bázi jako databázový server na Západočeské univerzitě. Pracují s databází nástrojem Oracle Db 8i, kterou naplňují uživatelé svými daty přes mod_perl - modul pro webový server Apache a spolu s vloženými procedurami provozují systém. Vývojové centrum v Plzni je zodpovědné za databázi IS/STAG2 a vložené procedury, které zprostředkuje škole. O zbytek se stará škola sama.

4.2.4 Case

Vývoj IS/ STAG jde stále dopředu. Využívá se CASE systém Oracle Designer 6i, který obsahuje tři základní části, grafický procesor, databázi a funkční obsluhu a pracuje jako podpůrný programový prostředek. Oracle Developer 6i (Forms builder, Report builder), tedy moduly na tvorbu formulářů a obrazovek, které fungují jako dynamické manuálové stránky a systém pro evidenci jednotlivých verzí vyvíjeného produktu CVS (Concurrent Versions System)

4.2.5 Upgrade systému

U databáze se používá systém aktualizčních skriptů, buď ve formě automatického spuštění, nebo si je administrátor spouští sám.

Aplikace na straně nativního klienta si administrátoři stahují sami a zároveň je uživatelům i sami distribuují.

Aplikace na straně webového klienta používají jednoduše aktualizční skripty.

4.2.6 Zabezpečení

U databáze Oracle se pracuje se zabezpečením na úrovni databáze nikoliv aplikace, k záznamům je povolený přístup přes VPD (Virtual Private Database), k aplikacím se dostaneme přes uživatelská konta, webový přístup je řízen přes HTTPS. U všech záznamů se eviduje, kdo ho založil a kdy a také kdo ho naposledy změnil a kdy.

5. Etapy vývoje IS/STAG

Rozdělení vývoje IS/STAG do etap se zavedlo proto, aby se proces vývoje lépe řídil a lépe se kontroloval, aby se postupovalo ve vývoji systematicky a pokud možno uceleně. Proces vývoje je totiž spojen s řadou krizových situací a problémů. Které se dají, díky rozdělení, jednodušeji zvládnout.

Etapy vývoje IS/STAG jsou:

1. Informační strategie organizace
2. Úvodní studie systému
3. Globální analýza a návrh
4. Detailní analýza a návrh
5. Implementace
6. Zavedení
7. Provoz, údržba a rozvoj

5.1 Informační strategie organizace

V roce 1993 se rozhodla Západočeská univerzita v Plzni přejít z normálního systému studia a známkování na systém kreditní. Tento systém je daleko složitější na administrativu a čistě v papírové formě, tedy zakládáním dokumentů do šanonů by nastal značný zmatek, nepřehlednost a častá chybovost, možnost duplicit získaných dat, případné ztráty dokumentů, proto se začalo uvažovat o elektronickém systému, který by předešel těmto problémům. Objevilo se několik řešení, koupit již hotový systém, který je vyzkoušený v praxi na nějaké jiné vysoké škole nebo upravit nějaký další systém pro konkrétní potřeby univerzity v Plzni nebo nechat si vytvořit systém na míru a poslední možnost, vytvořit si systém sám, pomocí vlastních sil. Aby se dosáhlo maximálního úspěchu, byla ustanovena komise odborníků, kteří probrali klady a zápory všech možností a alternativ. Jelikož se na českém trhu nepohyboval v té době žádný takový systém, který by splňoval školou podmíněné požadavky, tak tato

možnost vypadla. Stejně tak možnost nechat si systém vyvinout na zakázku. Nastal by tak totiž veliký zásah do školního rozpočtu. Proto jako nejlepší varianta se jevila, že si systém vyvine škola sama. Studium Informačních technologií má v Plzni tradici a zároveň bylo a je na vysoké úrovni. Proto tuto možnost uvítali jak vyučující, tak i studenti, kteří dostali možnost přímo se podílet na vzniku IS/STAG.

Západočeská univerzita požadovala od první verze systému, aby byl schopen plnit funkci předzázpisu. Tato funkce patří k nezákladnějším funkcím u školních informačních systémů. Přesto s ní jsou velké problémy, protože dochází k značnému přetěžování systému. Dále měl systém zvládnout tvorbu rozvrhů a studijních plánů. Samozřejmě se počítalo s tím, že nezůstane jen u těchto základních funkcí, ale že se systém bude vyvíjet dál, což se taky stalo a vývoj systému pokračuje do dneška. Momentálně se v Plzni zpracovává návrh na ICTS, který má zajistit studentům, aby kredity, které získají při studiu na zahraničních školách, jim byly uznány a zachovány při návratu na domovskou školu a tím pádem nemuseli některé předměty absolvovat znova.

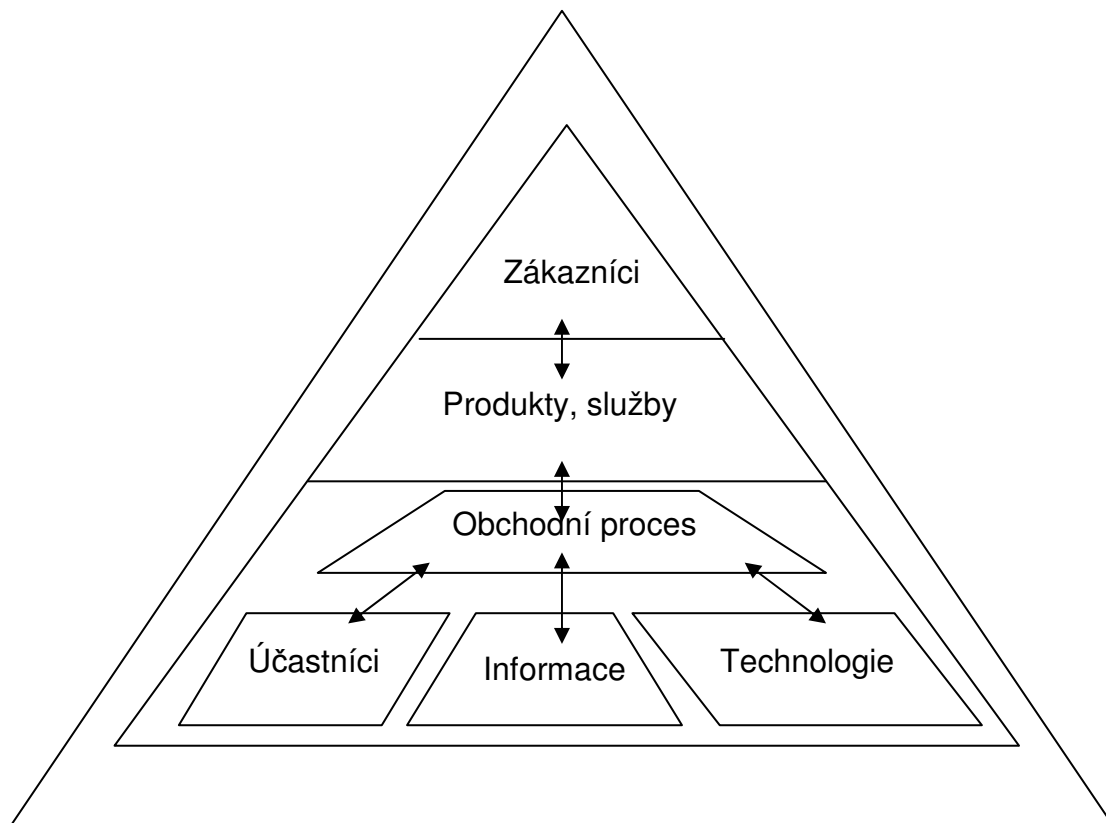
Kvůli vývoji byl sestaven tým obsahující vedoucího oddělení, programátory nativního klienta, analytiku, programátory TeXu, databázové programátory a nyní nové oddělení portálové programátory. Postupem času se tento tým obměňoval. Nyní obsahuje asi 12 stálých členů. Vedoucím vývoje se stal Ing. Petr Jiroušek, který je zároveň vedoucím pracoviště Projektce Informačního Systému. Skupinu analytiků vede Ing. Tomáš Kotouč. O programátory nativního klienta se stará Ing. Pavel Janeček. Portálové programátory vede Ing. Josef Krupička.

5.2 Úvodní studie systému

Vývojový tým se musel domluvit, s jakou technologií budou pracovat, v jakém programu budou programovat. Na vývoj tohoto systému dostali od vedení školy jeden rok. Na vývoji totiž záleželo, jestli bude škola připravená na přechod na kreditní systém. Škola kladla důraz na bezpečnost, protože v systému se měli uchovávat data o studentech, učitelích a dalších zaměstnancích. Proto když se vývojový tým rozhodoval v jakém programu navrhovat a zabezpečovat databáze, jednoznačně zvítězil program od společnosti Oracle, který splňoval požadavky platformové nezávislosti, vysokého výkonu při velkém objemu dat, spolehlivost, robustnost a vysoký stupeň zabezpečení a ochrany dat. Nejprve se jednalo o verzi Oracle Developer 3.0, během dalšího rozšiřování se přešlo na verzi Oracle Developer 4.5 a nyní se pracuje s verzí Oracle Developer 6i. Kromě Oraclu přicházely v úvahu ještě programy od firmy INFORMIX a PROGRESS. Ty ale nesplňovali bezpečnostní podmínky a nejevily se

celkově jako vhodné řešení. Škola si žádala zabezpečení na úrovni databáze a zároveň chtěla ušetřit nějaké peníze. Vývojáři předložili vedení školy modely na konceptuální úrovni, které se týkaly funkcí, které by měl systém splňovat. Přednesli jim to jako prezentaci na pravidelné schůzi a seznámili tak vedení školy a i budoucí uživatele, mezi něž patří vybraní jedinci z řad studentů a pracovníci, kteří budou systém obsluhovat, s postupem a výsledky vývoje.

Vstupy do systému by měli tvořit požadavky uživatelů, tedy studentů, učitelů a ostatních účastníků v systému. Ať už se jedná o požadavky týkající se předzázpisu nebo tvorby samotných rozvrhů či celých studijních plánů.



