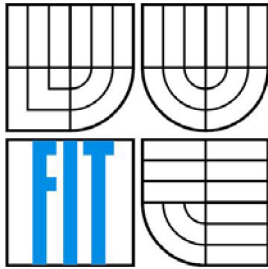


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU ŘÍZENÍ KNOW-HOW V ICT SPOLEČNOSTI

ANALYSIS AND DESIGN OF INFORMATION SYSTEM FOR CONTROLLING KNOW-HOW IN ICT
COMPANY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ POSPÍŠIL

VEDOUCÍ PRÁCE
SUPERVISOR

RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2007

Analýza a návrh informačního systému řízení know-how v ICT společnosti

Analysis and Design of Know-How Management System in ICT Company

Vedoucí: Kreslíková Jitka, RNDr., CSc., UIFS FIT VUT

Konzultant: Beránek Marek, Unicorn

Oponent: Bartík Vladimír, Ing., Ph.D., UIFS FIT VUT

Přihlášen: Pospíšil Jiří

Zadání:

1. Seznamte se s metodikou IBM Rational Unified Process (RUP).
2. Podrobně se seznamte s inepční a elaborační fází metodiky RUP.
3. Provedte inepční fází informačního systému řízení know-how v ICT společnosti.
4. Provedte elaborační fází navrženého systému. V rámci eleborační fáze vytvořte funkční prototyp navrženého informačního systému.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti využití metodiky RUP.

Část požadovaná pro obhajobu SP:

Splnění bodů zadání 1 - 3.

Kategorie: Softwarové inženýrství

Literatura:

Štork, R., Vitouš, O.: Rational Unified Process - stručný průvodce. Praha, Unicorn Multimedia 2000. ISBN 8023863584.

Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley 2005, ISBN 0321267974.

Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley, 2003, ISBN 0321197704.

Komentář:

Zadání z praxe. Zadává Unicorn a.s., Karolinská 650/1, 186 00 Praha 8,

Konzultuje: Marek Beránek, marek.beranek@unicorn.cz

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan

Jméno a příjmení: Jiří Pospíšil

Bytem: Oflenda 2, 539 01, Hlinsko v Čechách

Narozen/a (datum a místo): 26.10.1981 v Pardubicích

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem Božetěchova 2, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen „nabyvatel“)

Čl. 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
 diplomová práce

Název VŠKP: ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU
ŘÍZENÍ KNOW-HOW V ICT SPOLEČNOSTI

Vedoucí/ školitel VŠKP: RNDr. Jitka Kreslíková, CSc.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v*:

- tištěné formě – počet exemplářů
- elektronické formě – počet exemplářů ... 2.....

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

* hodící se zaškrtněte

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

Diplomová práce se zabývá problematikou návrhu informačního systému. Návrh systému je prováděn pomocí metodiky Rational Unified Process. Práce vytváří soupis požadavků a pohledů na systém. Provádí analýzu a konkrétní návrh informačního systému. Dle metodiky RUP jsou uskutečněny první dvě fáze této metodiky, fáze inepční a fáze elaborační. Vytvořená elaborační část dokumentu je podkladem pro vytvoření programového prototypu v prostředí Ruby on Rails za použití jazyka Ruby v kombinaci s HTML kódem.

Klíčová slova

Know-how, prvek, jazyk UML, metodika zabalený zdravý rozum (RUP), testování znalostí, analýza a návrh informačního systému, zahajovací fáze, přípravná fáze, konstrukční fáze, předávací fáze, diagram použití, diagram tříd, sekvenční diagram, problémová doména, softwarová architektura, Ruby.

Abstract

The thesis deals with a problem of the design of information system. The design of system is provided with Rational Unifeid Process methodology. This thesis creates a list of requests to system. It makes an analysis and a design of current information system. It useses a RUP methodology to realize two first phases, inception phase and elaboration phase. Created elaboration phase of document is a base for creating programming prototype in Ruby on Rails environment using Ruby language with combination of HTML code.

Keywords

Know-how, element, language UML, methodology Rational Unified Process (RUP), knowledge testing, analyze and design of information system, inception phase, elaboration phase, construction phase, transition phase, Use Case diagram, Class diagram, Sequence diagram, problem domain, software architecture, Ruby

Citace

Jiří Pospíšil: Analýza a návrh informačního systému řízení know-how v ICT společnosti, diplomová práce, Brno, FIT VUT v Brně, 2007

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením RNDr. Jitky Kreslíkové, CSc. Další informace mi poskytli zástupci firmy Unicorn a.s. Uvedl jsem všechny literární prameny a publikace, ze kterých jsme čerpal.

.....

Jméno Příjmení

Datum

Poděkování

Děkuji RNDr. Jitce Kreslíkové, CSc. za vedení a poskytnutí studijních podkladů. Dále Markovi Beránkovi ze společnosti Unicorn a.s. za poskytnutí naváděcích materiálů k práci. Také děkuji Vítkovi Stinkovi ze stejné společnosti za odborné konzultace k projektu a profesní zasvěcení do některých zásad této problematiky v praxi.

© Jiří Pospíšil 2007

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je ochráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	8
1.1	Cíl práce.....	8
1.2	Rozdělení práce	9
2	Metodika pro zadání systému	10
2.1	Rational Unified Process.....	10
2.2	Základní model RUP	12
2.3	Iterativní vývoj	15
2.4	Systémová architektura	20
2.5	Případy užití	23
3	Vize.....	25
3.1	Předmluva.....	25
3.2	Obchodní umístění.....	25
3.3	Zainteresané osoby a uživatelské popisy	27
3.4	Přehled produktu.....	31
3.5	Rysy produktu	33
3.6	Omezení	34
3.7	Kvalita systému, mezní hodnoty	34
3.8	Precedence a priorita.....	35
3.9	Jiné požadavky produktu	35
3.10	Požadavky dokumentace	36
4	Diagramy použití (USE-CASE)	38
4.1	Notace Use-Case diagramu.....	38
4.2	Předmluva.....	39
4.3	Správa údajů o zaměstnancích.....	41
4.4	Správa pohovoru.....	45
4.5	Správa elementů KH zaměstnance	48
4.6	Žádost o přidělení znalosti	51
4.7	Správa elementů KH.....	52
4.8	Správa údajů o projektu	56
4.9	Správa požadovaných elementů.....	60
5	Doplňková specifikace	63
5.1	Předmluva.....	63
5.2	Použitelnost	63

5.3	Spolehlivost.....	64
5.4	Výkonnost	64
5.5	Omezení Návrhu	65
6	Model problémové domény	66
7	Plán realizace	67
7.1	Funkčnosti realizované v jednotlivých etapách	67
7.2	Výstupy realizace a její časový harmonogram.....	67
8	Softwarová architektura	68
8.1	Předmluva	68
8.2	Reprezentace architektury	68
8.3	Cíle a omezení architektury	68
8.4	Use-case pohled	69
8.5	Logický pohled	69
8.6	Pohled nasazení	77
8.7	Výkonnost	77
8.8	Kvalita	78
9	Uživatelské rozhraní	79
9.1	Přihlášení	79
9.2	Správa zaměstnanců	80
9.3	Správa elementů KH.....	80
9.4	Správa elementů KH zaměstnance	81
9.5	Zobrazení podrobností o elementu	82
10	Závěr.....	83
11	Literatura.....	84
11.1	Seznam použité literatury a materiálů.....	84
11.2	Použité nástroje	84
11.3	Slovníček pojmů.....	84
11.4	Seznam obrázků	86
12	Přílohy.....	87

1 ÚVOD

1.1 Cíl práce

Cílem této diplomové práce je navrhnout informační systém (IS) pro sběr znalostí zaměstnanců softwarové firmy. Systém má sloužit jako sběrač informací o znalostech zaměstnanců. Umožňuje jejich skladování, možnost aktualizace a výpisů. Shromážděné informace o znalostech jednotlivých zaměstnanců slouží, k přehledné evidenci možností firmy při zpracovávání projektů. Pro rozsáhlejší firmu zabývající se vývojem software, kterou je například právě Unicorn a.s. poslouží IS jako efektivní řešení, pro přesouvání zaměstnanců v rámci jednotlivých projektů. Výběr zaměstnanců pro projekty, bude tedy probíhat ve spolupráci s informačním systémem. Při návrhu IS byla využita metodika Rational Unified Process, vyvinutá společností Rational Software Corporation.

Samotná práce se zabývá metodikou RUP a její použitelností při vytváření jednotlivých dokumentačních částí k projektu. Ukazuje jak vést analýzu a vývoj projektu pomocí metodiky RUP pro nově vytvářený IS. Vytvářená dokumentace k této diplomové práci se zabývá prvními dvěma fázemi a to fází inepční a elaborační. Řeší tedy návrh systému do stavu před výslednou realizací. V programové podobě je vytvořen prototyp na základě elaborační fáze. Prototyp je napsán v programovacím jazyku *Ruby*. Tento jazyk, stejně jako jeho prostředí *Ruby on Rails* je přizpůsoben novým požadavkům na vytváření softwarového produktu.

Diplomová práce nemá návaznost na ročníkový projekt, navazuje však na semestrální projekt. Z tohoto projektu byla pro diplomovou práci převzata kapitola 3. Vize. Kapitola charakterizuje vývoj systému v počáteční inepční fázi. Výstup z této kapitoly poskytl diplomové práci tvorbu následných kapitol. Jedná se zejména o kapitoly Use-Case, dále Model problémové domény a Plán realizace.

1.2 Rozdělení práce

Jednotlivé kapitoly postupně rozebírají problematiku analýzy až po část návrhu. Kapitola **2 Metodika pro zadání systému** je teoretickou částí a obsahuje souhrnné informace o metodice RUP a jejím zavedení. Kapitola **3 Vize** shromažďuje, analyzuje a definuje klíčové potřeby a vlastnosti informačního systému. Je zaměřena na vlastnosti vyžadované koncovými uživateli. Detaily, jak uspokojit tyto potřeby jsou popsány v případě použití a doplňkových specifikacích. Kapitola **4 Diagramy použití (use-case)** vyjadřuje vztahy mezi Aktéry mimo systém a Use Case uvnitř systému, zachycuje funkce, které systém poskytuje a definuje jejich využití. Detailní popis komunikace mezi aktory a systémem není v tomto diagramu obsažen. Další kapitolou je **5 Doplňková specifikace**, ta definuje požadavky systému, které nejsou zahrnuty v případech použití. Tyto požadavky, s use-case modelem tvoří úplnou množinu požadavků na systém. Požadavky se zabývají vlastnostmi systému jako jsou: použitelnost, spolehlivost, výkonnost a omezení návrhu. Následuje kapitola **6 Model problémové domény**, tento model znázorňuje množinu vzájemně závislých konceptů a vazeb systému. Reprezentuje obchodní pohled, modeluje data a aktivity. Kapitola **7 Plán realizace** rozděluje realizaci softwarové části na jednotlivé etapy. Každá z těchto etap má specifikován rozsah prováděných prací. Kapitola **8 Softwarová architektura** poskytuje obsáhlý přehled systému z pohledu rozdílné architektury. Skládá se z jednotlivých pohledů. *Use-case pohled* popisuje množinu případů užití, které mají značný dopad na architekturu a které reprezentují významnou funkcionalitu. *Logický pohled* popisuje nejdůležitější třídy, jejich organizaci v balíčky a do vrstev. Popisuje nejdůležitější use-case realizace, jednotlivé příklady realizací případů užití a k nim příslušné sekvenční diagramy. *Pohled nasazení* ukazuje vlastnosti systému plynoucí z reálného nasazení. Poslední objemově obsáhlou kapitolou je kapitola **9 Uživatelské rozhraní**. Ta dává možnost nahlédnout do realizace prototypu a jeho vzhledu.

2 METODIKA PRO ZADÁNÍ SYSTÉMU

Tato část diplomové práce popisuje zvolenou metodiku, kterou je Rational Unified Process. Text v této kapitole, je výkladem z literatury [1] s rozšířením informací z [2].

2.1 Rational Unified Process



Rational Unified Process je softwarový vývojový proces. Poskytuje ucelený přístup k určování úkolů a odpovědnosti v příslušné organizaci. Jeho prvotním cílem je zajistit výrobu vysoce kvalitního produktu, který bude vyhovovat všem požadavkům uživatele a zároveň bude odpovídat normám a dodržovat daný rozpočet.

Rational Unified Process byl vyvinut společností Rational Software. Jde o procesní strukturu, která může být adaptována a rozšířena podle potřeby organizace.

2.1.1 Iterativní vývoj

Iterativní vývoj umožňuje managementu provádět taktické změny v rámci konkurenčního boje. V jednotlivých iteracích může produkt realizovat omezenou funkčnost, část funkčnosti může být nahrazena jinými, již existujícími technologiemi.

Na základě vyhodnocení výstupů ukončené iterace mohou být upraveny plány dalších iterací tak, aby odrážely aktuální požadavky trhu. Jestliže se v tomto okamžiku rozhodneme zastavit vývoj jedné komponenty systému, můžeme původně plánované zdroje alokovat pro jiné aktivity.

2.1.2 Řízení požadavků

Řízení požadavků je systematický přístup k zjišťování, organizování a dokumentaci měnících se požadavků. Takovýto přístup umožňuje adekvátní kontrolu celého projektu. Požadavky jsou monitorovány a zpracovány do procesu v průběhu celého vývoje softwaru.

Tím se především snižují náklady a také se tak zamezuje různým prodlením a odkladům plánovaných termínů. Průběžně se zlepšuje kvalita softwaru a tím se zvyšuje zákaznickova spokojenost.

2.1.3 Komponentová architektura

Na počátku procesu vývoje softwaru se veškerá snaha zaměřuje především na vytvoření pružné systémové architektury. Rational Unified Process poskytuje systematický návod na vytvoření návrhu, vývoj a ověření takové architektury, která může být dokonce použita i v jiném projektu.

Softwarovou komponentu lze definovat jako netriviální část softwaru, modul, package nebo subsystém, který plní určitou funkci, má jasné hranice a může být součástí definované architektury.

2.1.4 Vizualní modelování

Pro účely vizuálního modelování nabízí Rational Unified Process standardní vyjadřovací mechanismy. Unified Modeling Language (UML) je grafický jazyk pro vizualizaci, specifikaci,

konstrukci a dokumentaci artefaktů. Poskytuje standardní prostředky pro znázornění systémových objektů, databázových schémat, použitých softwarových komponent apod.

UML je společný jazyk k vyjádření různých modelů, ale neříkám nám, jak vyvíjet software. Rational Unified Process se snaží tento jazyk co nejlépe využívat k modelování. Popisuje, které modely jsou potřeba, proč je potřebujeme a jak je zhotovit.

2.1.5 Kvalita

V Rational Unified Process není možná role, která by se zabývala jen kvalitou produktu. Kvalita totiž není záležitostí několika specialistů. Kvalita je odpovědnost každého člena týmu k organizaci.

Rational Unified Process se zaměřuje především na ověření a objektivní ohodnocení toho, zda produkt odpovídá požadované úrovni kvality.

2.1.6 Konfigurační a změnové řízení

V průběhu vývoje informačního systému dochází k mnoha modifikacím. Díky flexibilním změnám požadavků, technologií a samotných strategických cílů se iterativní vývoj zaměřuje především na změny v procesu vývoje.