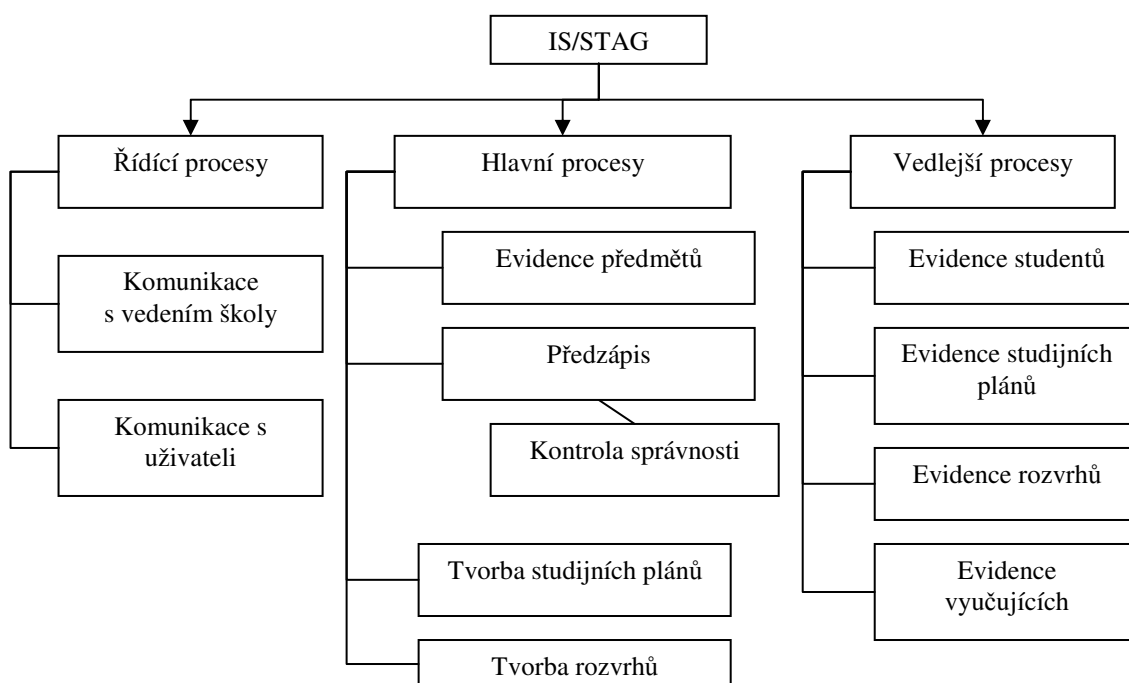
Obrázek 7 Model IS/STAG[14]

- Mezi účastníky IS/STAG patří správci systému, vývojáři, studenti, učitelé, veřejnost, ostatní zaměstnanci vysoké školy (studijní oddělení, knihovna).
- Z IS/STAG se získají informace o předmětech, o studentech, o učitelích, o studijních plánech, o předzázpise.
- Mezi navržené technologie patří OS DOS, později Windows NT, XP, Orion NT, Oracle Developer, Oracle Designer, počítače, které mají dostatečnou konfiguraci a provoz, připojení k Internetu, LAN, Server na němž poběží IS.
- Do obchodních procesů se zařadí přihlášení se do systému a pak vybrání si správné oblasti a práce v ní, například zaevidování návrhu studijního plánu, výsledků jeho

jednání, zveřejnění studijního plánu, vypisování kurzů, zápis, registrace uživatele, registrace kurzů.

- Do části produkty a služby se zařadí funkce předzápisu, tvorba rozvrhů, studijních plánů, informací o předmětech.
- Mezi zákazníky v IS/STAG se počítají studenti, učitelé, pracovníci na studijních odděleních, správci systému, vývojáři, studenti, učitelé, veřejnost, zaměstnanci vysoké školy, studenti.

Aby byla dodržena bezpečnost dat uložených v databázi, použije se právě Oracle, který zabezpečuje na úrovni databáze a nikoliv aplikace. Kvůli bezpečné komunikaci se připojuje z webu přes HTTPS (Hypertext Transfer Protocol over Secure Socket Layer).

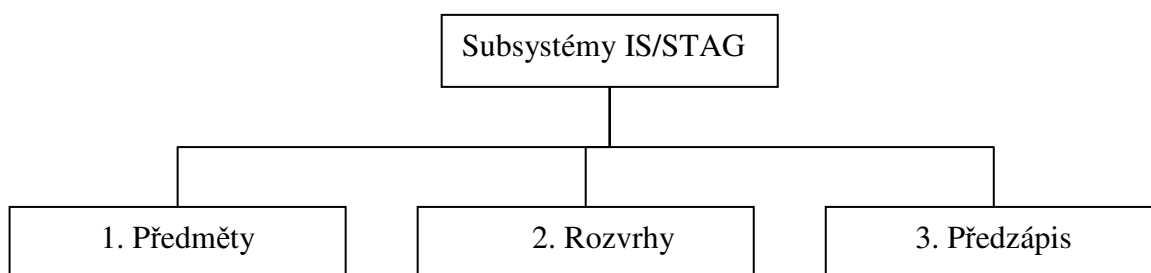


Obrázek 8 Diagram hierarchické struktury [5]

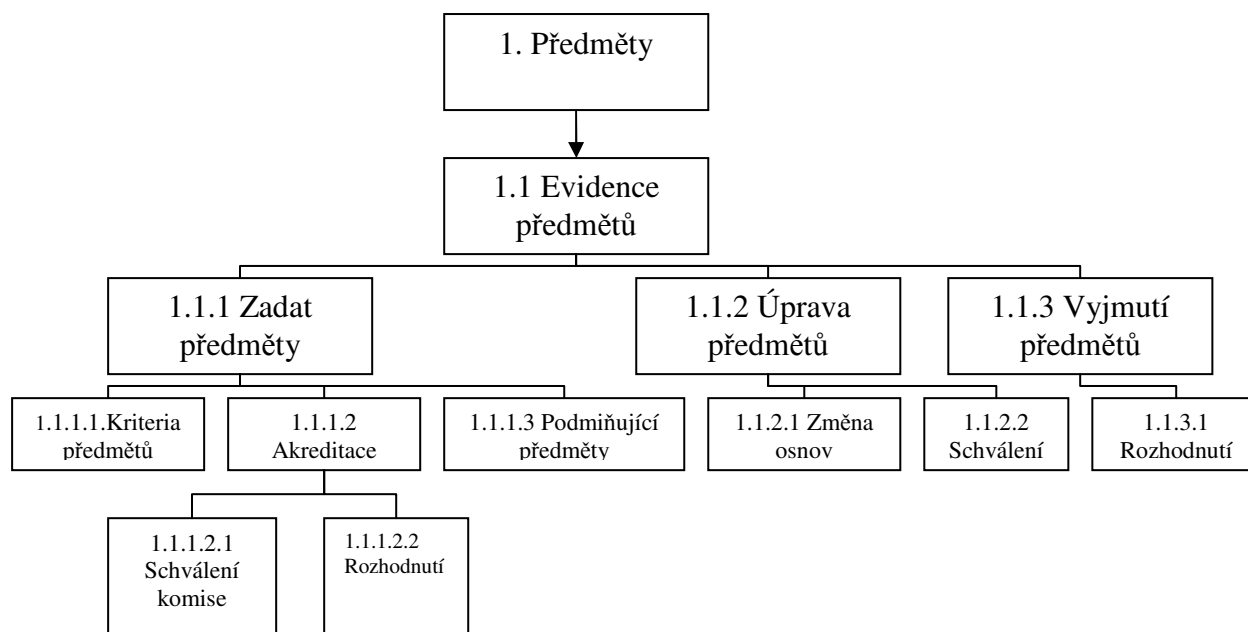
Po uskutečnění prezentace, která byla hodnocena jako úspěšná, tedy s tím, že se nevyskytly žádné námitky, se mohlo pokračovat v dalším vývoji a přejít na podrobné vypracování datové a funkční analýzy a koncept datového a funkčního modelu.

5.3 Globální analýza a návrh + Detailní analýza a návrh

Vedení školy si žádalo podrobnější informace, co se týče modelů IS/STAG. Čas byl totiž největším nepřítelem při vývoji. Na další společnou prezentaci přinesl vývojový tým funkční popis systému viz. Přílohy 1, 2 a 3. Dále také představil FSD (Function Structured Diagram) subsystémů IS/STAG.



Obrázek 9 FSD 1. úrovně [5]

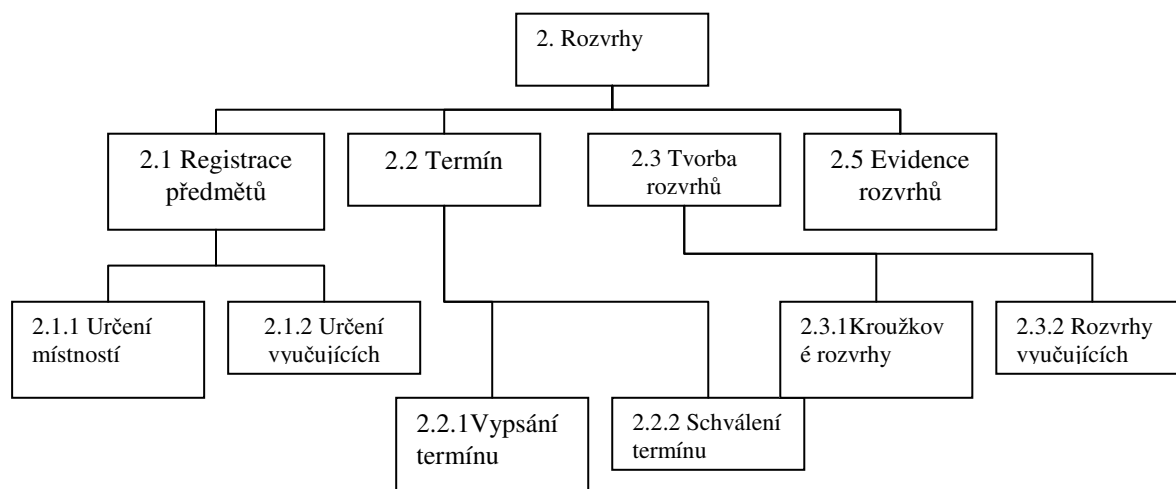


Obrázek 10 FSD 2. úrovně Předměty [5]

U subsystému předměty se jednalo hlavně o evidenci předmětů, kde jedinec, který má přístup k těmto funkcím, to se týká administrátorů, studijních referentek a učitelů, si může vybrat ze tří činností, které s předmětem provádět. Pokud se na škole začne učit nový předmět, tak před tím, než k tomu dojde, musí se zadat do evidence předmětů, kde se vyplní kritéria a podmínky, bez kterých by se nemohl vyučovat. Následně se může nechat předmět akreditovat, k tomuto úseku má přístup jen prorektor. Není to povinné, ale akreditací se zvyšuje prestiž samotné školy. Akreditaci provádí Akreditační komise vysoké školy a do systému konečné rozhodnutí zaznamená právě prorektor. Pokud se jedná o předmět, který navazuje na nějaký předchozí předmět, doplní se do systému nutné předměty, které je potřeba splnit, před tím, než se začne vyučovat tenhle předmět.

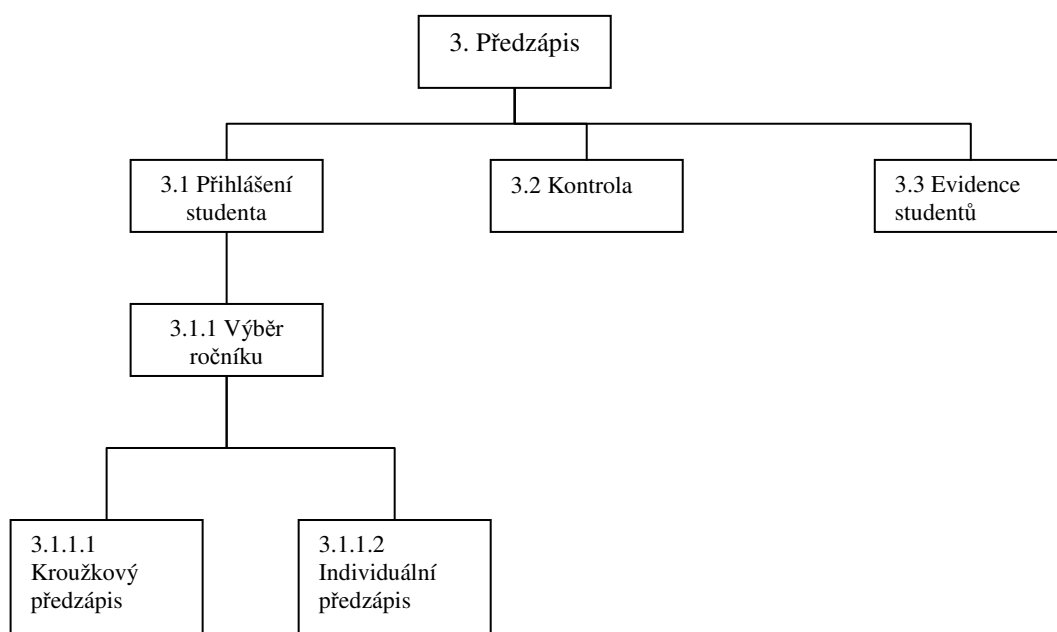
Jestliže se jedná o předmět, který již na škole funguje a vyučuje se, přesto je potřeba upravit určité drobnosti v osnovách nebo v podmínkách, přejde se do části úpravy. Předměty

se ze systému odstraňují, pokud už se nepočítá s jejich další výukou, pokud byly nahrazeny předměty modernějšími, či specifitějšími.



Obrázek 11 FSD 2. úroveň Tvorba rozvrhu [5]

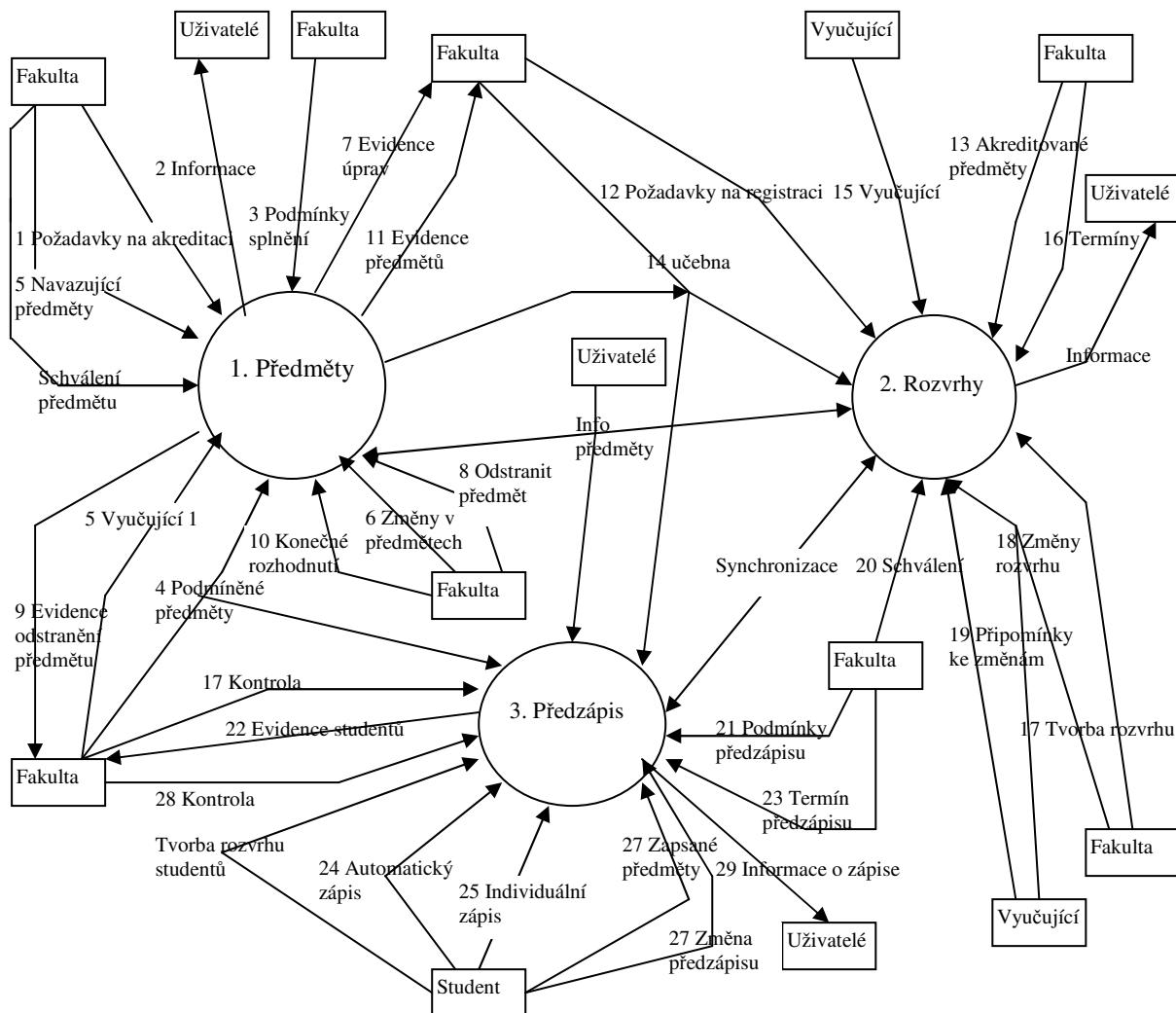
Subsystem Tvorba rozvrhu v sobě zahrnuje nejprve registraci předmětů, protože bez nich by nešel vytvořit rozvrh. Při registraci předmětů se určuje, v jaké místnosti se budou učit, kdo bude vyučujícím. Potom se vypisují varianty termínů, kdy by mělo být možno rozvrh vytvořit. Po následné diskusi se vybere nejvhodnější termín a ten se schválí. Během toho můžou rozvrháři začít sestavovat rozvrhy pro jednotlivé kroužky a samozřejmě pro vyučující. V evidenci rozvrhů si studenti a ostatní účastníci IS/STAG mohou zjistit, kdy se vyučuje a v jaké místnosti za účasti jakého vyučujícího, který předmět, plus ještě rozvrhy vyučujících a po uskutečnění předzázpisu se zde objeví i rozvrhy všech studentů.



Obrázek 12 FSD 2. úrovně Předzámis [5]

Funkce Předzámis slouží k zapisování studentů do předmětů v novém akademickém roce. Student se přihlásí do systému a vybere si ročník, do kterého se zapisuje. Dále má na výběr, jestli si chce udělat kroužkový předzámis, který mu umožní zapsat všechny A předměty, které je potřeba v daném roce splnit a to zcela automaticky. Ještě mu navíc sestaví i rozvrh. Student se musí akorát postarat o B, povinně volitelné a C, nepovinně volitelné předměty. Kroužkový předzámis může ale student ignorovat a může použít individuální předzámis, kde si musí všechny předměty zapsat sám, ať už se jedná o A, B nebo C předměty. Zde je výhoda, že si student může sestavit jemu přijatelný rozvrh. Část kontrola slouží k tomu, aby se zabránilo situacím, že by si student, ať už úmyslně nebo nedopatřením, zapomněl zapsat nějaký předmět ze skupin A nebo B. Bude na to upozorněn, že má do určitého data zjednat nápravu, pokud tak neučiní, studijní referentky tak učiní za něj. Evidence studentů zaznamenává základní data o studentech. Nejdůležitější je jejich identifikační číslo.

Dále byly předloženy DFD (Data Flow Diagram) diagramy datových toků. Ze kterých lze vyčíst, s jakými údaji se bude pracovat a jak připravit databázi, jak bude probíhat datový tok.



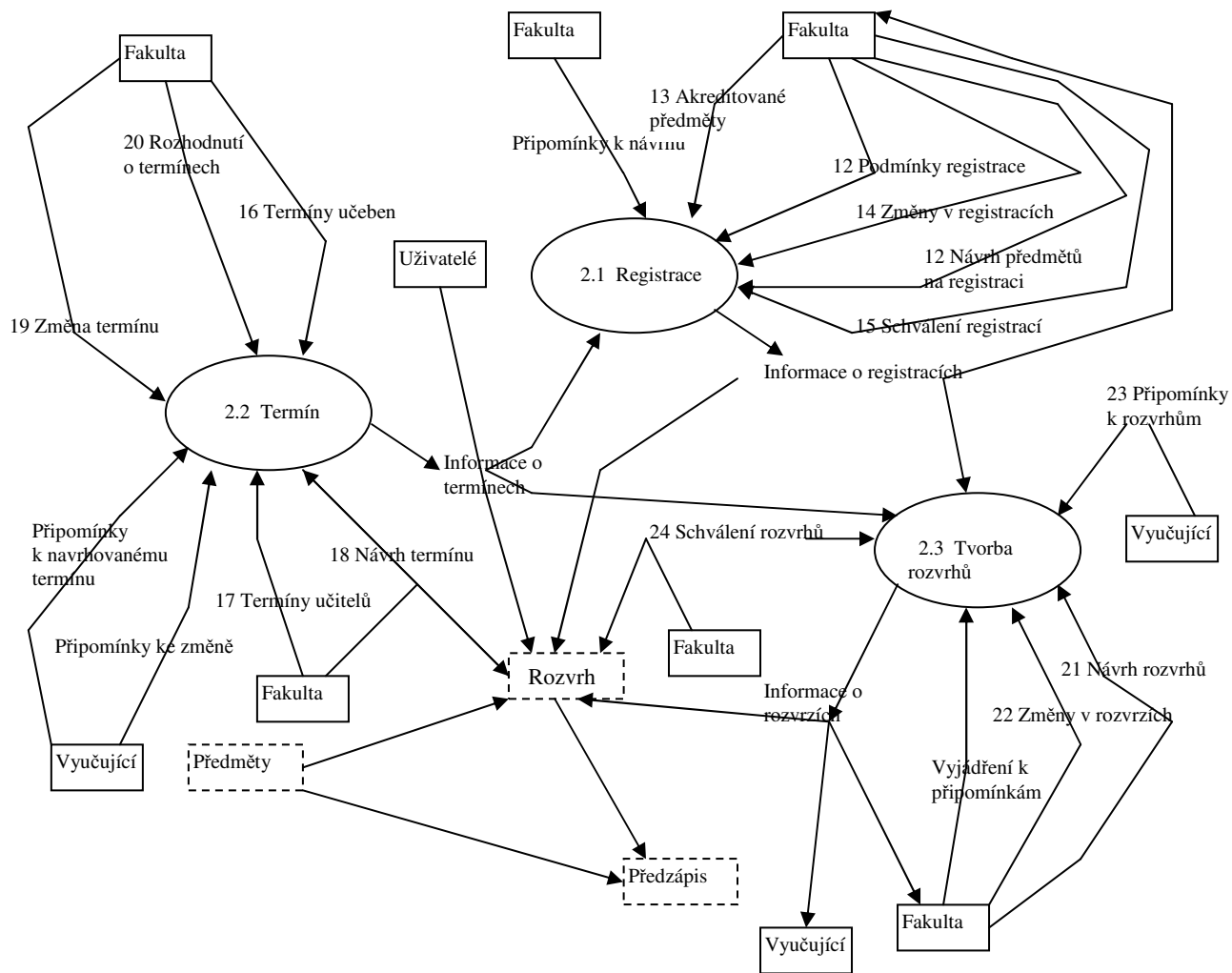
Obrázek 13 DFD subsystémy IS/STAG[5]