Právě zaměření na potřeby vývojové organizace dělá z řízení změn systematický přístup ke správě změn v požadavcích, návrhu a implementaci. Řízení změn se úzce váže na konfigurační řízení a měření.

2.1.7 Vývoj orientovaný na Případ užití

Často je velmi obtížné popsat na základě modelu, jak systém zajišťuje požadované funkce. Tato nesnáze vychází z potřeby trvalé a zřejmé souvislosti skrz celý systém, který plní určité úkoly. V Rational Unified Process je tato souvislost znázorněna modely Případů užití, které definují chování systému vzhledem k uživateli.

Rational Unified Process je proces orientovaný na Případ užití, tzn. že modely Případů užití (tedy modely věcné funkčnosti systému z pohledu uživatele – též Use-Case modely) jsou základem pro všechny další vývojové aktivity.

2.1.8 Podpora nástrojů

Aby byl proces efektivní, musí být podpořen adekvátními nástroji. Rational Unified Process integruje rozsáhlou paletu softwarových nástrojů, které umožňují automatizaci mnohých aktivit. Tyto nástroje se používají především při vizuálním modelování, programování, testování a při samotném řízení projektu.

Automatizace procesů je neocenitelná především při sdílení informací spojených s řízením změn a konfiguračním řízením v rámci středně velkých a velkých vývojových týmů.

2.1.8.1 *Podpůrné nástroje Rational Unified Process*

- Rational Rose - vizuální modelování.
- SoDa - automatizovaná tvorba dokumentace.
- RequisitePro - testování.

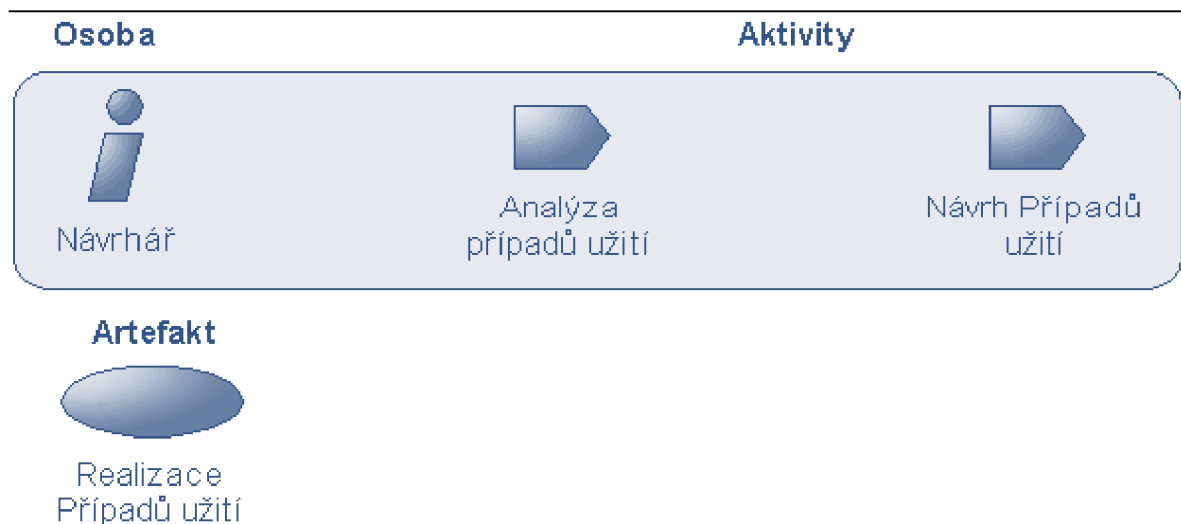
- ClearQuest - řízení změn, konfigurační řízení.

2.2 Základní model RUP

Rational Unified Process definuje kdo, jak, kdy a co dělá. K tomu používá čtyři základní prvky:

- Osoby (kdo?),
- Aktivity (jak?),
- Artefakty (co?),
- Pracovní postupy (kdy?).

Použití těchto prvků je znázorněno viz. obrázek 2.2.1



obrázek 2.2.1: Osoba, aktivity a artefakty

2.2.1 Základní vztahy

- Osoba definuje chování a odpovědnost jednotlivce nebo týmu.
- Chování je vyjádřeno aktivitou, kterou osoba vykonává. Každá osoba je spojena se souborem aktivit.
- Odpovědnost každé osoby je vyjádřena ve spojení s artefaktem, který osoba vytváří, upravuje nebo kontroluje.

2.2.2 Osoby

Osobu si můžeme představit jako roli v divadelní hře. Jeden člověk může hrát více rolí a naopak do jedné role může být obsazeno více lidí (viz obrázek 2.2.2). Tato odlišnost je důležitá, protože je přirozené myslet osobou jednotlivce nebo tým.

V Rational Unified Process osoba definuje, jak má jednotlivec (který je do ní obsazen) dělat svojí práci a jakých artefaktů je vlastníkem. Toto rozdělení má na starosti projektový manažer, který plánuje projekty a jejich personální obsazení.

2.2.2.1 Příklady osob

Systémový analytik	Řídí a koordinuje proces zpracování požadavků, tvorbu modelů Případů užití.
Návrhář	Vytváří model tříd a určuje, jak mají být zakomponovány do implementačního prostředí.
Návrhář testů	Je zodpovědný za plánování, návrh, implementaci a vyhodnocení testů systému.

Zdroj	Osoba 	Aktivity 
Pavel	Návrhář	Návrh projektu
Marie	Autor Případů užití	Detail Případů užití
Josef	Návrhář Případů užití	Návrh Případů užití
Silva	Revizor návrhu	Revize návrhu
Štefan	Architekt	Analýza architektury Návrh architektury

obrázek 2.2.2: Lidé a osoby

2.2.3 Aktivity

Aktivita je jednotka práce, která může být osobě přidělena. Aktivita má jasný cíl, většinou se jedná o tvorbu nebo správu artefaktů (například model, třída, plán,...). Každá aktivita je přidělena specifické osobě.

Aktivita musí být použitelná jako prvek plánování. Pokud je příliš malá, může být opomenuta, a pokud je naopak příliš velká, lze pokrok vyjádřit po částech aktivity. V objektově orientované terminologii je osoba aktivní objekt. Aktivity, které osoba vykonává, jsou pak operace vykonávané tímto objektem.

2.2.4 Artefakty

Artefakt představuje informaci, která je vytvořena, modifikována a používána v procesu vývoje. Je hmatatelným výsledkem projektu. Artefakty jsou používány jako vstup pro vykonání aktivity určitou osobou a jsou cílem nebo výstupem takovéto aktivity.

Artefakty nemusí být vždy dokumenty. Efektivnější je tvorba artefaktů vhodnými nástroji. Pokud je to nutné, lze dokumenty generovat pomocí těchto nástrojů. Takovéto dokumenty jsou pak založeny na aktuální verzi artefaktů.

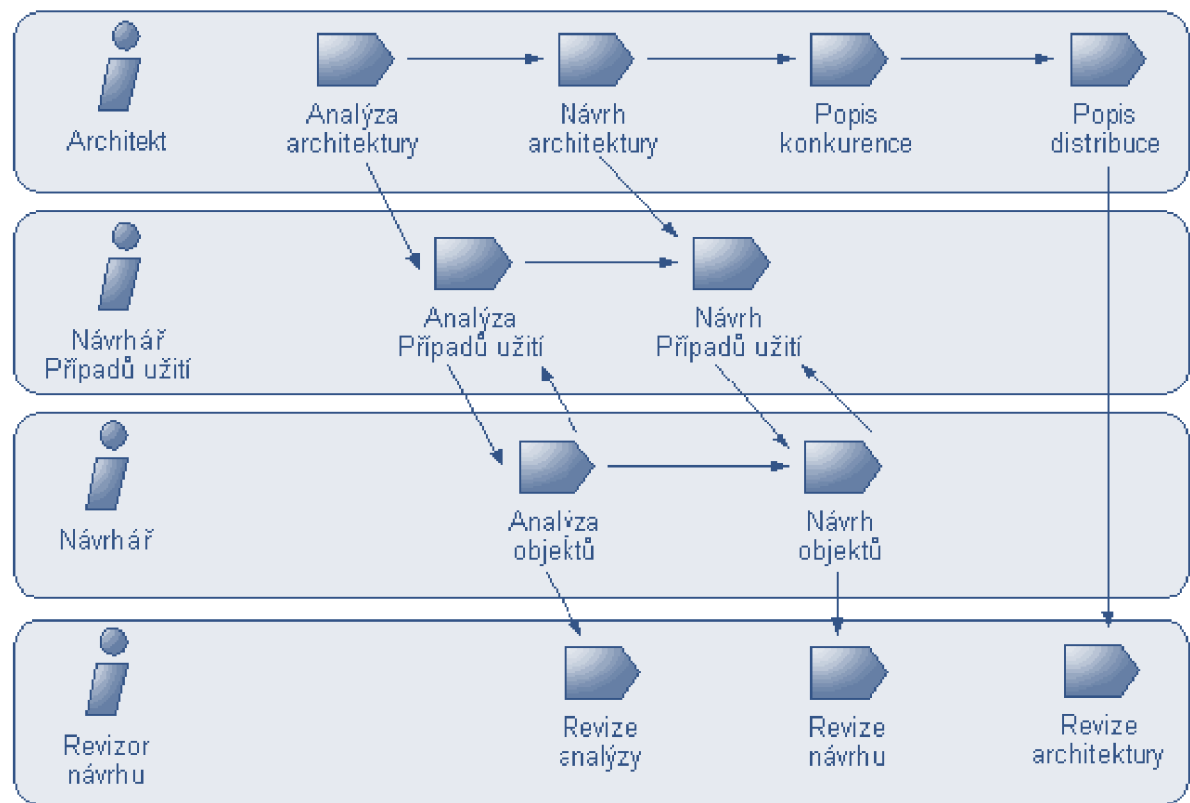
2.2.5 Pracovní postupy

Pouhý výčet všech osob, aktivit a artefaktů ještě nevytváří proces. Je třeba najít způsob, jak popsat posloupnost aktivit, které produkují hodnotné výsledky a ukazují interakce mezi osobami. Pracovní postup je posloupnost aktivit, které přinášejí předem definované výstupy (viz obrázek 2.2.3).

Často není možné znázornit všechny závislosti mezi aktivitami, zvláště jsou-li velmi úzce propojeny a týkají-li se stejné osoby či jednotlivce. Postupy týkající se následného odvozování jednotlivých artefaktů můžeme pak popsat v detailu pracovního postupu.

2.2.5.1 *Rozdělení pracovních postupů*

- Základní pracovní postupy,
 - Obchodní modelování,
 - Správa požadavků,
 - Analýza a návrh,
 - Implementace,
 - Testování,
 - Zavedení,
 - Řízení projektů,
 - Konfigurační řízení,
 - Správa prostředí.
- Iterační pracovní postupy,
 - Prezentují děje v průběhu jednotlivých iterací, návaznost iterací.
- Detaily pracovních postupů.
 - Týkají se specifické skupiny aktivit, které jsou úzce propojeny a jsou vykonávány společně.



obrázek 2.2.3: Příklad pracovního postupu

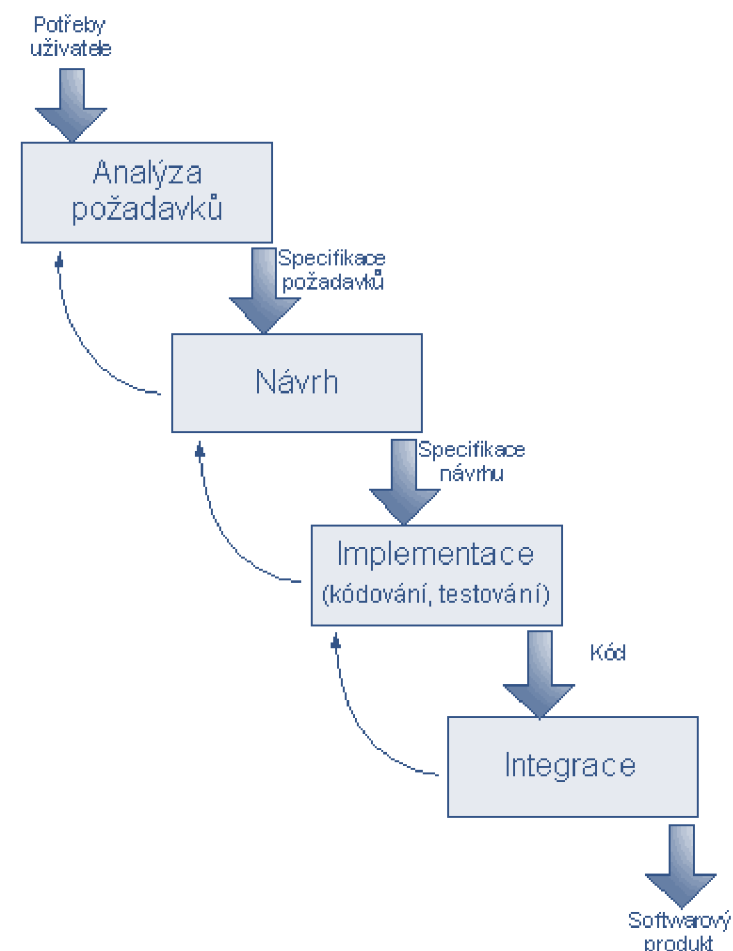
2.3 Iterativní vývoj

Iterativní vývoj vychází z dnes běžně používaného sekvenčního procesu vývoje (vodopád - viz obrázek 2.3.1). Ten funguje u malých projektech, ale způsobuje problémy u rozsáhlých řešení. Proto úlohu poněkud upravíme. Velký projekt rozdělíme na řadu malých částí (vodopádů).

V každé fázi nejdříve získáme požadavky, vytipujeme rizika, navrhujeme řešení, adekvátní část implementujeme a ověříme. Poté zapracujeme další požadavky, navrhujeme rozšíření, zapracujeme jej a opět ověříme. To je iterativní vývoj. RUP předpokládá 4 základní fáze, každá z nich je ukončena milníkem (okamžik, kdy rozhodujeme, zda ve vývoji pokračovat, změnit postup nebo jej ukončit).

2.3.1 Přehled fází a milníků

Fáze	Popis fáze	Milník
Počáteční fáze	Specifikace vize konečného produktu, rozsahu projektu	Rozsah systému
Elaborace	Plánování nezbytných aktivit a zdrojů, specifikace požadavků, návrh architektury	Architektura systému
Konstrukce	Výroba produktu, příprava nasazení	Úvodní funkčnost
Zavedení	Nasazení produktu, instalace, školení a podpora	Nasazení produktu



obrázek 2.3.1: Sekvenční proces

2.3.2 Počáteční fáze

V této fázi je založen obchodní případ a určen rozsah projektu. Je nutno identifikovat prvky, s nimiž bude systém komunikovat, a definovat povahu těchto prvků, což znamená identifikovat všechny Případy užití a popsat několik nejvýznamnějších z nich.

To vše se děje na základě vyhodnocení požadavků a omezení systému, stanovení kritérií, která systém musí splňovat, zvážení možných alternativ v poměru cena/termín/výkon, definování použitelných architektur (a komponent) a rozhodnutí, co se vyrobí/koupí/znovu použije.