tabulka 1 Popis datových toků

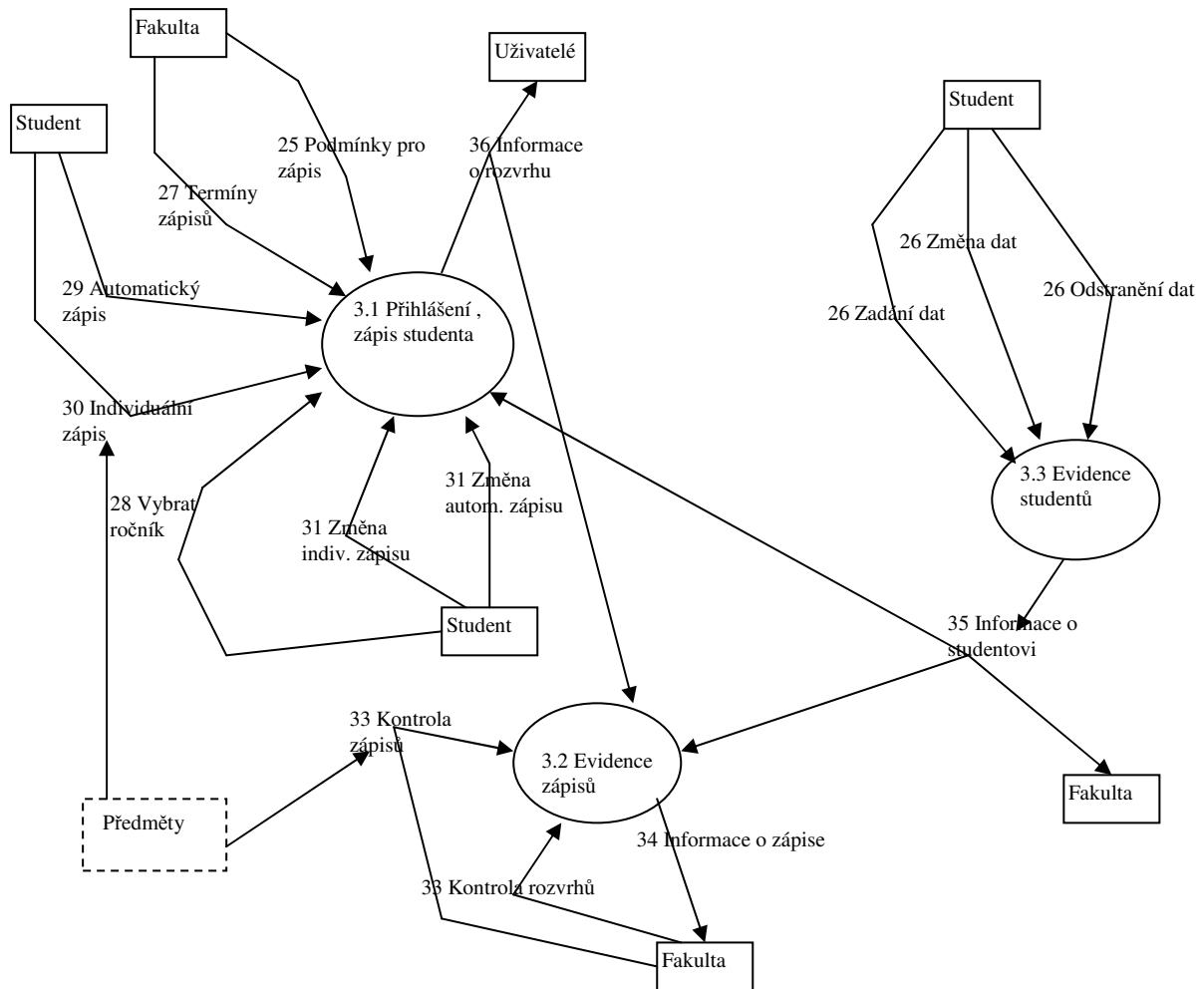
Číslo	Název datového toku	Popis datového toku	Řešení
1	Požadavky na akreditaci	Škola zažádá o akreditaci předmětu	Akreditační komise vysoké školy buď schválí nebo neschválí předmět. Požadované údaje zadá do formuláře F1: akreditace.
2	Informace 1	Studijní referentka zadá informace o předmětu, jeho cíle, přehled probírané látky, doporučenou literaturu, akreditaci, počet kreditů ..	Založení a vložení dat do formuláře F2 : Předmět
3	Podmínky splnění	Vyučující sestaví podmínky splnění předmětu	Studijní referentky vkládají podmínky do formuláře F2: Předmět
4	Podmíněné předměty	Komise určuje předměty, které musí student splnit, aby mohl navštěvovat tento	Doplňují formulář F2
5	Navazující předměty	Komise určuje, kvůli kterým předmětům, musí student tento předmět splnit	Doplňují formulář F2

Číslo	Název datového toku	Popis datového toku	Řešení
6	Změny v předmětech	Vyučující upraví obsah, podmínky předmětu	Aktualizace údajů v F2
7	Evidence úprav	Veškeré úpravy se zaznamenávají, pro další využití	Uložení do databáze
8	Odstranit předmět	Komise určí, zdali se předmět bude i nadále vyučovat	Nastavení předmětu jako neaktivní a nepoužívaný
9	Evidence odstranění předmětů	Studijní referentka sumarizuje předměty, jež jsou odstraněny	Uložení do databáze
10	Konečné rozhodnutí	Souhrn všech údajů týkajících se (ne)zavedení předmětu do výuky	Uložení dat do databáze
11	Evidence předmětů	Sumarizace všech předmětů	Aktualizace počtu předmětů a uložení do databáze
12	Požadavky a podmínky na registraci	Sestavení zápisových listů nutných k registraci předmětů	Vyplnění formuláře F3: Registrace
13	Akreditované předměty	Kontrola akreditovaných předmětů, jejich přednostní zařazení	Doplnění informací do F4
14	Změny v registracích	Fakulta upraví stanovy týkající se registrace	Aktualizace F3
15	Schválení registrací	Fakulta schválí předměty, které lze zahrnout do rozvrhu	Uložení dat do databáze
16	Učebny	Rozhodnutí o tom, v které učebně a kdy se bude jaký předmět učit	Vyplnění příslušných údajů v F4
17	Vyučující	Určení, který vyučující bude učit jaký předmět	Zadání potřebných informací do F4
18	Termín	Komise rozhodne o tom, jak bude probíhat akademický rok, co je sudý/lichý týden, kdy začíná, kdy končí.	Požadované údaje se uloží do databáze a vytiskne se sestava s termíny
19	Změna termínu	Pokud nastanou nějaké nepředložené situace, může komise na fakultě změnit termíny	Požadované změny se musí zahrnout do databáze o termínech. Možnost tisku nové sestavy
20	Schválení termínu	Fakulta schválí navržené, potažmo změněné termíny pro nový akademický rok	Uložení do databáze
21	Tvorba rozvrhu	Studijní referentka sestavuje rozvrhy podle získaných informací o předmětech, učebnách, vyučujících, termínech	Doplnění informací o rozvrhu, možnost tisku sestavy rozvrhu podle předmětu, učebny, vyučujícího.
22	Změny v rozvrhu	Navrhované změny v rozvrhu, předmětu, vyučujícího, učebny	Změna údajů v F4
23	Připomínky ke změnám v rozvrhu	Zkonzultování změn v rozvrhu a jejich schválení/neschválení	Uložení změn do databáze
24	Schválení	Fakulta schválí hotové rozvrhy	Uložení kontroly do databáze
25	Podmínky předzápisu	Co musí student splňovat, aby se mohl zapsat	Záznam z propojených formulářů F2 a F4
26	Informace o studentech	Student vyplní základní informace o sobě, hlavně dostane přiřazené identifikační číslo	Informace vyplní do formuláře F6
27	Termín předzápisu	Administrátoři připraví, kdo a kdy se může zapisovat, kdy probíhá kroužkový předzápis a kdy individuální	Stud. ref. vyplní informace ve formuláři F5

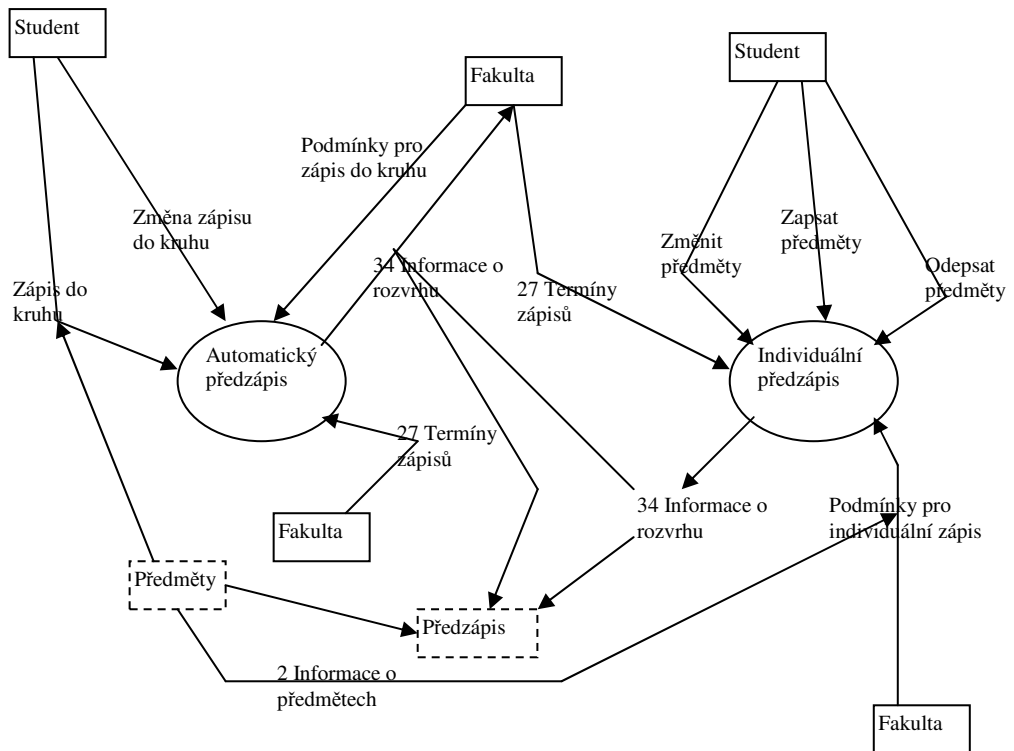
Číslo	Název datového toku	Popis datového toku	Řešení
28	Zapsání ročníku	Studenti se zapisují do pro ně příslušného ročníku	Vyplnění údajů v F5 a F6
29	Automatický zápis	Student se zapíše pomocí kroužkového zápisu. Sám si musí zapsat jen předměty B a C	Automatické vyplnění formuláře F5 :předzápas
30	Individuální zápis	Student si zapisuje všechny předměty sám	Samostatné vyplnění F5
31	Změna automatického zápisu Změna individuálního zápisu	Student si změní studijní kruh nebo předměty které si sám zapsal	Změna dat ve formuláři F5
32	Zapsané předměty	Souhrn všech informací, týkajících se zápisu, jaké jsou zapsané předměty, kolik je možné získat kreditů.	Tisk sestavy
33	Kontrola zápisů Kontrola rozvrhů	Systém překontroluje správnost zápisu, když tak upozorní na chybu, nebo ji následně sám odstraní	Aktualizace formuláře F5
34	Informace o zápise	Uživatelé mají možnost zjistit si jednotlivé informace o zápise jednotlivých studentů, vyučujících, učeben.	Možnost tisku jednotlivých informací



Obrázek 15 DFD 2. Rozvrh[5]



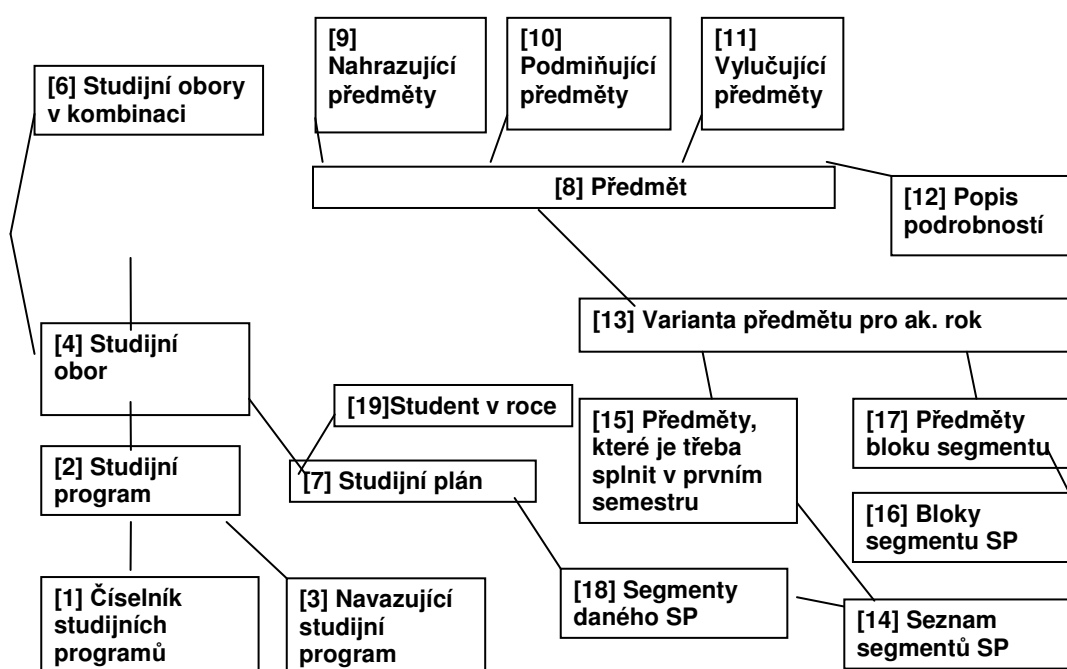
Obrázek 16 DFD Předzapis[5]



Obrázek 17 DFD Předzapis2[5]

Vývojové týmy se snažili vycházet z komplexního relačního datového modelu, který se nazývá ERA model a je nastíněn v příloze 4. Vývojáři navrhli tento model tak, aby byl schopen udržet při sobě velké množství nesterorodých dat. Tento model se bude v budoucnosti ještě mnohokrát měnit v závislosti na rozrůstání se IS/STAG. Data, která drží model pohromadě, by se měla ukládat do databázových tabulek na serveru. Počty těchto tabulek porostou do desítek.

Jedna z podmínek, kterou si vedení školy stanovilo, byla možnost tvorby studijních plánů. Tím by získali všichni studenti a pracovníci školy přístup ke studijním plánům, ke kontrole jejich návaznosti a přibyla by tak další forma sdělení informací uživatelům. Funkce studijní plány, stejně jako ostatní funkce, se mají opírat o tabulky z ERA modelu. Každé funkci jsou tak přiřazeny konkrétní tabulky a konkrétní data, samozřejmě s tou možností, že některé tabulky a data budou mít funkce společné. Díky tomu se nemusí pro každou funkci vytvářet vlastní databáze a předejde se případným duplicitám. Podle jakého klíče by se mělo postupovat při tvorbě studijního plánu lze vidět na obrázku 10.



Obrázek 18 Schéma studijních plánů [MICH1]

V hranatých závorkách v popise níže jsou uvedeny názvy skutečných tabulek používaných v IS/STAG. Tyto tabulky obsahují ohromné množství atributů. Pravidla a vazby mezi nimi jsou popsány v následujícím textu.

1. Nastavení systému začíná v nastavení číselníku studijních programů. Klíčovou hodnotou je zde kód studijního programu. [CIS_STUD_PGM]
2. Konkrétní studijní program je založen pro určitou fakultu s určitým typem studia (bakalářské, magisterské, navazující, doktorské) a s určitou formou studia (prezenční, distanční, kombinovaná). Dalšími atributy jsou např. název programu, titul, který získá absolvent a maximální délka studia tohoto programu. [STUDIJNI_PROGRAMY]
3. U studijního programu je možné nastavit, jaké studijní programy na něj navazují. [NAV_STUD_PROGRAMY]
4. Ke každému studijnímu programu je možné založit několik studijních oborů, u těch se zadává také minimální počet kreditů pro splnění, počet etap studia a anotace. [OBORY]
5. U studijního oboru je možné uvádět, jaký studijní obor ho podmiňuje (absolvování podmiňujícího oboru je podmínkou zapsání daného studijního oboru). [OBORY] – unární vazba
6. V systému je možné zavádět kombinace studijních oborů a přiřazovat jim příslušné studijní obory. [POM_KOMB, KOMBINACE]
7. Ke studijnímu oboru se zakládají pro každý rok studijní plány. Zpravidla se liší etapou studia, pro každou etapu jeden plán. Ale může existovat i více plánů pro tutéž etapu studia – ty se pak vzájemně liší verzí. [STUDIJNI_PLANY]
8. Dále jsou v systému založeny předměty. Předmět je charakterizován zkratkou předmětu a zkratkou katedry, která ho vyučuje. [PRED_HLAVICKY]
9. K předmětu se zadávají předměty, které tento předmět nahrazuje. [NP_NAHR]
10. K předmětu se zadávají předměty, jejichž absolvování podmiňuje absolvování tohoto předmětu – předměty podmiňující. [NP_POD]
11. K předmětu se zadávají předměty, jejichž absolvování znamená, že si student nesmí zapsat tento předmět – předměty vylučující. [NP_VYLUC]
12. U předmětu je možné (pro každý akademický rok zvlášť) uvést jeho podrobný obsah, rozpis jednotlivých přednášek, cvičení pod. [PRED_DETAILY, PRED_OKRUHY]
13. K předmětu se každý rok, kdy se předmět vyučuje, zakládá jeho varianta. Klíč identifikující předmět je rozšířen o rok. Tabulka variant udržuje řadu podstatných informací z hlediska zápisu předmětu – kreditní ohodnocení předmětu, počet a typ jednotek výuky v semestru, zda je předmět ukončen

zápočtem či zkouškou, zda je možné provést opakovaný zápis apod.

[PRED_VARIANTY]

14. V systému se vytvářejí segmenty studijního plánu. Segmenty tvoří vazbu mezi bloky předmětů a studijními plány. [PORADI_SEGMENTU]

15. Ke každému segmentu studijního plánu je možné uvést několik předmětů, které je nutné absolvovat během prvního semestru studia (forma prodlouženého přijímacího řízení) – v případě nesplnění je student vyloučen.

[SEM1]

16. Ke každému segmentu studijního plánu je možné uvést několik bloků předmětů. [SEGMENTY_ST_PLANU, BLOKY]

17. Ke každému bloku předmětů segmentu je možné přiřadit několik předmětů. Tato tabulka nese podstatné informace – doporučený ročník a semestr předmětu. Kreditní systém umožňuje zápis i mimo doporučené termíny, pokud jsou splněny podmínky pro zápis. [BLOKY_PREDM]

18. Segment studijního plánu je možné přiřadit více studijním plánům (je možné segmenty sdílet). [SEGMENTY_ST_PLANU]

19. V každém roce se přiřazují studenti k danému studijnímu plánu. Tabulka mimo jiné obsahuje také aktuální studijní průměr studenta a vazby do tabulek s osobními údaji studenta a s jeho známkami. [STUDENTI_NA_OBORU_V_ROCE]

[MICH1]

5.4 Implementace

Odpovědní členové vedení školy odsouhlasili a schválili tyto návrhy. A tím dali prostor programátorům a vývojářům k tvorbě samotných formulářů, které jsou důležité pro správu a vkládání dat od uživatelů. Škole se podařilo získat dotace na vývoj od Ministerstva školství a tím pádem disponovala dostatkem financí na zaplacení licence Profesionální verze produktu Oracle Developer a Designer, který využívá nástroje Oracle Forms a Report k tvorbě formulářů a zpráv. Tato licence je 6 krát dražší než licence verze Classic.

Vývojáři se snažili navrhnout formuláře tak, aby splňovali podmínky, které si nastínilo vedení školy.

Tedy Formulář Předmět umožňuje zadat údaje týkající se látky, která se bude v daném předmětu probírat, doporučenou literaturu, jednotlivé požadavky na splnění, jako jsou samostatné práce, zkoušky, zápočty, klasifikované zápočty, dále také, jestli je tento předmět podmíněn splněním jiného předmětu nebo ne. Údaje do tohoto formuláře vkládají sekretářky daného ústavu, kterým se předloží připravená data.

UIS - [Program předmětu]

Okno: Akce Editace Blok Záznam Pole Dotaz Okno Nápověda

Katedra: **KMA** Předmět: **MA1-A** Rok var.: **2006** Název: **Matematická analýza 1**

Varianty předmětů | Přehled látky | Požadavky | Doporučená literatura

Česky
získání zápočtu je třeba získat minimálně 66,66666% (40 bodů). Písemnou práci lze opravit v jednom náhradním termínu.
Zkouška - písemná a ústní část

Anglicky

Německy

Francouzsky

Požadavky pro splnění předmětu
 Záznam: 1/1 <PřOŠ <Ladě

Obrázek 19 Formulář F2 Předmět (požadavky)[15]

UIS - [Program předmětu]

Okno: Akce Editace Blok Záznam Pole Dotaz Okno Nápověda

Katedra: **KMA** Předmět: **MA1-A** Rok var.: **2006** Název: **Matematická analýza 1**

Varianty předmětů | Přehled látky | Požadavky | Doporučená literatura

Důležitost	Autor	Název
Doporučená		trial.kma.zcu.cz
Doporučená	Brabec, Jiří ; Martan, František ;	Matematická analýza I
Doporučená	Drábek, Pavel - Mika, Stanislav	Matematická analýza I

ISBN: 80-03-00044-0 ISSN: Typ: Fyzická Internetová adresa místa uložení: http://josef.uk.zcu.cz/F?func=item-global&doc_library=ZUPI