Výstupem jsou artefakty

- Vize (dokument obsahující obecnou vizi klíčových požadavků projektu, jeho hlavní rysy a omezení).
- Model Případů užití.
- Počáteční glosář projektu.
- Počáteční obchodní případ (obchodní souvislosti, kritéria úspěchu, finanční rozpočet).
- Počáteční ohodnocení rizik.

- Plán projektu (přehled fází a iterací).

2.3.2.1 Milník Rozsah systému

Na konci Počáteční fáze se objevuje první milník projektu: Rozsah systému. Jeho smyslem je navodit konsenzus mezi osobami zodpovědnými za realizaci projektu a objektivně prokázat, že projekt je realizovatelný v navržené kvalitě, kvantitě, termínu a rozpočtu (KKTR).

Jako podklady slouží výstupy Počáteční fáze. Ty musí mapovat všechny veličiny, které podstatně ovlivňují KKTR. V případě, že se nepodaří dosáhnout tohoto milníku, musí být projekt přehodnocen, případně zastaven.

Kritéria pro posouzení Počáteční fáze

- Souhlas všech zúčastněných s definicí rozsahu projektu, s cenovým a časovým odhadem.
- Porozumění požadavkům.
- Odhad nákladů, časový rozvrh, priority, rizika.
- Šířka a hloubka prototypu architektury.
- Aktuální náklady versus náklady plánované.

2.3.3 Elaborální fáze

Cílem této fáze je analyzovat klíčové problémy, vyvinout základy architektury, vytvořit plán projektu a odhadnout nejrizikovější prvky projektu. Součástí elaborace je návrh funkčního prototypu architektury.

Toto úsilí by se mělo zaměřit především na kritické Případy užití, identifikované v Počáteční fázi, které typicky zahrnují hlavní technická rizika projektu. Dá se říci, že tato fáze je nejrizikovější ze všech čtyř fází. Na jejím konci nastává důležitý okamžik, kdy se rozhodne o pokračování projektu.

Výstupem jsou artefakty

- Model Případů užití (Use-Case model).
- Dodatečné požadavky (včetně nefunkčních požadavků a požadavků, které nejsou spojeny se specifickým Případem užití).
- Popis softwarové architektury.
- Spustitelný prototyp architektury.
- Revidovaný seznam rizik a revidovaný obchodní případ.
- Vývojový plán pro celý projekt (včetně hrubého plánu s iteracemi a hodnotícími kritérii pro každou iteraci).
- Aktualizovaný vývojový případ s popisem procesu, který se má použít.

- Předběžný uživatelský manuál.

2.3.3.1 Milník Definice architektury

V tomto dalším důležitém okamžiku se prověřují detailní vlastnosti systému a jeho rozsah. Klíčovými úkoly jsou výběr architektury a odstranění hlavních rizik vyřešením technologických problémů nebo zvolením alternativních postupů.

Jako podklady výstupu Elaboráční fáze. Ty musí obsahovat prototyp architektury, konkrétní plán pro další fázi, soupis všech známých rizik, návrhy jejich eliminace, případně alternativní postupy. V případě, že se nepodaří dosáhnout tohoto milníku, musí být projekt přehodnocen, případně zastaven.

Kritéria pro posouzení Elaboráční fáze

- Je vize produktu stabilní?
- Je architektura stabilní?
- Je na prototypu vidět, že hlavní rizika byla nalezena a korektně vyřešena?
- Je plán pro další fázi dostatečně detailní a přesný? Je podpořen důvěryhodnými odhady?
- Souhlasí všichni zúčastnění s tím, že je možno dané vize dosáhnout, jestliže bude proveden aktuální plán - podaří se tedy vyvinout kompletní systém s použitím současné architektury?
- Lze akceptovat skutečné výdaje v poměru k plánovaným?

2.3.4 Konstrukční fáze

Během Konstrukční fáze jsou navrženy všechny zbývající komponenty a vlastnosti aplikace, jsou vyvinuty a integrovány do produktu. Všechny vlastnosti systému jsou důkladně otestovány. Konstrukční fáze je výrobním procesem, v němž se klade důraz na řízení zdrojů a kontrolu kvality, kvantity, termínů a rozpočtu.

Jednou z nejdůležitějších kvalit architektury je jednoduchost její konstrukce, proto je vyzdvihován vyvážený vývoj architektury a plánu během Elaboráční fáze. Výstup z konstrukční fáze je připraven k předání koncovým uživatelům.

Výstupem jsou artefakty

- Softwarový produkt integrovaný na odpovídajících platformách.
- Uživatelský manuál.
- Popis současné verze.

2.3.4.1 Milník Beta verze

Na konci Konstrukční fáze je třetí milník projektu: Beta verze. V tomto okamžiku prověřujeme, zda software, provozní prostředí a uživatelé jsou připraveni k nasazení systému do provozu, aniž bychom zúčastněné strany vystavovali velkým rizikům.

V případě, že se nepodaří dosáhnout tohoto milníku, musí být zavedení odloženo a je naplánována náhradní verze, která odstraní rizika nasazení systému do provozu, případně musí být upraven Plán nasazení.

Kritéria pro posouzení Konstrukční fáze

- Je tato verze dostatečně stabilní a zralá na to, aby byla nasazena mezi běžné uživatele?
- Jsou všichni zúčastnění připraveni na přechod produktu mezi uživatele?
- Jsou skutečné náklady stále přijatelné ve srovnání s těm plánovanými?

2.3.5 Fáze nasazení

Tato fáze se zaměřuje na činnosti potřebné k předávání softwaru do provozního prostředí. Zpravidla zahrnuje několik iterací (například Uživatelské akceptační testy, Pilotní provoz, Rutinní provoz) včetně servisních buildů a patchů řešících chyby.

Velké úsilí je vynaloženo na přípravu uživatelské a servisní dokumentace, školení uživatelů a podporu uživatelů (HelpDesk).

Fáze nasazení zahrnuje zejména

- Beta testování/Pilotní provoz.
- Paralelní nasazení společně s nahrazovaným systémem.
- Konverzi operačních databází.
- Školení administrátorů a správců, případně též uživatelů.
- Předání systému do rutinního provozu, zajištění HelpDesku.

2.3.5.1 Milník Nasazení

V tomto okamžiku se rozhodne, zda byly obsaženy stanovené cíle a zda je možno začít práci na jiném vývojovém cyklu. V některých případech se tento milník kryje s ukončením počáteční fáze dalšího cyklu.

Důležitá je správa funkčnosti komunikačního kanálu mezi servisním týmem a koncovými uživateli (zpravidla řeší centrální HelpDesk). Do servisního týmu je vhodné zařadit několik lidí (dle velikosti systému) z vývoje. Ti zaškolí ostatní a poté se mohou vrátit zpět do realizace.

Kritéria pro posouzení Fáze nasazení

- Jsou uživatelé spokojeni?
- Jsou skutečné náklady stále přijatelné v poměru k plánovaným?

2.3.6 Shrnutí

Sekvenční proces (neboli vodopád) je znamenitý pro menší projekty, ale selhává v delších rizikových projektech. Iterativní přístup rozděluje vývojový proces do několika iterací. Každá z nich je vlastně malým vodopádem.

Každá iterace zahrnuje základní aktivity (sběr požadavků, návrh, implementaci a vyhodnocení). Pro účely řízení projektu životní cyklu projektu rozdělen do posloupnosti čtyř fází: Počáteční fáze, Elaborace, Konstrukce a Fáze nasazení.

2.3.7 Výhody iterativního postupu

- Včasná eliminace rizik.
- Pružná reakce na přicházející změny.
- Zaměření na znovu-použitelnost komponent.
- Efektivní plánování zdrojů.
- Průběžné vyhodnocování kvality produktu.

2.4 Systémová architektura

Jestliže jsou hotové modely systému, je poměrně jasné, co všechno je od systému očekáváno, ale už nemusí být zřejmé, jak toho bude docíleno. Do modelů je nutné promítnout architekturu systému, která ozřejmí, jakým způsobem budou logické komponenty skutečně implementovány.

Architektura systému obsahuje důležitá rozhodnutí o struktuře systému, použitých komponentách a jejich vzájemných vztazích, způsobu spolupráce (výběr komunikačních rozhraní) apod. Architektura je tedy částí návrhu systému. Zachycuje nejdůležitější prvky, které mají vliv na kvalitu systému, především pak na výkon.

2.4.1 Využití modelů

Velká část Rational Unified Process je zaměřena na vizuální modelování. Modely nám pomáhají pochopit problém a znázornit ho spolu s jeho řešením. Model je totiž zjednodušením reality, které nám pomáhá zvládnout složité a rozsáhlé systémy.

Právě výběr modelů a techniky využití k jejich vyjádření má silný vliv na náš pohled na celý systém. Žádný jednoduchý model nezahrne všechny aspekty softwarového vývoje. Potřebujeme tedy několik modelů k vyjádření různých pohledů. Modely musí být důkladně propojeny, aby byly konzistentní a nevznikaly zbytečné redundance.

2.4.1.1 Výhody vizuálního modelování

- Zachovává strukturu a chování architektury a komponent.
- Ukazuje, zda jsou jednotlivé prvky kompatibilní.
- Skrývá nebo odkrývá detaily podle potřeby.
- Podporuje konzistenci mezi návrhem a implementací.

2.4.2 Znázornění architektury

Samotná architektura systému a použitý presentační styl pro její znázornění není to samé. Systémovou architekturu můžeme prezentovat více způsoby s různým stupněm podrobnosti.

Je nutné si uvědomit, že na systémové architektuře je zainteresováno více subjektů a každý z nich zajímá jiná úroveň podrobnosti. Přesto spolu musí umět komunikovat a společnými silami vytvořit návrh systému, který jsou schopni realizovat. Proto je vhodné architekturu prezentovat ve více pohledech (rozměrech či vrstvách).

2.4.2.1 Zainteresované osoby na architektuře

Projektový manažer	Slouží mu k organizování týmů a plánování vývoje
Systémový analytik	Používají k organizaci a členění požadavků, k porozumění technologickým omezením a rizikům
Návrháři	Používají architekturu k porozumění základním principům a k stanovení omezení jejich návrhu
Architekti	K rozhodnutí, zda architekturu vyvíjet nebo znovu použít
Subdodavatelé	K pochopení části, které vyvíjí oni
Koneční uživatelé a zákazníci	K vytvoření představy o tom, co kupují

2.4.3 Znázornění vrstev

V systémové architektuře můžeme uvažovat o několika pohledech na systém, a to podle účelu jednotlivých pohledů (například logická organizace systému, fyzická struktura komponent, funkční celky systému nebo rozdělení komponent dle technologických platform - operačním systémů).

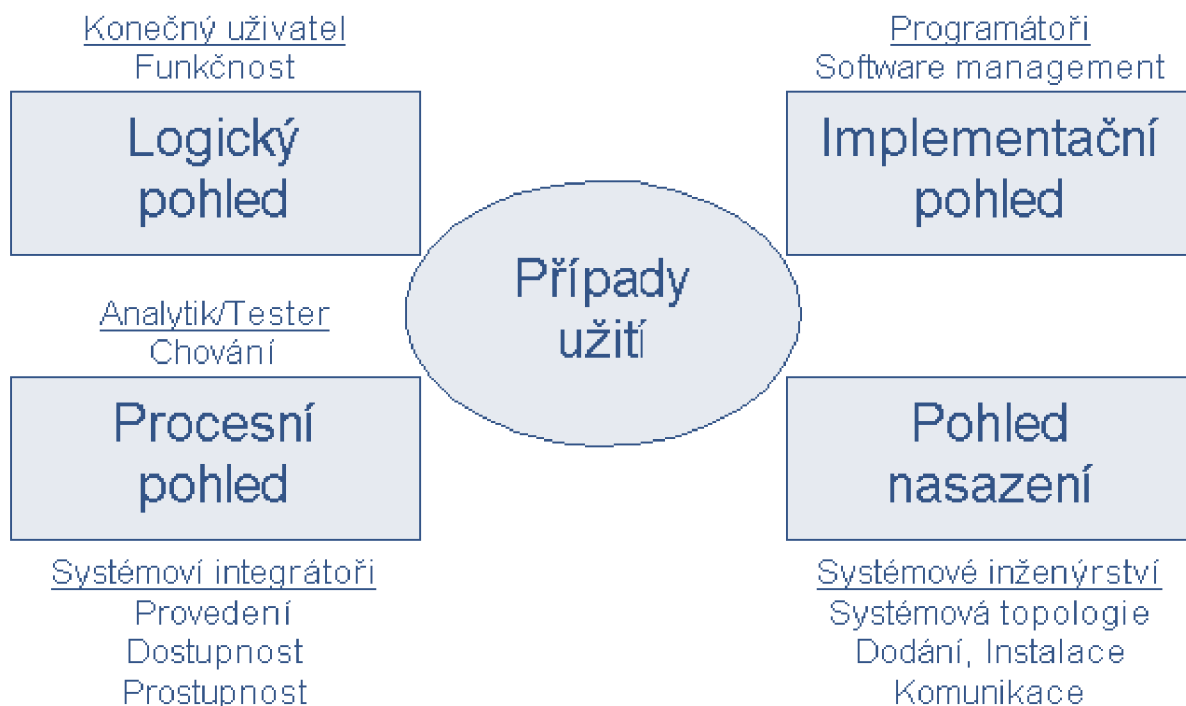
Takovým pohledům říkáme architektonické pohledy (vrstvy). Architektonický pohled je zjednodušený popis (abstrakce) systému z různých perspektiv a pozic, zahrnujících jednotlivé oblasti zájmu (nedůležité aspekty zanedbáme).

2.4.3.1 Pro každou vrstvu pohledu je nutné identifikovat

- Hledisko - oblast zájmu.
- Prvky, které budou zahrnuty do pohledu spolu se svými vzájemnými vztahy.
- Organizační principy.
- Prvky této vrstvy související s jinými vrstvami.
- Nejlepší proces tvorby této vrstvy.

2.4.4 Model 4+1

Jak ukazuje obrázek 2.4.1, Rational Unified Process navrhuje pět základních pohledů na architekturu.



obrázek 2.4.1: Model architektury 4+1

2.4.4.1 Významy jednotlivých pohledů

Logický pohled	Zobrazuje logickou strukturu systému z hlediska výsledné funkčnosti. Jde o abstraktní model návrhu, který identifikuje hlavní (věcné) packages, subsystemy a třídy.
Implementační pohled	Zobrazuje organizaci statických softwarových komponent (exe, dll, html). Znázorňuje rozčlenění systému do jednotlivých modulů. Strukturu definuje aplikační framework.
Procesní pohled	Znázorňuje procesy a jejich vzájemné působení. Řeší otázku konkurence a paralelismu procesů.
Pohled nasazení	Zaměřuje se na návaznost spustitelných programů a ostatních komponent, na topologii hardwarových a dalších softwarových komponent.
Případy užití	Obsahuje několik klíčových scénářů nebo Případů užití. Ty se používají k objevení a následnému návrhu architektury (v Počáteční a Elaboráční fázi). Později se jejich prostřednictvím ověřují různé pohledy na systém. Některé scénáře slouží v dokumentu softwarové architektury jako ilustrace pro znázornění fungování jiných pohledů.

2.5 Případy užití

Jak již bylo řečeno, modely pomáhají porozumět problému a nalézt jeho řešení. Je mnoho způsobů, jak namodelovat problém, vyjádřit požadavky a omezení vyvíjeného systému. Cílem je převést znázornění problému z tvaru pochopitelného pro koncového uživatele do podoby určené pro návrháře a tvůrce.

2.5.1 Typický postup návrhu

- Workshopy se zúčastněnými - definování základních aktérů a Případů užití.
- Zakreslení poznatků do modelu + Revize modelu - dopracování do finální podoby.
- Vytvoření popisů Případů užití + Revize popisů - vypracování popisů ke všem Případům užití.
- Návrh architektury, Design Model, Procesní model, Implementační model, Model nasazení.

2.5.2 Definice

Rational Unified Process definuje dva klíčové pojmy:

- **Případ užití** je posloupnost akcí, které poskytují konkrétním aktérům určitou hodnotu.
- **Aktér** je uživatel systému (nebo rozhraní externího systému), který se systémem pracuje (aktivuje jeho funkce).

Popis Případů užití říká, co se stane v systému, pokud je Případ užití aktivován. Funkčnost systému je definována řadou odlišných Případů užití, každý z nich reprezentuje specifický sled akcí.

2.5.3 Model Případů užití

Model Případů užití je model veškeré zamýšlené funkčnosti systému a jejího prostředí. Zahrnuje všechny Případy užití a osoby na procesu zúčastněné.

Při rozšiřování systému je užitečné kontrolovat, zda se v průběhu celého procesu používají stejné termíny. K tomu slouží dokument s názvem Slovník, nebo ještě lépe jednoduchý objektový model domény.

2.5.3.1 Vytvoření modelu Případů užití

- Identifikovat všechny aktéry, kteří k systému přistupují, a hlavní Případy užití.
- Vytvořit Slovník pojmů.
- Popsat Základní toky událostí, doplnit zbylé Případy užití.
- Doplnit popisy Případů užití do úrovně: Stručný popis, Základní tok událostí, Alternativní toky, Podmínky před spuštěním, Podmínky po spuštění a Rozšíření.

- Doplnit Slovník pojmů, strukturovat model Případů užití.

2.5.4 Použití Případů užití při vývoji

V Rational Unified Procesu jsou Příklady užití základem celého vývojového procesu. Model Případů užití je výstupem pracovního postupu Správa požadavků a mapuje požadavky uživatelů na funkčnost systému. Případy užití slouží jako komunikační jazyk mezi zákazníkem a vývojáři.

Při Analýze a návrhu propojují Případy užití požadavky a vlastní návrh systému (řešení požadavků). Z Případů užití se také vychází při návrhu konkrétních tříd a objektů. Odvozují se z nich Testovací případy nutné pro ověření funkčnosti systému. Lze z nich odhadnout rozsah a strukturu uživatelské dokumentace.

2.5.4.1 Případy užití zasahují do dalších aktivit

- Tvorba a ověření Modelu návrhu.
- Definice Testovacích případů a testovacích procedur.
- Plánování iterací.
- Tvorba uživatelské dokumentace.
- Nasazení systému.

2.5.5 Struktura modelu

Malý systém je možno vysvětlit pomocí několika Případů užití s dvěma či třemi aktéry. V případě většího systému je nutné definovat konstrukční a organizační pravidla. Jinak bude tvůrce zahlcen velkým množstvím Případů užití, které znesnadní plánování, odhad a znepráhlední požadavky. Konstrukční a organizační pravidla říkají jakým způsobem bude model strukturován a organizují jednotlivé posloupnosti tak, aby nedocházelo ke komplikaci projektové konzistence.

Prvním konceptem je vytváření packages, tzn. seskupování příbuzné funkčnosti (substémů). Často se narazí na Případy užití poskytující stejnou funkčnost v různých subsystémech. V těchto případech se využijí některé z metod dekompozice.

2.5.5.1 Jak strukturovat model?

- Sdružovat subsystémy do packages.
- Vytipování společných Případů užití a jejich zobecnění (znovu-použitelnost).
- Dekomponování úloh, skládání větších Případů užití z malých, dobře definovaných Případů užití.
- Používání rozšíření (oddělení obecné funkčnosti od specifických Případů užití).

3 VIZE

3.1 Předmluva

Účelem této části diplomové práce je shromážďovat, zanalyzovat a definovat klíčové potřeby a vlastnosti Informačního systému evidence znalostí zaměstnanců. Vize je zaměřena na ty vlastnosti systému, které jsou vyžadovány zainteresovanými osobami (osoby zúčastněné), koncovými uživateli a na skutečnost, proč tyto potřeby existují. Detaily, jak uspokojit tyto potřeby jsou popsány v případě použití (Kapitola 4) a doplňkových specifikacích (Kapitola 5).

Účel

Specifikuje a shromážďuje vize projektu v této části dokumentu.

Rozsah

Vize je aplikována na strukturu pracovníků IT společnosti a požadavků vytváření projektů firmou. Tento systém bude vývojové, softwarové společnosti umožňovat evidovat:

- Databázi znalosti zaměstnanců a úroveň těchto znalostí.
- Výsledky přezkoušení zaměstnanců, jejich zápis do databáze.

Systém umožní monitorovat možnosti a dovednosti pracovníků podstatných při tvorbě projektu.

Definice, zkratky a akronymy

Viz. Slovníček pojmů.

Odkazy

Použitá literatura [7].

3.2 Obchodní umístění

3.2.1 Obchodní příležitost (Business Opportunity)

Systém získává a plní databázi znalostmi zaměstnanců. Projekt umožní přehledněji, rychleji a efektivněji plánovat jednotlivé projekty oproti stávajícímu způsobu. Při tvorbě rozsáhlejšího projektu, je třeba pracovníky z jednotlivých oddělení dle potřeb zapojit na základě jejich znalostí spojených s prací na projektu. Systém umožní přesouvání zaměstnanců podílejících se na tvorbě projektů tam, kde je jejich působnost zrovna zapotřebí. Nový systém tento způsob výběru bude provádět na základě projektových požadavků. Informační systém bude navrhopat skupiny projektantů, potřebných pro konkrétní projekt. Databáze znalostí pracovníků firmy bude dostupná přes informační systém firmy.

3.2.2 Sdělení problému

Problém	Evidence znalostí pracovníků firmy není obsažena v žádném databázovém systému firmy.
Ovlivní	Zaměstnance pracující na projektu; přidělování projektů; přezkoušení znalostí.
Dopad problému	Pomalost a neefektivnost přesunu zaměstnanců na jiný projekt. Neznalost skutečného aktuálního stavu znalostí zaměstnance.
Úspěšné řešení	Zvýší rychlosti plánování projektů a přesuny zaměstnanců mezi nimi. Získání evidence znalosti zaměstnanců ve formě IS.

Tabulka 1

3.2.3 Produktové poziční sdělení

Pro	Zaměstnance společnosti vyvíjející softwarové produkty.
Co	Evidovat zaměstnance, pracovní schopnosti, přezkušování, přidělování zaměstnanců na projekty.
Systém evidence znalostí zaměstnanců	Softwarový produkt
Přínosy	Zaznamenání a evidence zaměstnanců podle jejich skutečně ověřených znalostí. Jednoduché nalezení alternativních náhradníků pro projekt.
Konkurenční alternativa	Stávající systém získávání a evidence znalostí.
Náš produkt	Zaznamená znalosti po přezkoušení do Informačního systému. Viditelný rozdíl v plánování. Rychlejší sestavení projektového týmu. Snadná reorganizace projektového týmu.

Tabulka 2

3.3 Zainterесované osoby a uživatelské popisy

Tato část popisuje uživatele systému evidence znalostí zaměstnanců ve společnosti, poskytuje profil zainterесovaných osob a uživatelů zapojených do projektu a klíčové problémy.

- zaměstnanec
- vedoucí
- školitel
- garant KH (know-how)
- projektant

3.3.1 Tržní demografie

Produkt je určen pro evidenci znalostí zaměstnanců firmy, její aktualizaci a je využit při přidělování zaměstnanců na jednotlivé projekty. Primárně bude systém vyhovovat požadavkům firmy Unicorn a.s.

3.3.2 Shrnutí zainterесovaných osob

Název	Popis	Odpovědnost
Zaměstnanec	Zaměstnanec	Pracovat na přiděleném projektu. Podávat žádosti o přezkoušení.
Vedoucí	Zaměstnanec	Zapíše stupně znalostí pracovníků. Má na starosti vedení přezkoušení pracovníků.
Školitel	Zaměstnanec, lokální školitel.	Přezkoušení zaměstnanců při pohovorech.
Garant KH	specialista na elementy znalostí.	Provádí sestavení systému elementů znalostí.
Projektant	Zaměstnanec	Komunikuje s klientem. Na základě rozhovoru s klientem vybere z IS potřebné znalosti pro práci na projektu.

Tabulka 3

3.3.3 Uživatelské shrnutí

Jméno	Popis	Zainteresaná osoba
Zaměstnanec	Pracovník pracující na projektu.	Zaměstnanec
Vedoucí	Eviduje znalosti zaměstnanců v databázi.	Vedoucí
Školitel	Testuje zaměstnance ze znalostí.	Školitel
Garant KH	Sestavuje a definuje elementy znalostí.	Garant KH
Projektant	Rozděluje práci zaměstnancům dle databáze znalostí.	Projektant

Tabulka 4

3.3.4 Uživatelské prostředí

IS bude umístěn na serveru firmy dostupného z Internetu. Uživatelé systému budou seznámeni s funkcí IS dle sepsané příručky dostupné v síti Internet. Připojení do IS nebude závislé na platformě operačního systému připojovacího se uživatele. Připojení bude vyžadovat internetový prohlížeč splňující standardy normy W3C. Zaměstnanci, budou moci pomoci IS přistupovat pod omezenými právy do tohoto systému, prohlížet, kontrolovat úroveň svých znalostí a zaregistrovat si přezkoušení.

3.3.5 Profily zainteresovaných osob

3.3.5.1 Zaměstnanec

Popis	Kvalifikovaný pracovník firmy. Pracuje na projektech firmy.
Typ	Uživatel základní
Odpovědnosti	Zajišťuje zpracování přidělené části na projektu. Aktivní komunikace s ostatními členy týmu.
Kritéria úspěchu	Prohlížení údajů o přiznaných znalostech s možností podání elektronické žádosti o přezkoušení na vyšší kvalifikaci v některé z vyžadovaných znalostí.
Zapojení	Do komunikace s ostatními zaměstnanci, projektantem, podávání žádostí o přezkoušení.

Tabulka 5

3.3.5.2 *Vedoucí*

Popis	Kvalifikovaný pracovník firmy. Přijímá žádosti o přezkoušení, řídí proces přezkoušení zaměstnanců. Aktualizuje data v IS dle výsledku zkoušky.
Typ	Uživatel pokročilý
Odpovědnosti	Aktualizace znalostí zaměstnanců, vyřizování žádostí zaměstnanců o přezkoušení při pohovorech. Zápis znalostí do IS. Zajištění účasti školitele k přezkoušení konkrétního pracovníka.
Kritéria úspěchu	Snadné prohlížení zaměstnanců s výpisem znalostí. Prohlížení údajů znalostí všech zaměstnanců. Editace znalostí, přezkoušení pro vyšší kvalifikaci.
Zapojení	Komunikativní spolupráce se zaměstnanci pracujícími na projektech. Aktivní účast na pohovorech a přezkušování.

Tabulka 6

3.3.5.3 *Školitel*

Popis	Přezkouvá zaměstnance ve své profesní oblasti.
Typ	Uživatel
Odpovědnosti	Dostavení se k přezkoušení zaměstnance a jeho přezkoušení.
Kritéria úspěchu	Přezkoušení všech zaměstnanců s žádostí o přezkoušení.
Zapojení	Přezkoušení zaměstnanců. Komunikace s vedoucím.

Tabulka 7

3.3.5.4 *Garant KH*

Popis	Připravuje elementy znalostí, pro zápis do IS.
Typ	Uživatel
Odpovědnosti	Odpovědný za důkladnou přípravu elementu systému elementu znalostí, které jsou vyžadovány po zaměstnancích při jejich přezkoušení.
Kritéria úspěchu	Snadná správa elementů znalostí.
Zapojení	Komunikace se školitelem.

Tabulka 8

3.3.5.5 Projektant

Popis	Zadávat požadované znalosti na projekt do IS, k vypracování skupiny potřebných projektantů. Řídí práci zaměstnanců na projektech.
Typ	Uživatel
Odpovědnosti	Zpracování požadavků na projekt, zhodnocení požadovaných znalostí na projekt, výběr projektových pracovníků.
Kritéria úspěchu	Nalezení a vybrání vhodného týmu pro tvorbu projektu.
Zapojení	Rozhovor se zákazníkem na požadavcích projektu. Updaty.

Tabulka 9

Uživatelské profily

Pokryty předchozí sekcí.

3.3.6 Klíčové požadavky, potřeby uživatele

Seznam problémů definovaných zainteresovanými osobami.

Potřeba	Priorita	Důvod znepokojení	Aktuální řešení	Navrhnutá řešení
Přehled úrovně znalostí zaměstnanců.	vysoká	Současný způsob neumožňuje kontrolu skutečného získání znalostí.	Současný systém celkový přehled neumožňuje.	Plnění databáze IS při pohovorech, přezkoušení.
Zadání, aktualizace, editace znalostí zaměstnanců.	vysoká	Neexistence systému umožňujícího tyto operace.	není	Aktualizace a editace, bude provádět osoba s těmito právy přístupem do IS z firemní sítě i ze sítě Internet.
Generování potřebných pracovníků pro jednotlivé projekty.	střední	Výběr zaměstnanců probíhá na základě neprověřených informací o znalostech zaměstnanců.	Současný systém způsobu přidělování projektů.	System umožní dle zadaných kritérií vybrat pro projekt seznam zaměstnanců vhodných pro jeho řešení.
Možnosti zobrazení historie přezkoušení a dat.	střední	Není potvrzena evidence o nabytých znalostech.	Výpis z současného systému.	Historie bude součástí IS.

Tabulka 10

Alternativy a konkurence

V současné době není znám konkurenční program.

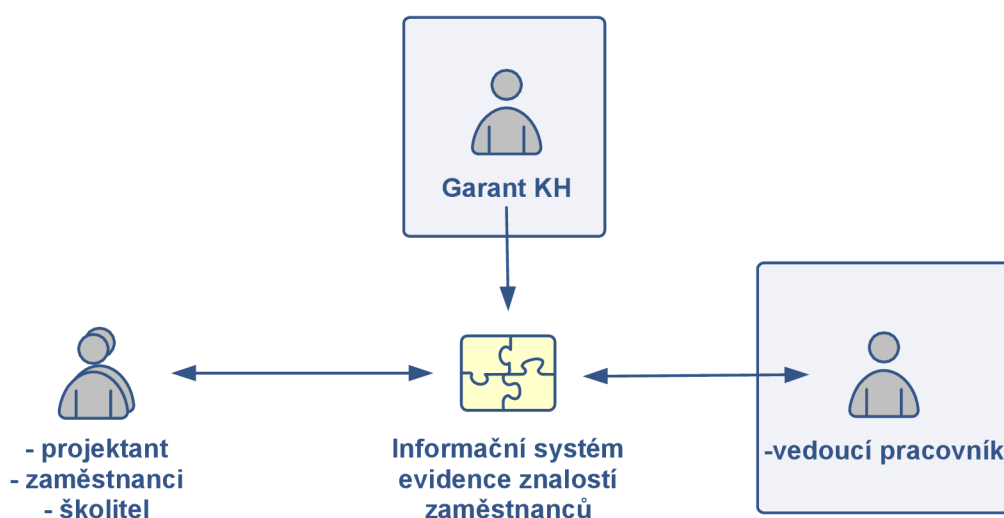
3.4 Přehled produktu

Tato kapitola poskytuje obecný pohled na systém evidence znalostí zaměstnanců.