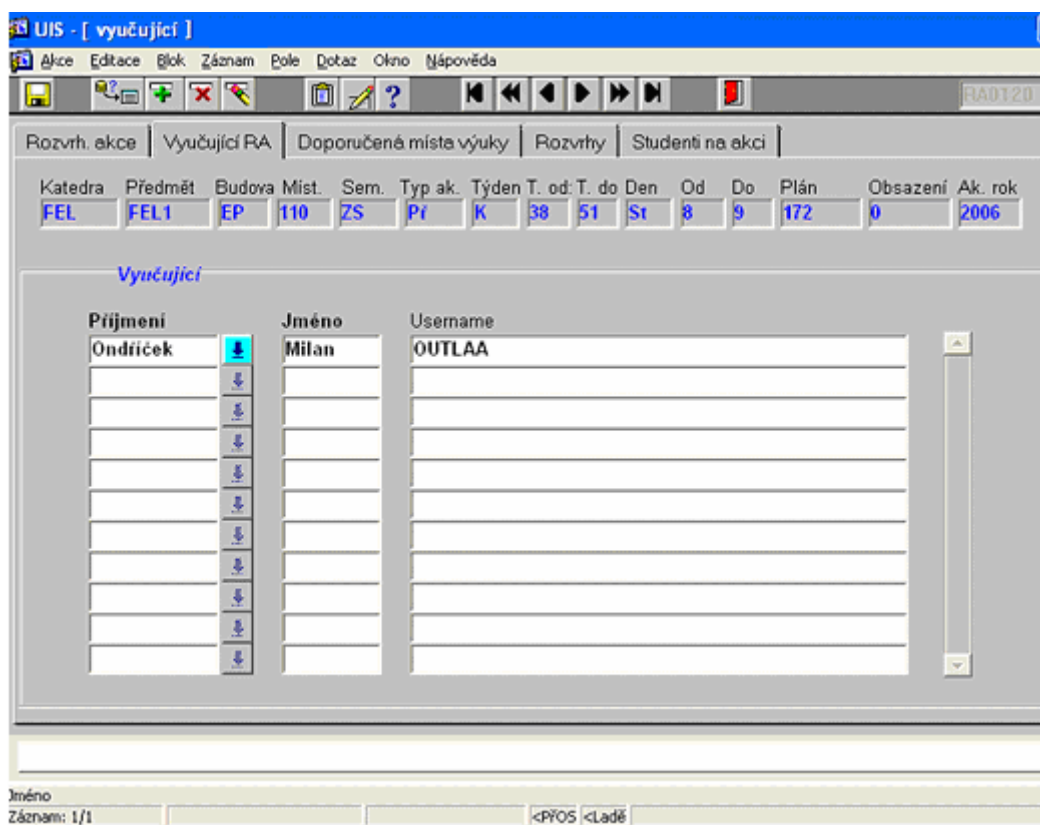
Identifikátor v knih. systému: 000017257 Rok a místo vydání: 1989 Praha Zdroj. dok.: Poznámka: Jazyk:

Vyhledat titul v knihovně dle aktuálního: ISBN autora názvu

Vázev literatury
 Záznam: 2/3 Seznam hodnot <PřOŠ <Ladě

Obrázek 20 Formulář F2 Předmět (literatura) [15]

Oproti tomu do Formuláře F4 Rozvrhy se vkládají k jednotlivým předmětům, na jednotlivé vyučovací hodiny jména učitelů a pracovníků instituce, kteří budou zajišťovat chod výuky a čísla učeben, kde bude výuka probíhat. Opět data zadávají do systému sekretářky daných ústavů. O správnost údajů a rozvrhů se starají rozvrháři.



Obrázek 21 Formulář F4 Vyučující [15]

Formulář F5 Předzázpis si spravují studenti sami. Záleží na nich, zda li se rozhodnou k sestavování si rozvrhu individuálně nebo kroužkově. Výhody a nevýhody jsou již popsány výše

IS/STAG - Předzázpis
[Návod](#) [Změna hesla](#)

Karel ŠTĚPÁN - E0400441
Ak. rok 2008/2009, Studijní program : B6209

[Informační ve veřejné správě](#)
Povinné segmenty studijního plánu

Kdy mám povolen předzázpis

Normální		Kroužkový	
Od	Do	Od	Do
04.09.2008 - 08:00	05.09.2008 - 23:59		

Celkem zapsáno kreditů

ZS ▶ rozvrh	0
LS ▶ rozvrh	0

Katedra : % Zkratka : %

Název : %

Vyberte v levém sloupečku blok předmětů. Pokud studujete více oborů, pak vyberte nejprve obor. Poté se Vám na tomto místě zobrazí seznam předmětů v daném bloku. Pokud znáte konkrétní předmět nebo jeho název, pak použijte vyhledávání předmětu v horní části obrazovky. Po kliknutí na zkratku předmětu se Vám zobrazí rozvrh. Pokud budete potřebovat návodů, klikněte na otázník v pravém horním rohu.

Název bloku	St.	Kreditů			Předmětů	
		Min.	Max.	Zaps.	Min.	Zaps.
Povinné předměty	A	153	-			0
Číslo jazyk. 2. ročník	B	0	-			0
Číslo jazyk. 1. ročník	B	4	4			0
Povinné volitelné předměty - 2. ročník	B	3	18			0
Povinné volitelné předměty - 3. ročník	B	7	8			0
Volitelné předměty	C	0	-			0

Obrázek 22 Formulář F5 Individuální předzázpis[16]

Vedení školy si žádalo, aby získalo dostatečný přehled o počtech svých studentů a také aby se nejednalo pouze o anonymní číslo, nýbrž o konkrétního jedince. Proto si přáli zřídit evidenci studentů. Data o sobě do systému student vyplní sám přes internetový browser. Ty se následně uloží do databáze. Například při změně adresy musí samozřejmě dojít k opravě údajů. Buďto je provede student sám nebo změnu nahlásí na příslušné studijní oddělení a studijní referentka pomocí vlastního přístupu k databázi záznamy upraví.

Editace osobních údajů studenta

Trvalá adresa

Jméno: Příjmení: **Rodné číslo:** Datum narození: **Ulice:** **Obec:** **Okres:** **Pošta:** **Stát:**
 Karel ŠTĚPÁN 840905 05.09.1984 Nádražní 841 Králky Ústí nad Orlicí 56169 Králky Česká republika

Pokud jsou v těchto datech nesrovnalosti, obraťte se na studijní oddělení Vaší fakulty

Adresa určená pro doručování a další kontakty

Ulice: ;
 Číslo/ulice: ;
 Okres: ;
 Obec: ;
 Část/obce: ;
 Pošta: ;
 Telefon: ;
 Mobil: ;
 E-mail: ; Uvádějte maximálně jednu adresu a neuvádějte adresu na mobilní operátory.

Bankovní účet

Vámi uvedený bankovní účet bude využíván i na případné zaslání stipendia jako např. bytovacích stipendia, prospěchového stipendia nebo mimořádného stipendia.
 Je možno zadat jen číslo bankovního účtu vedeného v české měně některou z bankovních institucí se sídlem na území ČR.

Pokud máte bankovní účet u České spořitelny, uvádějte nové číslo účtu, aťkoliv číslo se specifickým symbolem.
 Zkontrolujte, zda zadáváte svůj bankovní účet a ne např. účet školy nebo koleji.

Za správnost bankovního účtu ručíte a přebíráte na sebe odpovědnost za náklady vzniklé případným špatně uvedeným číslem bankovního účtu.

IBAN: ; Pokud máte veden účet v zahraničí, uveďte i IBAN Vaší banky.
 Číslo účtu: - ; Pokud Vaše číslo účtu neobsahuje pomlčku, vepište ho jen do druhé části, za pomlčku. Pokud je delší než 10 znaků, vepište levě přebývající znaky před pomlčku.
 Banka: ; Čtyřmístný kód banky

Obrázek 23 Formulář F6 student (student) [18]

UIS - [Evidenční karta studenta]

ES0010

Výběr studentů | Student | Studované obory | Výsledky studia | Osobní údaje | Osobní údaje 2 | Matrika | Cizí studia

Student na studijním programu

Osobní číslo: F03088 | Rodné číslo: 7304212102 | Příjmení: JANTSCHOVÝ | Jméno: Pavla | Titul: | Kód: B6107 | Fak.: FF | Typ: Bakalářský | Forma: P

C. vz.: N | Obor: 6101R008 | Spec. Et.: 00 | D. nástupu: 15.09.2003 | D. ukončení: 05.06.2007 | Stav: Nestuduje | Předch. os. č.: | Poznámka: | Číslo na kartě: |

Vykazovat VZP | Vykazován | Absolvent | | | |

v jednotlivých akademických rocích

Ak. rok	Místo	Roč. D. zápisu	D. ukončení	Kód	Kombinace	Fin.	Přeruš. ZS	LS	Poznámka	Kruh	Sk.	A	B	Pl.
2006	P	4	01.09.2006	05.06.2007	1	1	<input type="checkbox"/>	<input type="checkbox"/>		2		A	B	Pl.
2005	P	3	01.09.2005	31.08.2006	0	1	<input type="checkbox"/>	<input type="checkbox"/>		2		A	B	Pl.
2004	P	2	01.09.2004	31.08.2005	0	1	<input type="checkbox"/>	<input type="checkbox"/>		2		A	B	Pl.
2003	P	1	15.09.2003	31.08.2004	0	1	<input type="checkbox"/>	<input type="checkbox"/>		2		A	B	Pl.

Přerušení v akademickém roce

Přerušeno od: | Přerušeno do: | Nástup po přer.: |

Doktorandi - rozšiřující informace

Katedra: | Školitel: | Název individuálního stud. plánu: |

D. předpokl. ukončení studia: |

Osobní číslo: | Záznam: 1/1 | Seznam hodnot | <PřOS <Ladě

Obrázek 24 Formulář F6 student (referentka) [19]

6. Slabá místa vývoje IS/ STAG a jejich protiopatření

Je vidět, že systém studijní agentury je složitý a komplikovaný. Tato složitost se umocňuje ještě tím, čím víc uživatelů se v systému pohybuje. Potom stačí malá nepozornost při sestavování pravidel, při tvorbě systémového harmonogramu a může dojít k problémům, které vedou k velkým nepříjemnostem.

Jako nejslabší článek, který se objevil při vývoji IS/ STAG, by se mohl, jako už mnohokrát předtím označit lidský faktor. Do vývoje se totiž zapojilo velké množství lidí. Hlavní tíhu vývoje na sobě nesli programátoři z řad učitelů. To samo o sobě už je předurčilo k tomu, že vývoji nemohli věnovat 100% svého času. Museli ho rozdělit mezi povinnosti ve škole a samotný vývoj. Dále škola chtěla ušetřit potřebné peníze a do projektu zapojovala i studenty, což je jistě správné a pro studenty velice přínosné. Studenti ve formě semestrálních, bakalářských a diplomových prací se snažili objasnit konkrétní problémy vzniklé vývojem. Ale přece jenom znalosti studentů nejsou na tak odborné úrovni, aby se vyvarovali všech nedostatků. Proto jejich práce po nich museli kontrolovat znovu hlavní vývojáři a předělávat chyby, jichž se studenti dopustili. Což samo o sobě zpomalovalo postup. Některým studentům byla navržena další spolupráce s ZČU, mohli tedy pracovat i nadále na vývoji IS/ STAG. Bohužel pro IS/ STAG tito jedinci z řad programátorů, informatiků a dalších odvětví, ať už to jsou studenti nebo učitelé, jsou v dnešní době velice žádaným zbožím na pracovním trhu. A vysoká škola nemůže konkurovat nabídkám a hlavně platebním podmínkám, kterých se

dostane těmto jedincům v soukromém sektoru. To mělo a má za důsledek, že většina odchází za lepším a do vývoje se musí zapojit noví členové, to sebou přináší nové zaučování, seznamování se systémem, pronikání do problematiky, ztrátu času, který se mohl věnovat jiným věcem.