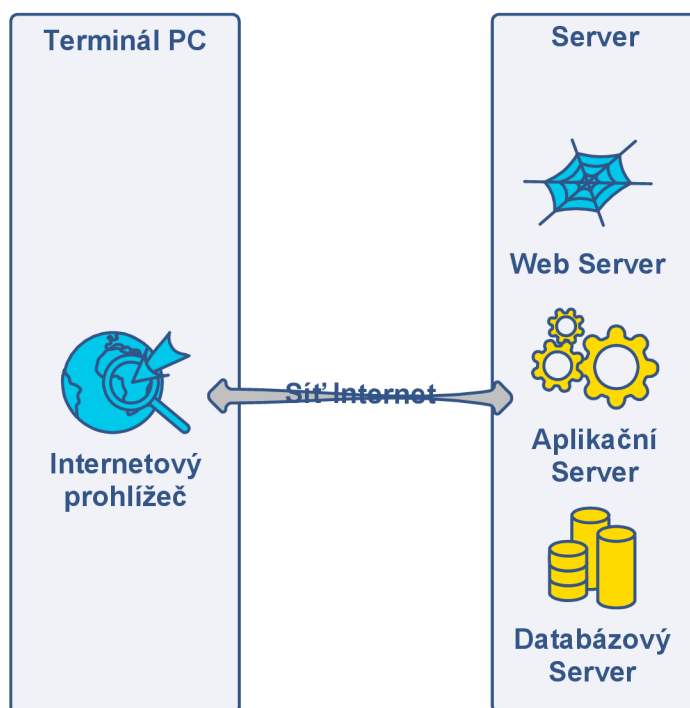
Způsob přezkoušení při pohovorech a zaznamenání znalostí. Rozhraní umožňuje přistupovat k databázi, vytváření tiskových sestav.

3.4.1 Produktový pohled

System bude umístěn na serveru, ke kterému budou z terminálů přistupovat uživatelé pomocí přihlášení. System bude dostupný přes internetový prohlížeč z lokální i světové sítě Internet. Server s aplikací bude umístěn na serveru firmy Unicorn a.s. s možností vnějšího přístupu, tj. bude vlastnit veřejnou IP adresu. Umístění na serveru může být také součástí již existujícího webového serveru, pokud to jeho výkonové parametry umožní. Tento systém umožní zaznamenávat a evidovat znalosti zaměstnanců, vytvářet skupiny pracovníků schopných se zapojit do problematiky projektu na patřičných úrovních a také umožní přehledný tisk těchto materiálů. Tuto situaci zachycuje obrázek 3.4.1. Klientské počítače budou moci přistupovat z různých síťových bodů kdekoliv to bude možné. Každý přístup do systému, bude vyžadovat autentizaci uživatele. Uživatel bude vyzván k vyplnění identifikačního jména a zadání přístupového hesla. Podle přihlášení a práv uživatele se zobrazí příslušné možnosti, které IS nabízí.



obrázek 3.4.1: Systém evidence znalostí zaměstnanců – logický pohled



obrázek 3.4.2: Systém evidence znalostí zaměstnanců – přehled

3.4.2 Shrnutí schopností

Tabulka definuje hlavní rysy systému evidence znalostí zaměstnanců společnosti v závislosti na výhodách a vlastnostech systému. Podrobnější popis vlastností je věnován v kapitole 3.5.

Výhody zákazníka	Podporované vlastnosti
Aktuální informace o znalostech pracovníka.	Okamžitý přístup do databáze a tím zobrazení aktuálních informací.
Aktualizace znalostí pracovníků z jakéhokoliv místa s přístupem do systému.	Systémem editovaná databáze s její okamžitou aktualizací.
Snadný přístup.	Umožní snadný přístup a zobrazení potřebných informací z míst k tomu potřebných.
Přístup z jakéhokoliv místa, počítače.	Aplikační server umožňuje připojení do IS z jakéhokoliv počítače nacházejícího se v síti Internet. Například domácí prostředí.
Bezpečnost přístupu	Pro přihlášení do systému bude třeba vyplnit autorizační údaje. Rozděluje uživatele do skupin podle práv přístupu.

Tabulka 11

Předpoklady a závislosti

Vývoj Systému evidence znalostí zaměstnanců musí být ukončen do konce listopadu 2007, aby mohl být včas nasazen do ostrého provozu od ledna 2008.

Náklady a cena

Finanční omezení pro vývoj systému, které se nesmí překročit činí 1 500 000 Kč.

Do nákladů vývoje systému je nutné započítat nákup hardwarových stanic, v případě že stávající stroje nebudou tomuto nároku vyhovovat. Dá se však předpokládat, že stávající servery budou výkonově dostačující a nezatíží jejich funkci na tolik, aby došlo k viditelnému zpomalení běžícího systému.

Pro využití klientských počítačů nebude zapotřebí dalších nákladů. Postačující k tomuto účelu budou stávající počítače s nainstalovaným webovým prohlížečem. Jako klientské stroje, mohou také sloužit osobní či přenosné domácí počítače zaměstnanců.

Licencování a instalace

Na produkt, nebudou uplatněny licence, jedná se o interní systém firmy. Instalace bude vyžadována pouze na serveru. Instalace na straně klienta nebude zapotřebí.

3.5 Rysy produktu

Kapitola definuje a popisuje vlastnosti systému evidence znalostí zaměstnanců. Vlastnosti zahrnují nejdůležitější schopnosti systému, uživatelům kteří systém využívají.

Přihlášení

Uživatelé systému se přihlásí s platným uživatelským jménem a heslem. Každý uživatel má vlastní přihlašovací jméno. Heslo je uživateli v první fázi přiděleno a má možnost si jej změnit po přihlášení v systému změnit.

Přístupová práva

Systém obsahuje různé úrovně správy databáze podle druhu přihlášení.

Vybraným osobám jsou přiřazeny vyšší práva a tím je zaručeno, že potřebné údaje budou moci měnit pouze osoby k tomu určené.

Aktualizace systémových nastavení

Jedná se o informace a aktualizace spojené se samotným chodem informačního systému. Možnost uživatele měnit heslo se promítne při nastavení každým uživatelem přímo do databázového systému a zůstane v něm pro daného uživatele uloženo.

Zadání a aktualizace znalostí

Aktualizaci zadává pověřený pracovník s právy k updatu do IS, který je uloží do databáze. Aktualizace se týká znalostí pracovníků pracujících na projektech. Tu provádí vedoucí pracovník při pohovorech a přezkoušení zaměstnance.

Informování pracovníků emailem

IS rozešle informaci o přidělení projektu všem vybraným pracovníkům emailem.

Déle informuje o zpracované aktualizaci znalostí daného zaměstnance.

Žádosti o přezkoušení

Zaměstnanci mohou požádat o přezkoušení svých znalostí na vyšší úroveň, nebo rozšíření výčtu svých znalostí pomocí IS.

Zadání kritérií projektu

Pro výběr týmu pro projekt, je nutné zadat požadované schopnosti. Ty zadává po analýze s klientem projektant. IS vytvoří a navrhne vedoucímu pracovníkovi tým pracovníků vhodných pro tento úkol. Projektant si z výběru vybere některou jeho variantu, nebo má možnost samostatného výběru explicitně.

Evidence znalostí a správa

Systém bude umožňovat vytvářet tiskové sestavy specifického výběru a také generování těchto sestav do XHTML kódu a PDF.

3.6 Omezení

Kromě předpokladů a závislostí vyjmenovaných v kapitole „*Předpoklady a závislosti*“ jsou omezení pro systém evidence znalostí zaměstnanců následující:

Systém by neměl vyžadovat žádný dodatečný hardware.

Systém nebude obsahovat veškeré informace o zaměstnancích, tyto data bude přejímat ze stávajícího databázového systému firmy, ke kterému bude mít umožněn přístup.

3.7 Kvalita systému, mezní hodnoty

Tato kapitola definuje rozmezí kvality systému v závislosti na výkonu, robustnosti, toleranci chyb, použitelnosti a podobných charakteristik pro Systém evidence znalostí zaměstnanců.

Dostupnost: Systém by měl být dostupný 24 hodin denně, 7 dní v týdnu.

Upotřebitelnost: Systém by měl být co nejjednodušší na ovládání a používání.

Použitelnost: Systém by měl obsahovat on-line uživatelskou nápovědu. Uživatelé si mohou tištěnou formu uživatelské příručky v případě potřeby vytisknout.

Udržovatelnost: Systém by měl být navržen tak, aby jeho údržba byla co nejjednodušší. Všechna uživatelská nastavení by měla být uložena v konfiguračních souborech a měly by být měnitelné bez rekonpilace systému.

3.8 Precedence a priorita

V této kapitole jsou uvedeny relativní priority jednotlivých navrhovaných vlastností systému. Funkčnosti systému definované v kapitole Víze by měly být zahrnuty v prvních dvou verzích systému. Všechny rysy systému, které jsou potřebné pro editaci a prohlížení vztahů mezi zaměstnanci a znalostmi, budou implementovány v první verzi.

Jednotlivé vlastnosti systému jsou porovnány z hlediska důležitosti, přínosu uživateli, obtížnosti a rizika. Výsledkem tohoto zhodnocení je rozhodnutí, které vlastnosti systému budou implementovány v jednotlivých verzích systému.

Očekává se, že systém evidence znalostí zaměstnanců bude uvolněn do provozu ve společnosti během 2-4 vydání verze systému.

Verze 1 musí obsahovat následující funkcionalitu:

- Přihlášení.
- Přístupová práva.
- Prohlížení znalostí zaměstnance.
- Zadávání, aktualizace a prohlížení znalostí o zaměstnancích.
- Evidence elementů znalostí.

Verze 2 by měla obsahovat:

- Žádost o přidělení znalosti.
- Možnost uživatele měnit přístupové heslo.

Verze 3 by měla obsahovat:

- Vytváření tiskových sestav.
- Výpis požadovaných elementů

Kompletní přechod na nový systém se očekává s ukončením třetí verze.

3.9 Jiné požadavky produktu

Seznam požadavků na produkt obsahuje seznam použitelných standardů, hardwarových požadavků nebo požadavků na výkonnost a požadavky na prostředí.

Použitelné standardy

Klientské PC by mělo obsahovat internetový prohlížeč splňující standart validitu normy W3C nebo Microsoft Internet Explorer ver. 5.5 a vyšší.

Systémové požadavky

System by měl být schopný importovat základní data do databáze znalostí ze stávající interní databáze zaměstnanců, nebo k nim přistupovat.

Komunikace s webovou službou bude probíhat pomocí protokolu SOAP.

Klientská aplikace bude vyžadovat internetový prohlížeč.

Výkonové požadavky

System by měl podporovat až 100 současně pracujících uživatelů vůči centrální databázi a lokálnímu serveru v jakýkoliv čas.

Doba odezvy systému by neměla být delší než 5 sekund.

Požadavky na prostředí

Umístění v serverové místnosti s ostatními počítači obsluhující síťový provoz firmy.

3.10 Požadavky dokumentace

Tato kapitola popisuje požadavky kladené na dokumentaci k Systému evidence znalostí zaměstnanců.

Uživatelská příručka

Uživatelská příručka by měla popisovat používání systému z pohledu pracovníků firmy pracujících na projektu v týmu vytvořeného dle jejich znalostí.

Uživatelská příručka by měla obsahovat:

- Minimální požadavky na systém.
- Využití systému na klientské stanici.
- Přihlášení.
- Odhlášení.
- Všechny funkčnosti systému.
- Informace o zákaznické podpoře.

Uživatelská příručka by měla obsahovat zhruba 50 až 100 stran. Stránka uživatelské příručky by měla být formátu A5. Uživatelská příručka by měla být dostupná v on-line či v tištěné verzi.

On-line nápověda

On-line nápověda by měla být k dispozici uživateli v každé funkční části systému. Každá kapitola v uživatelské příručce by měla být k dispozici také skrze on-line nápovědu.

Průvodce instalace, konfigurací, soubor „read me“

Instalační příručka určená pro server by měla obsahovat:

- Minimální systémové požadavky.
- Instalační instrukce.
- Konfiguraci Systému evidence znalostí.
- Informace o zákaznické podpoře.
- Informace o možnosti objednání upgradů.

Soubor „Read Me“ by měl obsahovat informace o postupu instalace. Tento soubor by měl být dostupný na instalačním disku a měl by být umístěn také v adresáři, ve kterém bude nainstalován Systém řízení know-how.

Tento soubor by měl obsahovat:

- Nové vlastnosti systému v dané verzi.
- Seznam známých chyb a jejich oprav.
- Provizorní řešení neočekávaných chyb systému.

Označení a balení

Na uživatelské dokumentaci bude umístěno logo společnosti Unicorn Systems.

Úvodní verze systému budou určeny pouze pro společnost Unicorn.

K systému nebudou vyhotoveny marketingové a propagační materiály.

4 DIAGRAMY POUŽITÍ (USE-CASE)

4.1 Notace Use-Case diagramu

Use Case diagram vyjadřuje vztahy mezi Aktéry mimo systém a Use Case uvnitř systému, zachycuje funkce, které systém poskytuje a definuje jejich využití. Výhodou případů použití je, že je popsán přirozeným jazykem a tím dobře srozumitelný. Detailní popis komunikace mezi aktory a systémem není v tomto diagramu obsažen. V provedení Use Case existují vztahy podporované UML standardem, které popisují grafický zápis těchto vztahů (include, extended, generalization a association). Asociační vztah mezi aktérem a Use Case je často nazýván komunikační asociace (communicate). K informacím o řízení podnikových znalostí bylo použito literatury [4].

Diagram užití spadá pod diagramy chování systému, obsahuje tyto základní prvky:

Use Case (případ užití)



Use Case

(from Use Case View)

Definují řady aktivit, včetně různých variant, které může systém vykonávat ve vztahu k Aktérovi (actor).

Element modelu zachycuje funkčnost systému, je jednoznačně identifikován svým stručným a výstižným názvem.

zobrazení: elipsa

Actor (aktér, účastník)

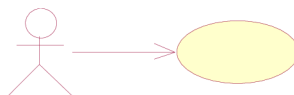


Actor

Akteři jsou lidé nebo externí (neživé) systémy, na které bude systém působit a používat jej, až bude v provozu. Actor je považován za cokoliv s externím pohledem na systém.

zobrazení: symbol osoby

Association



Actor

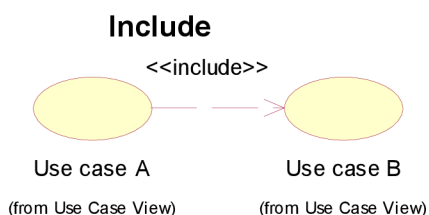
Use Case

(from Use Case View)

Vztahy mezi *aktérem a případem užití* se nazývají **asociace**.

zobrazení: plná čára mezi aktérem a use case s nepovinným označením «*communicate*». Případná orientace je vyjádřena šipkou ve směru interakce.

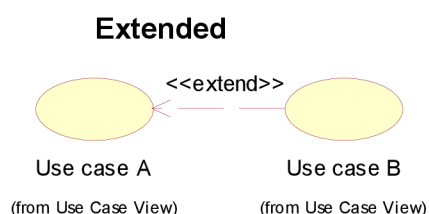
Relationship vztahy mezi případy užití (závislosti)



Use Case A zahrnuje Use Case B

Tzn. pokud se nějaká funkcionálna v různých částech opakuje, vyjme se a vytvoří se nový Use Case, pouze s touto dílčí funkcionalitou. Na něj pak odkazují jiné Use Case. Z vnějšího pohledu se jedná pouze o jeden případ užití.

zobrazení: orientovaná čára vyjádřena šipkou mezi dvěma případy užití k zahrnovanému případu užití s typem «include» v lomených závorkách.



Use Case B rozšiřuje Use Case A o jinou funkčnost

Používá ke znázornění volitelného chování; chování, které se projeví pouze za určitých podmínek, nebo několika odlišných toků, které mohou být spuštěny výběrem aktéra. Na rozdíl od konstrukce s vazbou include musí být extended případ užití smysluplný a proveditelný samostatně.

zobrazení: orientovaná čára vyjádřena šipkou mezi dvěma případy užití k rozšiřovanému případu užití s typem «extended» v lomených závorkách.

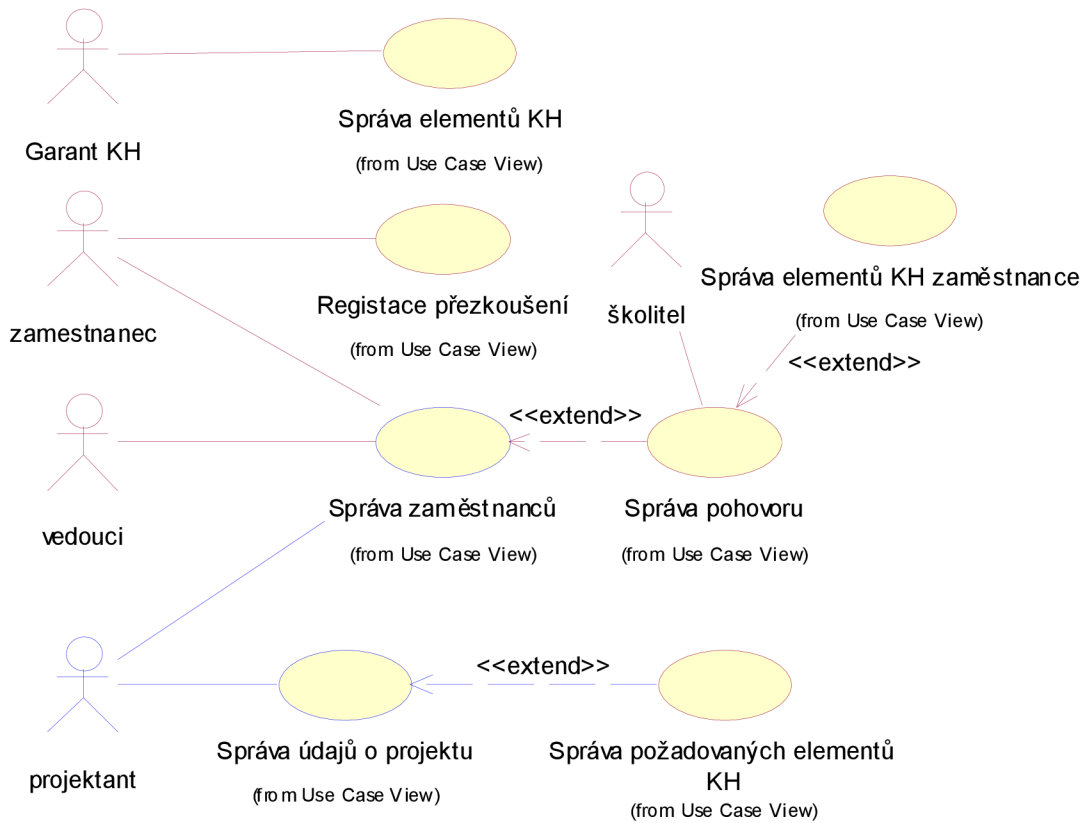
Informace o notaci zápisu byly čerpány z literatury uvedené v [3] a [5].

4.2 Předmluva

Tato kapitola popisuje množinu případů užití ve vztazích vyjadřujících funkčnost systému. Obsahuje případy a jejich jednotlivé toky událostí. V toku událostí je popsáno, co systém má dělat, nestará se o to jak. Tzn. není sepsán v jazyce implementačním.

Hlavním případy jsou:

- Správa zaměstnanců
- Správa pohovoru
- Správa elementů KH zaměstnance
- Žádost o přidělení znalosti
- Správa elementů KH
- Správa údajů o projektu
- Správa požadovaných elementů KH

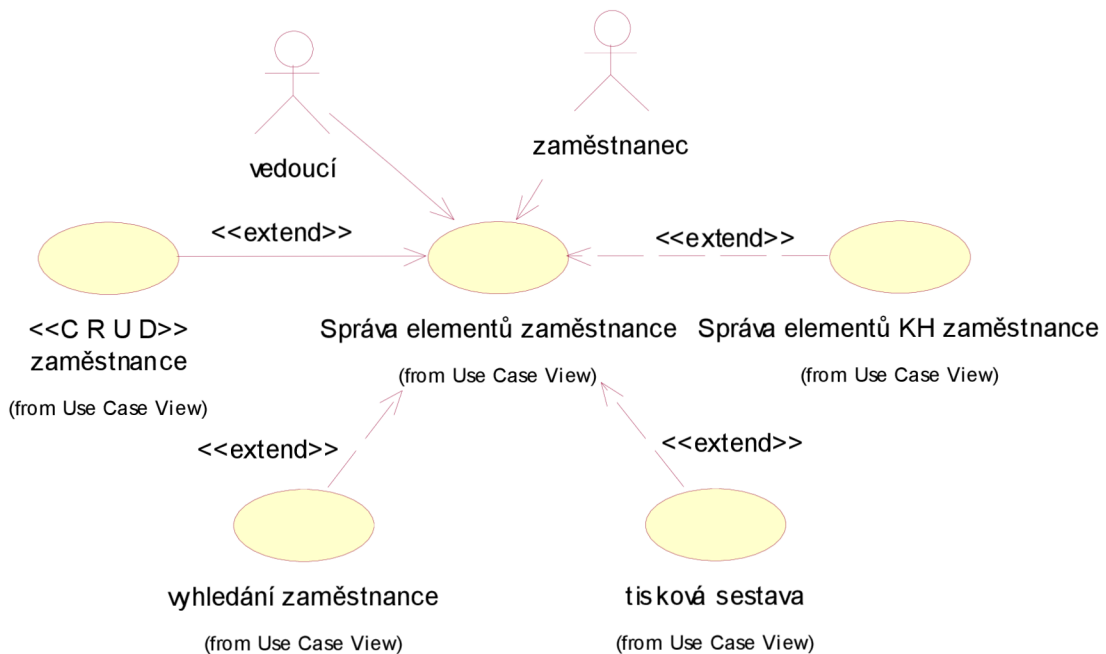


obrázek 4.2.1: Případy užití – hlavní

Modrou barvou jsou odlišeny externí zásahy do systému. Které se systémem úzce spolupracují, přitom však do systému nespádají.

Následují kapitoly s podrobnějším popisem případů užití. Případy užití obsahují zavedenou metodiku <<C-R-U-D>> jež symbolizuje 4 základní Use Case, kterými jsou přidání, zobrazení(čtení), aktualizace a smazání (Create, Read, Update, Delete).

4.3 Správa údajů o zaměstnancích



obrázek 4.3.1: Příklad užití – Správa údajů o zaměstnancích

Popis

Příklad užití umožňuje spravovat *vedoucímu* pracovníkovi údaje o zaměstnanci a správu elementů tj. údaje přidávat, mazat, upravovat, zobrazovat i vyhledávat. *Zaměstnanci* umožňuje údaje o zaměstnancích vyhledávat a prohlížet.

Speciální požadavky

Budou specifikovány během další iterace.

Předpoklady

Přihlášení Aktéra do systému.

Ukončovací podmínky

Návrat k úvodní obrazovce.

Tok událostí

Zde je zahrnutý tok událostí pro správu zaměstnanců. Úroveň je zobrazena po výběru správy zaměstnanců v hlavním případě užití.

Základní tok událostí

Přidání zaměstnance

1. Aktér zvolí *přidat zaměstnance*.
2. Zobrazí se formulář pro přidání zaměstnance.
3. Aktér zadá informace o zaměstnanci.
4. Po potvrzení zadaných informací, systém provede kontrolu zadaných dat a v případě chybného vyplnění zobrazí před-vyplněný formulář znovu s označením špatně vyplněných údajů, nebo údajů které jsou nevyplněné, ale jsou povinné. Po správném vyplnění formuláře se provede kontrola existence zaměstnance v systému. Systém přidá nového zaměstnance do systému a přidělí mu jednoznačné uživatelské číslo ID. Operaci lze také v průběhu zadávání údajů přerušit, v tom případě případ užití končí.
5. V případě přidání více zaměstnanců do systému se opakují kroky 1-4.

údaje o zaměstnanci obsahují tyto údaje:

- jméno, příjmení, rodné číslo, tituly před a za jménem, adresa bydliště (ulice, číslo popisné, obec, část obce, PSČ, stát), emailová adresa, telefonní číslo, ID projektů na kterých pracuje.

povinné údaje obsahují tyto údaje:

- *jméno, příjmení, rodné číslo, číslo popisné, obec, stát.*

Alternativní toky

Zobrazení zaměstnance

1. Systém zobrazí seznam zaměstnanců se základními údaji.
2. Pro zobrazení podrobností tj. kompletního výpisu položek zaměstnance Aktér potvrdí výběrem konkrétního zaměstnance a výběrem možnosti pro zobrazení podrobností.
3. Systém načte všechny informace o zaměstnanci a vypíše je na obrazovku.

základními údaji při zobrazení zaměstnanců jsou údaje:

- ID, příjmení, jméno, rodné číslo, město

1. Existence zaměstnance

V případě nalezení existujícího zaměstnance se stejným rodným číslem systém zobrazí upozornění, že uživatel s tímto rodným číslem již existuje. Nabídne Aktérovi změnu údajů právě přidávaného zaměstnance, nebo editaci zaměstnance se kterým se shoduje rodné číslo (v případě že se jedná o osobu s jiným jménem a příjmením) nebo zrušení prováděné operace, v tom případě případ užití končí.

2. Editace zaměstnance

2. Systém zobrazí seznam zaměstnanců se základními údaji.
3. Aktér vybere a potvrdí řádek se zaměstnancem, kterého bude chtít editovat.
4. Systém načte a zobrazí veškeré dostupné údaje o zaměstnanci ve formuláři pro možnou editaci.
5. Aktér upraví zobrazené změnitelné údaje, které je třeba editovat.
6. Po potvrzení uložení údajů, systém provede kontrolu zadaných dat a v případě chybného vyplnění zobrazí před-vyplněný formulář znovu s označením špatně vyplněných údajů, nebo údajů které jsou nevyplněné, ale jsou povinné. Po správném vyplnění formuláře systém aktualizuje editovaného zaměstnance do systému a přidělí mu jednoznačné uživatelské číslo ID.
7. V případě editace změn u více zaměstnanců se opakují kroky 2-6.

základními údaji při zobrazení zaměstnanců jsou údaje:

- ID, příjmení, jméno, rodné číslo, město, ID projektů na kterých pracuje

zobrazené změnitelné údaje jsou:

- jméno, příjmení, rodné číslo, tituly před a za jménem, adresa bydliště (ulice, číslo popisné, obec, část obce, PSČ, stát), emailová adresa, telefonní číslo, element know-how a jeho stupeň znalosti (více elementů), žádost o přezkoušení, pohovor kterého se zaměstnanec účastnil (více pohovorů).

Povinnými údaji jsou:

- jméno, příjmení, rodné číslo, číslo popisné, obec, stát.

3. Smazání zaměstnance

1. Systém zobrazí seznam zaměstnanců se základními údaji.
2. Aktér vybere řádek se zaměstnancem, kterého chce smazat a potvrdí tento výběr.
3. Zobrazí se systémem generované potvrzení smazání zaměstnance.
4. Aktér potvrdí toto smazání zaměstnance.
5. Systém zkontroluje, zda zaměstnanec nemá podanou žádost o přezkoušení nebo nepracuje na nějakém projektu. V případě, že ano vypíše upozornění na tuto skutečnost a navrátí Aktéra na seznam zaměstnanců. V opačném případě zaměstnance smaže ze systému.
6. V případě mazání více zaměstnanců ze systému se opakují kroky 2-5.

zobrazení základních údajů:

- ID, příjmení, jméno, rodné číslo, město

4. Vyhledávání zaměstnance

1. Aktér zvolí vyhledat zaměstnance.
2. Zobrazí se formulář pro vyhledání zaměstnance.
3. Aktér zadá informace o zaměstnanci podle kterých chce vyhledávat.
4. Po potvrzení vyhledávání, systém provede vyhledání zaměstnanců v systému. Pro vyhledání je potřebný alespoň jeden zadaný údaj. Vyhledání zaměstnanci se vypíše na obrazovku společně s formulářem pro vyhledávání a počtem nalezených údajů. Po té je možné vybrat konkrétního zaměstnance, nebo upřesnit vyhledávání a nechat vyhledat znovu. V případě výpisu většího množství zaměstnanců je výpis výběr rozdělen na více stránek.
5. V případě vyhledání více zaměstnanců v systému se opakují kroky 1-4.

Údaje, které lze použít k vyhledávání jsou:

- ID, jméno, příjmení, rodné číslo, tituly před a za jménem, adresa bydliště (ulice, číslo popisné, obec, část obce, PSČ, stát), emailová adresa, telefonní číslo, ID projektů na kterých pracuje.

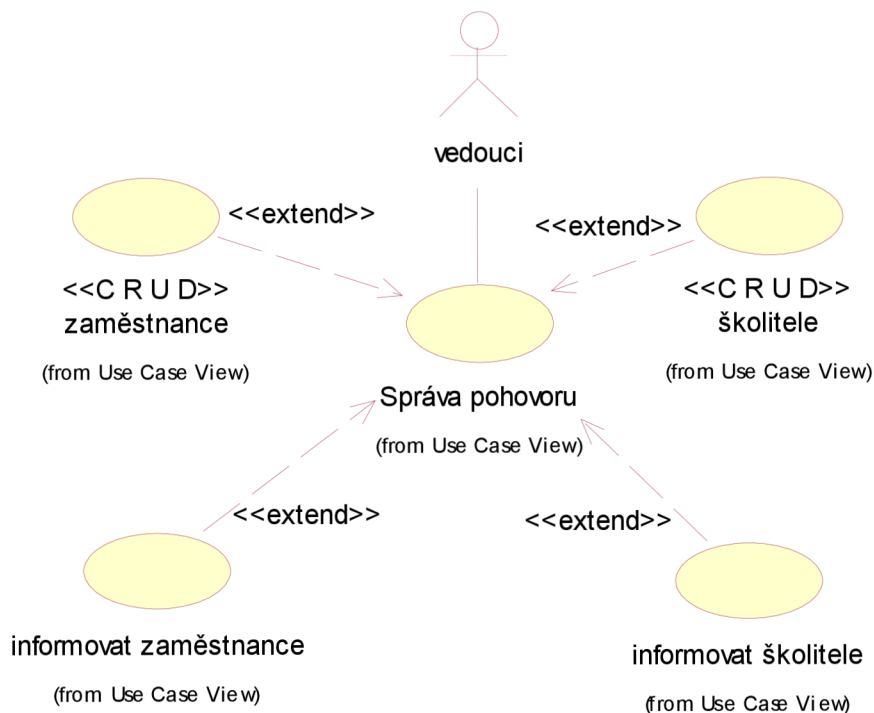
5. Správa elementů zaměstnance

funkčnost případu užití je popsána v kapitole 4.5

Rozšíření

- Přihlášení uživatele
- Odhlášení uživatele
- Vytvoření tiskové sestavy
- Správa zaměstnance

4.4 Správa pohovoru



obrázek 4.4.1: Správa pohovoru

Popis

Případ užití umožňuje spravovat *vedoucímu* pracovníkovi pohovory, kterých se účastní zaměstnanci. Součástí pohovorů je přezkoušení spolu se školiteli, kde se přezkouvají zaměstnanci.