Jako dostatečné protiopatření by se jevilo, kdyby škola lépe ohodnotila práci vývojového týmu a to hlavně po finanční stránce, aby pominula vidina peněz získaných v jiných institucích. Bohužel, co se týče finanční stránky tak si škola nemůže dovolit razantnější výdaje, než má nyní. Musí tedy zapracovat na motivaci. Snažit se studenty, absolventy a učitele přesvědčit o výhodách setrvání na akademické půdě. A pokračování v práci pro školu. Nabídnout jim jiné výhody, jako jsou třeba přístup ke kvalitní technice, práce s perspektivními lidmi, vývoj nových věcí, možnost seznámení se s jinými systémy, zahraniční stáže a další.

Další problém, který se vyskytl, byla komunikace mezi členy vývoje a členy vedení školy, kteří schvalovali konkrétní návrhy a předkládali svoje vize o systému. Chyba byla v nedostatečné znalosti vývoje IS, kvůli tomu se rozcházel v některých případech představy vývoje a školy. Tomu šlo zabránit dostatečnou studií této problematiky.

Podle slov pana Ing. Kotouče se nestalo během těch 15 let, co je systém v provozu a nadále se zdokonaluje, že by se vyvinul modul, část, úsek systému, který by následně nepřešel do provozu a musel by se vyhodit. Veškeré problémy, které by se mohly objevit v běžném provozu, se zatím vždy podařilo nalézt a objasnit v průběhu analýzy systému. Jinak řečeno nevyužívalo se ani se nevyužívá agresivní programování. Čeká se na podnět od zákazníka.

O IS/ STAG projeví zájem i zahraniční univerzity z Německa, Anglie, Slovenska a Indie. Nakonec ho ale žádná zahraniční univerzita nekoupila a neprovozuje ho. IS/ STAG momentálně není schopen naplnit požadavky zahraničních univerzit. Největším nedostatkem IS/ STAG je to, že nativní klient se naprogramoval jen pro Českou republiku. Pokud by IS/ STAG chtěli využívat zahraniční univerzity, musel by se celý znova přeprogramovat a přepsat v konkrétním jazyce daného státu, nebo který by si univerzita přála. A navíc v případě Anglie se vyskytl další nedostatek. Anglické univerzity rozdělují školní rok na trimestry, ne tedy jako v České republice na semestry. A IS/ STAG není schopen bez hlubšího zásahu do zdrojového kódu reagovat na tyto požadavky. Určitě je to škoda a během vývoje, mohly vývojové týmy pamatovat na možnost prodeje systému nejen v rámci České republiky. Přineslo by to zviditelnění jak České republice, tak samotné ZČU.

Problémy, které jsou zmíněny níže, se týkají zavádění IS/ STAG do praxe na vysokých školách. Informace jsou volně převzaté z prezentace IS_STAG.

Pokud konkrétní škola, zavádějící systém do provozu nevytvoří jasná pravidla, která se týkají studia, tím je myšleno, pokud se uvnitř organizace striktně nenastaví úlohy jednotlivých osob a pravidla studia, potom systém nemůže fungovat.

Pokud nastane situace, že v systému na sebe budou jednotlivé funkce, které jsou zpracovávány různými osobami navazovat a pokud nebude sestaven jejich přesný a podrobný harmonogram, mohou vzniknout problémy se zacyklením systému a jeho neprůchodností.

Problémy nastávají také při přechodu ze starého systému na nový v tomto případě na IS/STAG. Nový systém bývá propracovanější a závislejší na číselníky a závislostní pravidla. Starý systém k těmto problémům a pravidlům přistupuje většinou volněji. Kvůli tomu je potřeba data, která využíval starý systém probrat a pročistit, aby neobsahovaly duplicity atd. Nový systém také přísněji dbá na „čistotu“ dat.

Dalším problémem bývají samotní uživatelé. Nový systém od nich vyžaduje nové seznamování se systémem. Znovu se musí učit, jak postupovat v kterých situacích, jak řešit problémy. Samozřejmě jim systém přiřazuje i určitou odpovědnost. Právě ta bývá příčinou toho, že uživatelé nechtějí přijmout svoji roli v systému a svoji odpovědnost posílají dále. Díky tomu později systém kolabuje.

Databázový server, který pracuje se systémem, se nachází po většinu roku v relativní nečinnosti a nevytíženosti. Bohužel pak přijdou situace, jako je začátek předzápisu, konec předzápisu nebo ze strany učitelů mezní termín zadávání známek. V těchto situacích nastává značné přetížení serveru. Škola totiž uvede to provozu jen takový server, který stačí na provoz během většiny roku nebo prostě takový na který má peníze, ale v období zápisů tento server nestíhá a stává se zahlceným. Většinou škola nepočítá se tím, že je potřeba změnit nastavení parametrů databáze a že je potřeba usměrnit počty uživatelů přistupujících k systému. Tedy aby se ve stejnou dobu nekonal předzápis, třeba se zadáváním známek atd. V rámci předzápisu se vyplatí sestavit postupný přístup studentů, aby se tak nápor rozdělil.

Problémy přinášejí i informace, které je možné ze systému vygenerovat. Jejich užitečnost je přímo úměrná čistotě dat, které se nacházejí uvnitř databáze.

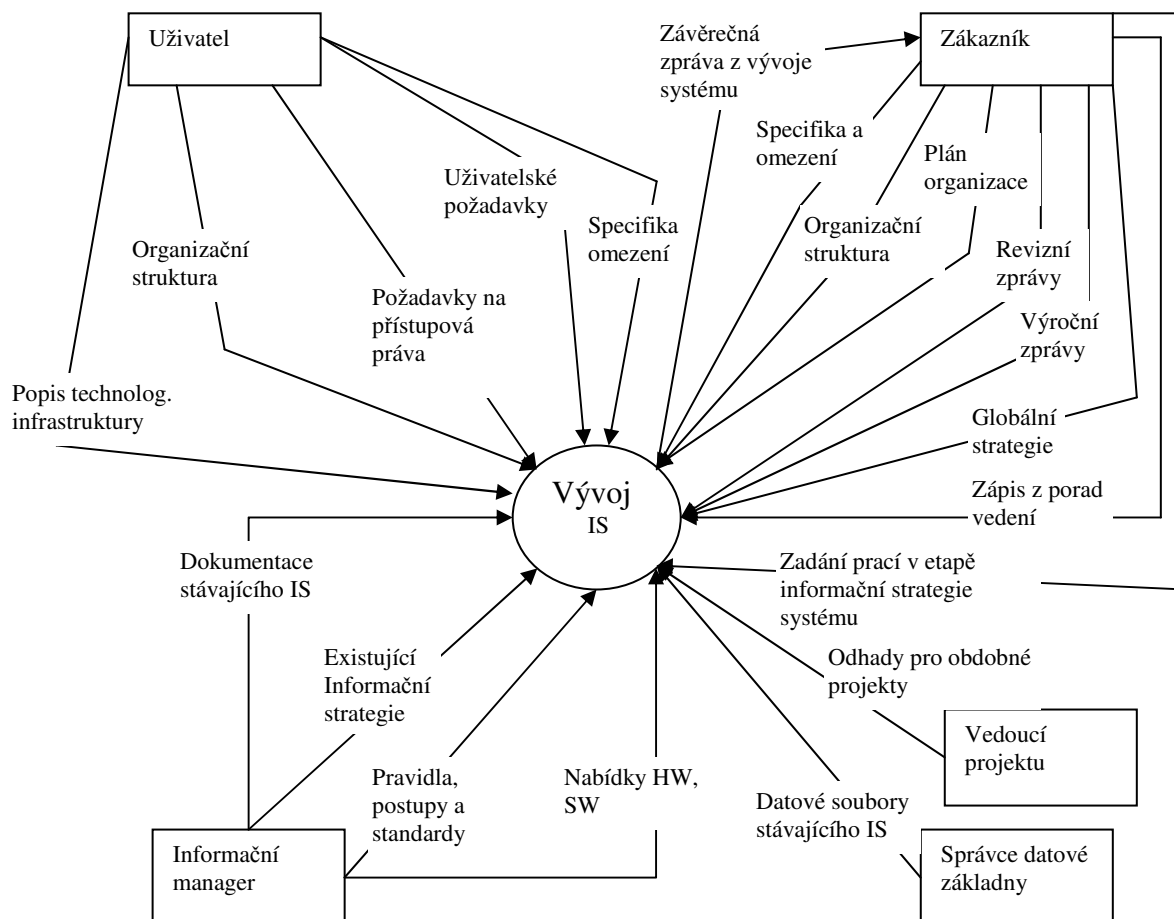
Všem těmto problémům lze předcházet dostatečnou přípravou před a během zahájení vývoje a provozu systému. Záleží také na komunikaci mezi uživateli a vývojáři, ať už se vývoj nachází v jakékoliv etapě. Je důležité nastavit správný harmonogram systému. Například určit období v rámci studijního roku, rozlišit, kdy nastane sudý a kdy lichý týden, v jaké dny se nahrazuje výuka, nebo vymezení krajních termínů v rámci studia. A sestavit přísná pravidla, která se musí striktně dodržovat. Obsluze systému by mělo být zajištěno podrobné školení o funkcích systému, jeho ovládání, seznámit je s historií. Naučit je

předcházet problémům, viz zahlcení systémů vlivem nedostatečně výkonného serveru. Uživatelé, kteří jsou v případě IS/STAG většinou z řad studentů, by se měli také seznámit se svojí rolí ve IS/STAG. Měli by projít školením jednotlivých funkcí dle modulů. Administrátoři by si měli navzájem sdělovat během každoročních setkání svoje poznatky a nálezy. To samé platí i pro vývojáře, ti by se navíc neměli stranit kontaktu s uživateli a konzultovat s nimi jejich nápady a návrhy. [3]

7. Přílohy

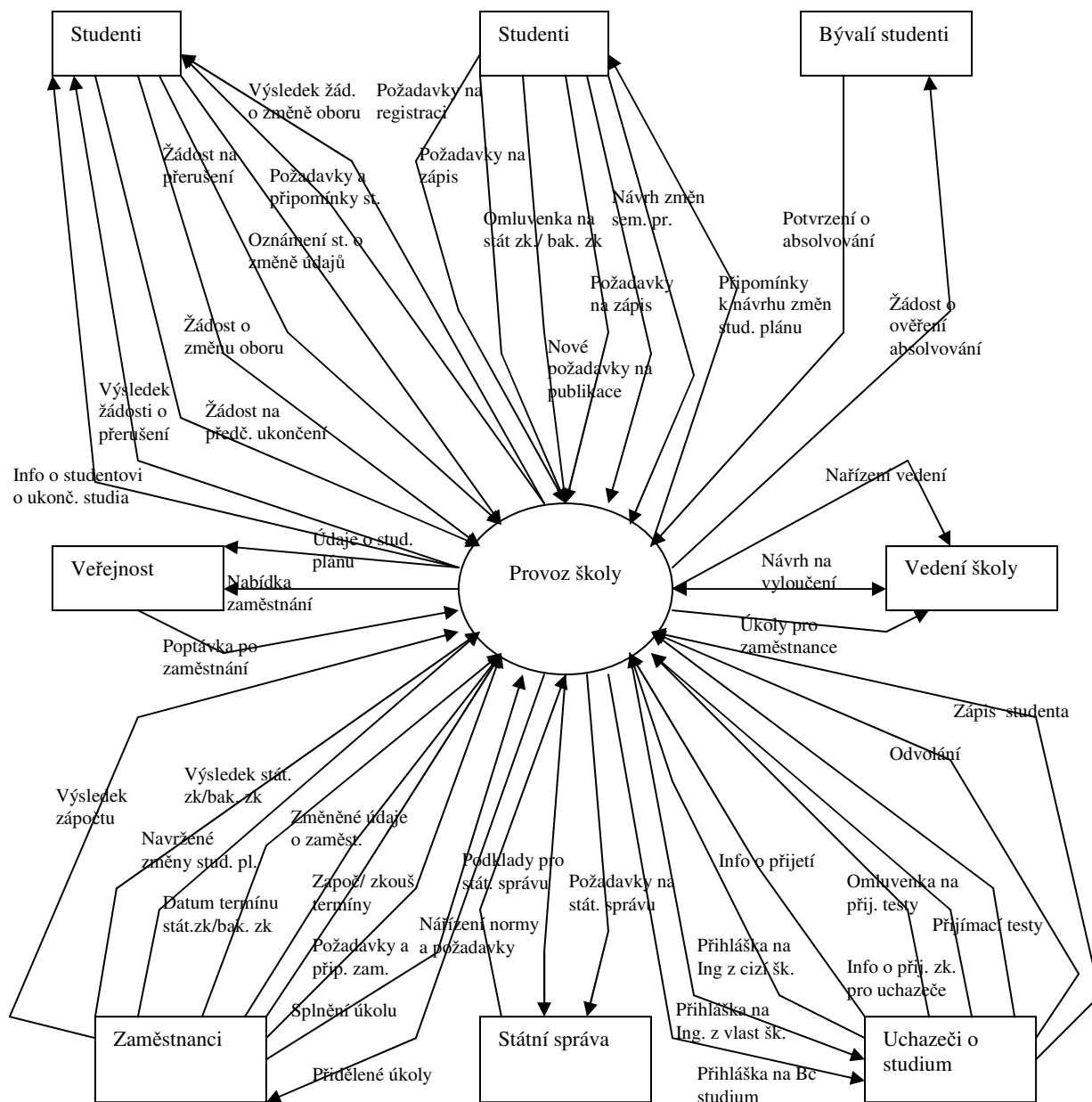
Příloha 1: Funkční model procesu vývoje IS/STAG

Kontextový diagram vývoje IS/STAG



Obrázek 25 Funkční model procesu vývoje IS/STAG [ŘEP1]

Příloha 3: Kontextový diagram VŠ podle IS/STAG



Obrázek 27 Kontextový diagram VŠ podle IS/STAG [ŘEP1]

8. Závěr

Ve své práci jsem se pokusil popsat návrh informačního systému studijní agentury, jinak řečeno IS/ STAG a nalezení jeho slabých stránek během vývoje a jejich případné odstranění. Informace jsem čerpal z literatury, z elektronických zdrojů a hlavně z osobní komunikace s členy vývojového týmu s pány Ing. Tomášem Kotoučem vedoucím analytického oddělení a Ing. Petrem Jirouškem hlavním vedoucím vývojového oddělení. Kteří ochotně odpovídali na mnou položené otázky a zodpověděli mi spoustu věcí, které jsem následně použil ve svojí bakalářské práci.

Jelikož IS/ STAG je velice obsáhlý a rozvětvený systém, nesnažil jsem popsat do hloubky jeho návrh a vývoj. Neboť jednotlivé etapy ve vývoji systému by vydaly na samostatné bakalářské práce. Mým cílem bylo podat jakýsi ucelený přehled vývoje, návrhu IS/ STAG se zveřejněním jeho slabých stránek, které se vyskytly v průběhu jeho vývoje a navržením jejich protiopatření. Při tvorbě práce jsem se setkal s problémem, jak moc do detailů mám určitou problematiku popsat. Abych neopustil zadané téma a zároveň abych neopomněl důležité informace týkající se bakalářské práce.

Na závěr bych chtěl podotknout, že vypracování této práce mě obohatilo o nové informace, týkající se informačních systémů a samotného IS/ STAG. A věřím, že z těchto znalostí budu moci čerpat v budoucnu.

9. Zdroje:

Internet

- [1] *Studijní agenda STAG* [online]. 2000 [cit. 2008-5-1]. Dostupný z WWW: <http://www.upce.cz/studium/studium-stag/>
- [2] *INFORMAČNÍ SYSTÉM STUDIJNÍ AGENDY* [online]. 2000 [cit. 2008-5-1]. Dostupný z WWW: http://support.zcu.cz/images/Sezn%C3%A1men%C3%AD_s_IS-STAG_2005.pdf
- [3] *IS_STAG_uvodni_prezentace.ppt* [online]. 2000 , 1.11.2006 [cit. 2008-5-1]. Dostupný z WWW: www.stag.zcu.cz/dokumenty/IS_STAG_uvodni_prezentace.ppt
- [4] *Popis IS/STAG* [online]. 2000 [cit. 2006-11-12]. Dostupný z WWW: <http://stag.zcu.cz/dokumenty/popis.php>
- [5] Vlastní zdroj
- [6] *Popis IS/STAG* [online]. 2000 [cit. 2008-5-1]. Dostupný z WWW: <http://stag.zcu.cz/dokumenty/rc.php>
- [7] POUR, Jan. *Uc_ist_kapitola.pdf(application/pdf objekt)* [online]. 2002, 2002 [cit. 2008-05-01]. Kódováno ve WIN-1250. Text v češtině. Dostupný z WWW: http://www.vsem.cz/data/docs/uc_ist_kapitola.pdf
- [8] QUEST SOFTWARE. *TOADSoft.com - Home of the World's #1 Tool for Database Applications Developers and DBAs!* [online]. 2006, 2006 [cit. 2008-05-10]. Kódováno ve WIN-1250. Angličtina. Dostupný z WWW: http://www.toadsoft.com/toad_oracle.htm
- [9] *Oracle Forms - Wikipedia, the free encyclopedia* [online]. 2008 , 28.4.2008 [cit. 2008-05-10]. Dostupný z WWW: http://en.wikipedia.org/wiki/Oracle_Forms
- [10] ORACLE. *Oracle Reports 10g* [online]. 2006, 2007 [cit. 2008-05-10]. Kódováno ve WIN-1250. Angličtina. Dostupný z WWW: <http://www.oracle.com/technology/products/reports/index.html>
- [11] ORACLE. *Oracle Designer - Product Information* [online]. 2006, 2007 [cit. 2008-05-11]. Kódováno ve WIN-1250. Angličtina. Dostupný z WWW: <http://www.oracle.com/technology/products/designer/index.html>
- [12] ORACLE. *Oracle 9i Database* [online]. 2006, 2007 [cit. 2008-05-11]. Kódováno ve WIN-1250. Angličtina. Dostupný z WWW: <http://www.oracle.com/technology/products/oracle9i/index.html>
- [13] NETOS. *Tenký klient - Terminal services - NETOS - správa sítí a jejich dodávky, outsourcing, zabezpečení a monitoring sítí, tenký klient - Citrix - Terminal services,*

poradenství v IT, konzultace [online]. 2004 [cit. 2008-05-11]. Kódováno ve WIN-1250. Čeština. Dostupný z WWW: <<http://www.netos.cz/produkty%20a%20sluzby/tenky%20klient/>>.

[14] *Index of /upce/pisr/cv-9-2006-04-18* [online]. [2006] , 1.5.2006 [cit. 2008-05-12]. Čeština. Dostupný z WWW: <<http://www.403.cz/upce/pisr/cv-9-2006-04-18/>>.

[15] Vývojové centrum ZČU. *Animace funkcí systému* [online]. 2002, 1. 8. 2008 [cit. 2008-08-10]. Dostupný z WWW: <http://stag.zcu.cz/animace/IS_STAG_sekretarky.ppt>.

[16] ZČU. *S/STAG - Předzáměs* [online]. 2002, 20. 8. 2008 [cit. 2008-08-20]. Čeština. Dostupný z WWW: <[https://stag.upce.cz/relay/predzapis/pg\\$_predzapis.startup](https://stag.upce.cz/relay/predzapis/pg$_predzapis.startup)>.

[17] ZČU. *S/STAG - Předzáměs* [online]. 2000, 20. 8. 2008 [cit. 2008-08-20]. Čeština. Dostupný z WWW: <[https://stag.upce.cz/relay/kpredzapis/pg\\$_kpredzapis.main](https://stag.upce.cz/relay/kpredzapis/pg$_kpredzapis.main)>.

[18] ZČU. *Studijní agenda* [online]. 2000 [cit. 2008-08-20]. Čeština. Dostupný z WWW: <[https://stag.upce.cz/apps/stag/userprefs/pg\\$_userprefs.osobni](https://stag.upce.cz/apps/stag/userprefs/pg$_userprefs.osobni)>.

[19] Vývojové centrum ZČU. *Animace funkcí systému* [online]. 2002, 1. 8. 2008 [cit. 2008-08-10]. Dostupný z WWW: <http://stag.zcu.cz/animace/IS_STAG_studijni_referentky.ppt>.

[20] KLEČKOVÁ. *Základy informačních systémů* [online]. Západočeská univerzita. 2008 [cit. 2008-08-28]. Čeština. Dostupný z WWW: <www.kiv.zcu.cz/~kleckova/Predmety/ZIS/download/ZIS_11n_2008.pdf>.

Diplomové práce

[MICH1] Michajlov M.: „Vizualizace studijního plánu“, Západočeská Univerzita, Plzeň 2002

Literatura

[VOŘ1] Voříšek, J.: „Strategické řízení informačního systému a systémová integrace“, Management Press, Praha 1997

[ŘEP1] Řepa, V. : „Analýza a návrh informačních systémů“, EKO Press, Praha 1999

[JAY1] Jayaratna, N. : „Understanding and Evaluating Methodologies“, McGraw Hill Europe, 1994

[VRA1] Vrána, I., Richta, K.: „Zásady a postupy při zavádění podnikových informačních systémů“, Praha: GRADA, 2004, 188 s. ISBN 80-247-1103-6

[KRA1] Král, J.: „Informační systémy“, Veletiny: Science, 1998, 356 s. ISBN 80-86083-00-4

[BUC1] Buchacelová, A.: „Metodiky vývoje a údržby informačních systémů“, Praha: Grada Publishing, a. s., 2004, ISBN 80-247-1075-7

10. Seznam zkratek

CASE	Computer-Aided Software Engineering
CPM	Critical Path Method
CVS	Concurrent Versions System
DFD	Data Flow Diagram
DOS	Disk Operating System
ERA	Entity- Relation- Atribut
FSD	Fiction Structured Diagram
GHz	Giga Herz
HTTPS	Hyper Text Transfer Protokol over Secure Socked Layer
HW	Hardware
ICTS	International Credit Transfer System
IEM	Information Engineering Metodology
IS	Informačn
MB	Mega Byte
OLTP	Online Transaction Processing
OS	Operation Systém
PC	Personal Computer
PERT	Program Evaluation and Review Technice
RAM	Random-Access Memory
RAP	Rapid Application Prototyping
SDM	System Development Metodology
SQL	Structured Query Language
SW	Software
TASW	Type Aplication Software
V- MODEL	Vorgehens Modell
WEB, WWW	World wide web
VGA	Video Graphics Array
VPD	Virtual Private Database
ZČU	Západočeská Univerzita