Speciální požadavky

Nastavení SMTP serveru pro odchod emailových zpráv.

Následující budou specifikovány během další iterace.

Předpoklady

Přihlášení Aktéra do systému.

Aktér se nachází v případě použití „*Správa zaměstnanců*“

Ukončovací podmínky

Návrat k obrazovce správy zaměstnanců.

Tok událostí

Zde je zahrnutý tok událostí pro správu pohovoru se zaměstnancem. Úroveň je zobrazena po výběru správy pohovoru ve správě údajů o zaměstnancích.

Alternativní toky

1. Odebrání zaměstnance z pohovoru

1. Systém zobrazí seznam zaměstnanců přihlášených na pohovor se základními údaji.
2. Aktér vybere řádek se zaměstnancem, kterého chce odebrat z pohovoru a potvrdí výběr.
3. Zobrazí se systémem generované potvrzení odebrání zaměstnance z pohovoru.
4. Aktér potvrdí odebrání zaměstnance z pohovoru.
5. Systém zkontroluje, zda zaměstnanec nemá podanou žádost o přezkoušení. V případě, že ano vypíše hlášku o tomto faktu a navrátí Aktéra do správy pohovorů. V opačném případě pohovor zaměstnance odebere ze systému. O tomto kroku informuje zaměstnance emailem.
6. V případě rušení více pohovorů zaměstnancům ze systému se opakují kroky 2-5.

základní údaje pro zobrazení seznamu zaměstnanců přihlášených na pohovor jsou:

- ID, příjmení, jméno, rodné číslo, místo konání pohovoru, datum a čas pohovoru

2. Odebrání školitele z přezkoušení

1. Systém zobrazí seznam školitelů přihlášených na pohovor pro přezkoušení zaměstnanců se základními údaji.
2. Aktér vybere řádek se školitelem, kterého chce odebrat z pohovoru a potvrdí tento výběr.
3. Systém zkontroluje, zda lze za školitele přiřadit náhradu pro přezkoušení. V případě, že lze, zobrazí se systémem generované potvrzení odebrání školitele z pohovoru. V opačném případě, když nelze nahradit školitele, zobrazí se systémem generované potvrzení odebrání školitele z pohovoru s informací, že za školitele nelze přiřadit ekvivalentní náhradu.
4. Aktér potvrdí odebrání školitele z přezkoušení.
5. O tomto kroku systém informuje školitele emailem.
6. V případě rušení více přezkoušení školitelům ze systému se opakují kroky 2-5.

základní údaje pro zobrazení seznamu školitelů přihlášených na pohovor jsou:

- ID, příjmení, jméno, rodné číslo, expertní znalost ze které školitel přezkouvá, místo konání pohovoru, datum a čas pohovoru

3. Informování zaměstnance a školitel o pohovoru

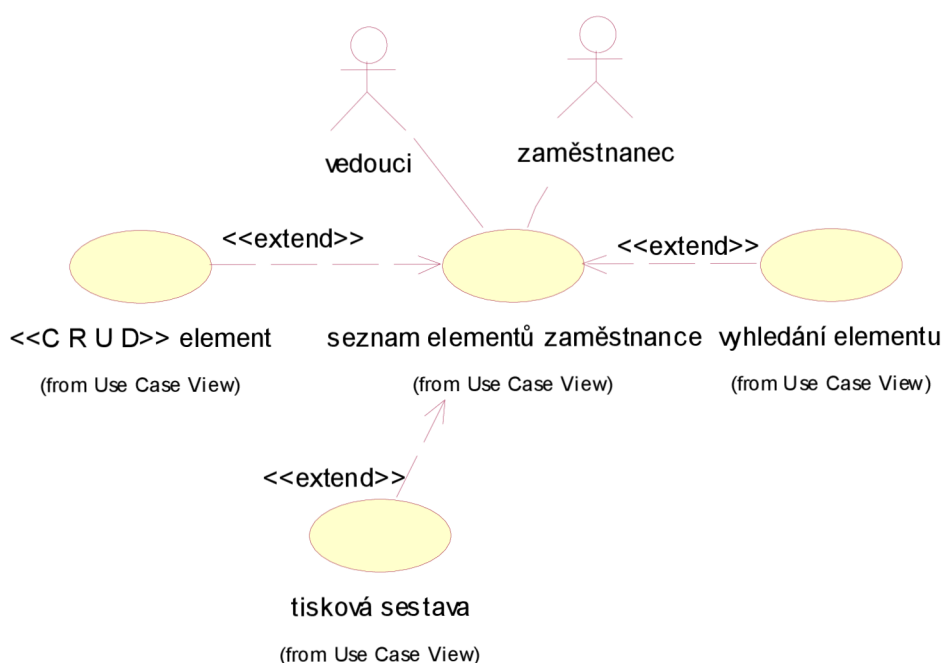
Informování zaměstnance nebo školitele proběhne automaticky generováním systému. Tato skutečnost se provede v případě přidání, smazání nebo změny pohovoru pro zaměstnance

nebo školitele a to odesláním emailu s potřebnými informacemi. Po odeslání údajů s informací o zrušení, přesunu s datem, časem a místem pohovoru případ užití končí.

Rozšíření

- Správa pohovoru
- Přiřazení uživatele
- Úprava pohovoru
- Vytvoření tiskové sestavy

4.5 Správa elementů KH zaměstnanec



obrázek 4.5.1: Správa elementů KH zaměstnanec

Popis

Případ užití umožňuje spravovat *vedoucímu* pracovníkovi údaje o elementech zaměstnanec. Tj. umožňuje elementy k zaměstnancům přidávat, mazat, upravovat, zobrazovat i vyhledávat. *Zaměstnanec* má umožněno údaje o elementech vyhledávat a prohlížet.

Speciální požadavky

Budou specifikovány během další iterace.

Předpoklady

Přihlášení Aktéra do systému.

Aktér se nachází v případě použití „*Správa pohovoru*“

Ukončovací podmínky

Návrat do správy údajů o zaměstnancích.

Tok událostí

Zde je zahrnutý tok událostí pro správu elementů zaměstnance. Úroveň je zobrazena po výběru správy elementů zaměstnance ve správě údajů o zaměstnancích.

Základní tok událostí

Přidání elementu KH

1. Aktér zvolí přidání elementu.
2. Zobrazí se formulář pro přidání elementu.
3. Aktér vybere ze seznamu elementů konkrétní element, který chce přidat.
4. V položce stupeň znalosti vybere aktér stupeň, kategorii znalosti pro každého jednotlivého zaměstnance.
5. Po potvrzení výběru znalosti, systém provede kontrolu existence elementu a uloží znalost s jeho stupněm do systému. Operaci lze také v průběhu výběru přerušit, v tomto případě případ užití končí.
6. V případě přidání více elementů zaměstnance do systému se zopakují kroky 1-5.

kategorie rozčlenění znalostí probíhá:

- dle tří vědomostních kategorií (junior, senior, expert) tedy hodnoty 1,2,3.

Alternativní toky

Zobrazení elementu KH

1. Systém zobrazí seznam elementů zaměstnance se základními údaji.
2. Zobrazení podrobností o elementu Aktér provede výběrem konkrétního elementu a potvrzením pro zobrazení podrobností.
3. Dojde k zobrazení podrobností.

zobrazení seznamu elementů se základními údaji:

- Element KH, stupeň znalosti (1-3)

při zobrazení podrobností se zobrazí tyto údaje

- Systém načte a zobrazí popis elementu pro každý jeho stupeň společně se zkratkou elementu.

1. Existence elementu KH

V případě akce *Přidání elementu KH* kdy systém nalezne existující element KH zaměstnance, systém zobrazí chybové hlášení. Pokud je právě zapisovaný stupeň

znalostí menší nebo roven stupni znalostí již v systému obsaženému elementu systém zobrazí hlášení o nekorektnosti operace a navrátí aktéra do menu přidání elementu s před-vyplněným formulářem znalosti. V případě, že právě zapisovaný stupeň je vyšší, systém provede změnu stupně znalosti existujícího elementu na přidávaný stupeň.

2. Editace elementu KH

1. Systém zobrazí seznam elementů daného zaměstnance.
2. Aktér vybere řádek s elementem zaměstnance, který bude chtít editovat potvrdí operaci.
3. Systém načte a zobrazí údaje o elementech zaměstnance ve formuláři pro možnou editaci.
4. Aktér upraví údaje, které je třeba editovat.
5. Po potvrzení uložení editace, systém aktualizuje element KH daného uchazeče.
6. V případě editace změn elementů u více zaměstnanců se opakují kroky 2-5.

zobrazení seznamu elementů daného zaměstnance obsahuje:

- Zkratku elementu KH, název elementu KH, stupeň znalosti

zobrazené údaje které se budou editovat jsou:

- Element KH, stupeň znalosti (1-3)

3. Smazání elementu KH

1. Systém zobrazí seznam elementů daného zaměstnance.
2. Aktér vybere řádek s elementem zaměstnance, kterého chce smazat a potvrdí jeho smazání.
3. Zobrazí se systémem generované potvrzení smazání elementu KH.
4. Aktér se utvrdí že se jedná o element, který chce smazat a potvrdí smazání.
5. Systém smaže element KH.
6. V případě mazání více elementů ze systému se opakují kroky 2-5.

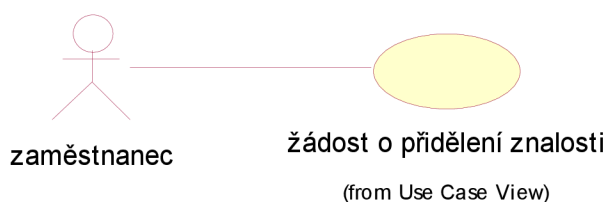
seznam elementů daného zaměstnance obsahuje:

- Zkratku elementu KH, název elementu KH, stupeň znalosti

Rozšíření

- Vytvoření tiskové sestavy
- Správa požadovaných elementů

4.6 Žádost o přidělení znalosti



obrázek 4.6.1: Žádost o přidělení znalosti

Popis

Případ užití umožňuje zaměstnanci podat žádost o přidělení znalosti. Na pohovoru bude otestován podle své žádosti. Znalost mu bude přidělena po jeho úspěšném přezkoušení na následujícím pohovoru. *Zaměstnanec* má možnost zvolenou znalost s popisem vyhledat a prohlédnout jestli se skutečně jedná o tu, kterou chce přiznat.

Speciální požadavky

Budou specifikovány během další iterace.

Předpoklady

Přihlášení zaměstnance do systému.

Ukončovací podmínky

Návrat na úvodní obrazovku zaměstnance.

Tok událostí

Je zde zahrnut tok událostí pro registraci přezkoušení zaměstnance. Úroveň je zobrazena po výběru „žádost o přidělení znalosti“ zaměstnance v hlavní obrazovce zobrazení zaměstnance.

Základní tok událostí

1. Žádost o přidělení znalosti

1. Systém zobrazí seznam všech dostupných elementů.
2. Pokud uživatel má již některé znalosti přiděleny, jsou zobrazeny ve výpisu na začátku seznamu rozdílnou barvou s patřičným stupněm znalosti. Pro výběr stupně znalosti jsou zobrazeny jeho možnosti.
3. Pokud chce zaměstnanec navýšit úroveň, některé z již získaných znalostí, je mu to umožněno ze stejného seznamu. Zobrazené úrovně pro již získané znalosti zobrazují výběr z úrovní vyšších.
4. Zaměstnanec vybere znalost a zvolí požadovanou úroveň, kterou by chtěl uznat a potvrdí tento výběr.
5. Pro registraci více znalostí se opakují kroky 3-4.

6. Po přidání všech žádaných znalostí do výběru, zaměstnanec žádost o přidělení znalosti potvrdí.
7. V dialogovém okně se vypíše souhrn změn, které zaměstnanec učinil a systém požádá potvrzení *žádosti o přidělení znalosti*. Operaci *podání žádosti* lze přerušit, v tom případě případ užití končí.
8. Po potvrzení registrace se data aktualizují v systému.

zobrazení seznamu všech dostupných elementů obsahuje:

- Zkratku elementu KH, název elementu KH, stupeň znalosti

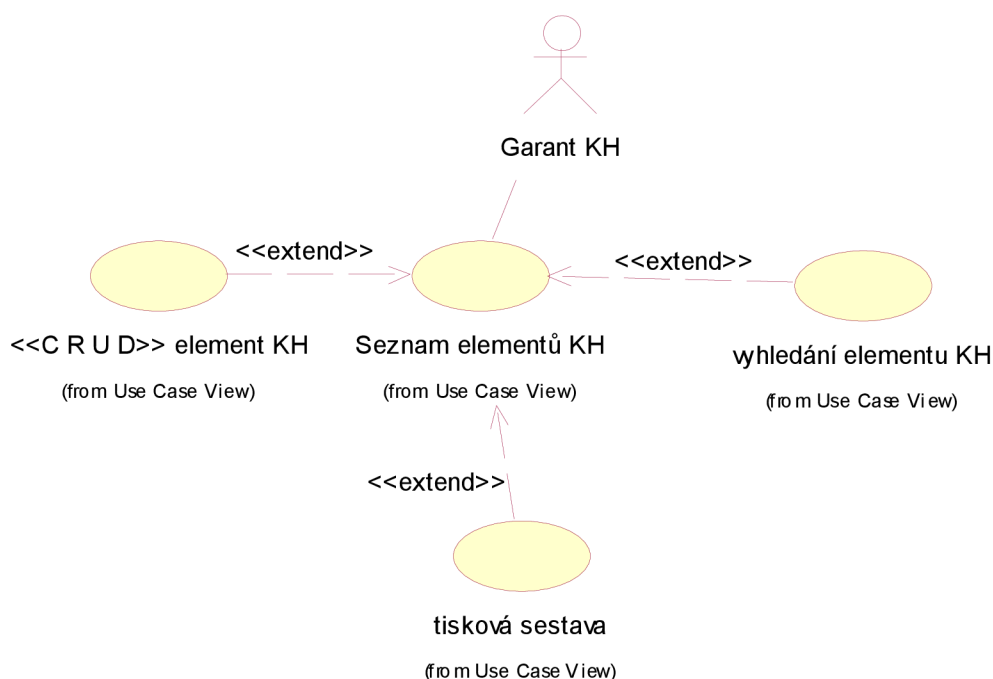
zobrazení možností stupně znalosti nabízí:

- čtyři možnosti, žádná (prázdné pole), hodnot 1-3 (junior, senior, expert).

Rozšíření

- Vytvoření tiskové sestavy

4.7 Správa elementů KH



obrázek 4.7.1: Správa elementů KH

Popis

Případ užití umožňuje spravovat elementy KH. Umožňuje přidávat, editovat i mazat elementy, měnit jejich popis i označení. V případě většího počtu elementů v systému lze použít vyhledávání. Název elementu obsahuje celý název v délce jedné věty. Zkratka elementu, je jedinečná kombinace písmen a číslic. Popis elementu obsahuje informace o tom, jaké znalosti je třeba mít pro přidělení konkrétní úrovně. Umožňuje také vytvářet

Aktéroví tiskové sestavy pro samotný tisk nebo export do souboru ve formátu XHTML nebo PDF.

Speciální požadavky

Generátor PDF.

Předpoklady

Přihlášení Garanta KH do systému.

Ukončovací podmínky

Návrat na úvodní obrazovku.

Tok událostí

Je zde zahrnut tok události pro evidenci elementů.

Úroveň je zobrazena po výběru „*Správa elementů KH*“ na hlavní obrazovce.

Základní tok událostí

1. Přidání elementu KH

1. Systém zobrazí seznam dostupných elementů.
2. Aktér zvolí přidání elementu KH.
3. Zobrazí se formulář, kde aktér vyplní potřebné údaje.
4. Po vyplnění aktér potvrdí formulář pro vytvoření elementu.
5. Systém zkontroluje správnost zadaných údajů. V případě chybného zápisu navrátí systém aktéra do před-vyplněného formuláře s označením chybně zapsaných údajů. V opačném případě vytvoří a uloží element.
6. Pro přidání více nových elementů se opakují kroky 2-5.

zobrazené dostupné elementy jsou:

- Zkratku elementu KH, název elementu KH, popis elementu

formulář pro vyplnění údajů při přidávání elementu obsahuje:

- Zkratku elementu KH, název elementu KH, popis elementu.

Alternativní toky

1. Existence elementu KH

V případě, že při přidání elementu v kroku 4 dojde v databázi ke schodě jmen či zkratky elementu, navrátí systém aktéra do před-vyplněného formuláře s označením chybných údajů s informací že vyplněné údaje se shodují s již vytvořenými. Ty aktér napraví a provede krok 4. přidání nového elementu znovu.

2. Editace elementu KH

1. V každém řádku je umožněno uskutečnit editaci již existujícího elementu.
2. Řádek s elementem se zpřístupní pro zápis ve formulářové podobě.
3. Aktér doplní, nebo změní informace ve zkratce elementu, názvu elementu a popisu elementu.
4. Po změně zápisu aktér potvrdí uložení změn u daného elementu.
5. Systém zkontroluje správnost zadaných údajů. V případě chybného zápisu navrátí systém aktéra do před-vyplněného formuláře s označením chybně zapsaných údajů. V opačném případě uloží editovaný element do systému.
6. Pro editaci více položek se opakují kroky 2-5.

Popis elementu zahrnuje informace o potřebných znalostech popisuje:

- tři úrovně - junior, senior, expert.

3. Smazání elementu KH

1. V každém řádku je umožněno smazání elementu.
2. V případě požadavku na smazání elementu, se pro řádek vygeneruje dialog, potvrzení smazání elementu.
3. Aktér potvrdí smazání elementu, systém zkontroluje, zda tento element nemá někdo ze zaměstnanců přiřazen jako uznáný. V případě, že ne smaže element ze systému.
4. V opačném případě, systém vypíše chybovou hlášku a navrátí aktéra do seznamu elementů.
5. Pro smazání více položek se opakují kroky 2-4.

4. Vyhledávání elementu KH

1. Aktér zvolí vyhledání elementu.
2. Zpřístupní se prázdný formulář, do kterého aktér zadá údaje podle kterých chce vyhledávat.
3. Aktér potvrdí započetí vyhledávání podle zvolených údajů.
4. Systém vypíše seznam všech nalezených elementů.
5. Pro opakování hledání se aplikují kroky 1-4.

údaje podle kterých je možné vyhledávat jsou:

- Zkratku elementu KH, název elementu KH, popis elementu.

5. Vytvoření tiskové sestavy

1. Aktér zvolí vytvoření tiskové sestavy v seznamu vypsaných elementů.
2. Na stránce se umožní Aktérovi označit řádky s elementem, který chce přidat do tiskové sestavy. Aktér má možnost výběru položek, vše označit, veškeré označení zrušit, nebo má možnost vybrat elementy jednotlivě.
3. Pro zvolené řádky s elementy umožní vybrat sloupce pro tisk.
4. Po vybrání polí pro vytvoření tiskové sestavy aktér potvrdí vytvoření tiskové sestavy.
5. Vytvoří se stránka se všemi aktérem vybranými řádky a sloupci v samostatném okně prohlížeče.
6. Nyní je umožněno vytisknout sestavu na tiskárně standardním postupem pomocí prohlížeče, nebo zvolit uložení do souboru. V nabídce uložení do souboru jsou formáty XHTML a PDF.

možné sloupce tisku elementů umožní vybrat sloupce pro tisk:

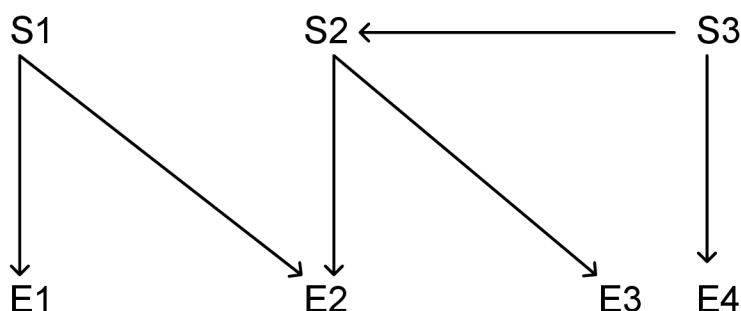
- Zkratka elementu KH, název elementu KH, popis elementu

Rozšíření

- Zobrazení elementu
- Vytvoření tiskové sestavy

Závislosti specializací

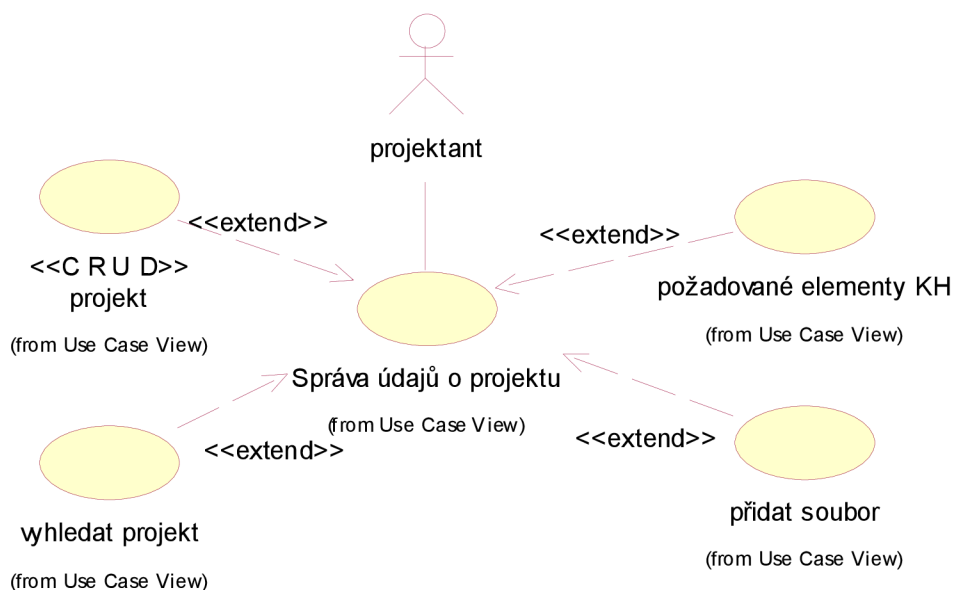
Jednotlivé specializace a elementy mezi sebou mohou být závislé. Běžná je závislost specializace na jednotlivých elementech, které specializaci tvoří. Závislost mezi samotnými specializacemi nebo jejich konkrétními stupni znalostí je umožněna také. Může také dojít k závislosti více specializací na jednom elementu jak zobrazuje obrázek 4.7.2.



obrázek 4.7.2: Závislosti elementů a specializací

To znamená, že specializace S1 se skládá ze dvou elementů E1 a E2. Specializace S2 z elementů E2 a E3. Specializace zahrnuje Specializaci S2, tedy elementy E2 a E3 a také samostatně element E4

4.8 Správa údajů o projektu



obrázek 4.8.1: Správa údajů o projektu

Popis

Případ užití umožňuje spravovat údaje o projektech, na kterých se pracuje. Správa údajů projektu zahrnuje informace o projektu z hlediska zpracování. Umožňuje vytvářet, zobrazit informace o vytvořených projektech, odebírat i aktualizovat projekty. V případě většího počtu projektů v systému lze použít vyhledávání. Ve správě údajů o projektu, jsou také obsaženy informace o tom, jaké znalosti na jaké úrovni je třeba mít pro zpracování projektu. Správa umožňuje také vytvářet aktérovi tiskové sestavy pro samotný tisk i export do souboru ve formátu XHTML nebo PDF.

Speciální požadavky

Databáze souborů, generátor PDF.

Následující budou specifikovány během další iterace.

Předpoklady

Přihlášení projektanta do systému.

Ukončovací podmínky

Návrat na úvodní obrazovku.

Tok událostí

Je zde zahrnut tok události pro správu údajů o projektu.

Úroveň je zobrazena po výběru „*Správa projektu*“ aktérem na hlavní obrazovce.

Základní tok událostí

1. Přidání projektu

1. Aktér zvolí přidání projektu.
2. Zobrazí se formulář pro přidání projektu.
3. Aktér zadá informace o projektu, kterými jsou:
4. Po potvrzení vyplněného formuláře, systém provede kontrolu zadaných dat a v případě chybného vyplnění zobrazí před-vyplněný formulář znovu s označením špatně vyplněných údajů, nebo údajů které jsou nevyplněné, ale jsou povinné. Provede se kontrola existence projektu v systému.
5. Po správném vyplnění formuláře systém přidá nový projekt do systému i s příloženým souborem a přidělí mu jednoznačné projektové číslo ID. Operaci lze také v průběhu přerušit, v tom případě případ užití končí.
6. V případě přidání více projektů do systému se opakují kroky 1-5.

zobrazené informace o projektu, jsou:

- zkratka projektu, název projektu, popis, datum zadání, seznam přidělených zaměstnanců, dokumentační soubor, požadované elementy projektu, stav projektu
- Název projektu obsahuje celý název v délce jedné věty. Zkratka projektu, je jedinečná kombinace písmen a číslic.

Povinnými údaji jsou:

- zkratka projektu, název projektu, popis, datum zadání, dokumentační soubor a stav projektu

Kontrola existence projektu v systému se provede:

- podle jména a zkratky

Alternativní toky

Zobrazení projektu

1. Systém zobrazí seznam projektů se základními údaji.
2. Pro zobrazení podrobností tj. kompletního výpisu položek projektu aktér potvrdí výběrem konkrétního projektu a výběrem zobrazení podrobností.
3. Systém načte všechny informace o projektu a vypíše je na obrazovku.
4. Pro zobrazení lze použít filtr.

základní zobrazení seznamu projektů je s těmito údaji:

- ID projektu, zkratka projektu, název projektu, popis, datum zadání, stav projektu

zobrazené podrobné (všechny) informace o projektu a vypsané na obrazovku:

- ID projektu, zkratka projektu, název, popis, datum zadání, seznam přidělených zaměstnanců, seznam vyžadovaných elementů, dokumentační soubor, stav projektu.

použití filtru umožňuje:

- výpis projektů podle stavu ve kterém se nacházejí, data jejich zadání

1. Existence projektu

V případě, že při přidávání projektu v kroku 4 dojde k nalezení existujícího projektu, systém zobrazí upozornění, že projekt již existuje a navrátí aktéra zpět s před-vyplněným formulářem s označením chybných údajů k jejich doplnění. Aktér může také přerušit prováděnou operaci, v tom případě případ užití končí.

2. Aktualizace projektu

1. Systém zobrazí seznam projektů se *základními údaji*.
2. Aktér vybere řádek s projektem, který bude chtít editovat potvrdí tento výběr.
3. Systém načte a zobrazí veškeré dostupné údaje o projektu ve formuláři pro možnou editaci.
4. Aktér upraví údaje, které je třeba editovat.
5. Po potvrzení úpravy projektu, systém provede kontrolu zadaných dat a v případě chybného vyplnění zobrazí před-vyplněný formulář znovu s označením špatně vyplněných údajů, nebo údajů které jsou nevyplněné, ale jsou povinné. Provede se kontrola existence projektu v systému.
6. V případě aktualizace změn u více projektů se opakují kroky 2-5.

základními údaji při zobrazení seznamu projektů jsou:

- ID projektu, zkratka projektu, název projektu, popis, datum zadání, stav projektu

zobrazené editovatelné údaje jsou:

- ID projektu, zkratka projektu, název, popis, datum zadání, seznam přidělených zaměstnanců, seznam vyžadovaných elementů, dokumentační soubor, stav projektu

povinnými údaji jsou:

- zkratka projektu, název projektu, popis, datum zadání, dokumentační soubor, stav projektu

kontrola existence projektu v systému se provede:

- podle jména a zkratky

3. Odebrání projektu

1. Systém zobrazí seznam projektů se základními údaji.
2. Aktér vybere řádek s projektem, který chce smazat a potvrdí smazání.
3. Zobrazí se systémem generované potvrzení smazání projektu.
4. Aktér potvrdí smazání projektu.
5. Systém projekt smaže ze systému.
6. V případě mazání více projektů ze systému se opakují kroky 2-5.

základní údaje zobrazí seznamu projektů jsou:

- ID projektu, zkratka projektu, název projektu, popis, datum zadání, stav projektu

možnost potvrzení smazání se aktivuje, až ve chvíli, kdy:

- pole „stav projektu“ je nastavené na pozici „zrušen“

4. Vyhledání projektu

1. Aktér zvolí vyhledat projekt.
2. Zobrazí se formulář pro vyhledání projektu.
3. Aktér zadá informace o projektu podle kterých chce vyhledávat.
4. Po potvrzení vyhledávání, systém provede vyhledání projektů v systému podle zadaných kritérií. Pro vyhledání je potřebný alespoň jeden zadaný údaj. Vyhledané projekty se vypíší na obrazovku společně s formulářem pro vyhledávání a počtem nalezených údajů. Po té je možné vybrat konkrétní projekt, nebo upřesnit vyhledávání a nechat vyhledat znovu. V případě výpisu většího množství projektů je výpis výběr rozdělen na více stránek.
5. V případě vyhledání více projektů v systému se opakují kroky 1-4.

k vyhledávání projektu lze použít:

- ID projektu, zkratka projektu, název, popis, datum zadání, seznam přidělených zaměstnanců, seznam vyžadovaných elementů, stav projektu.

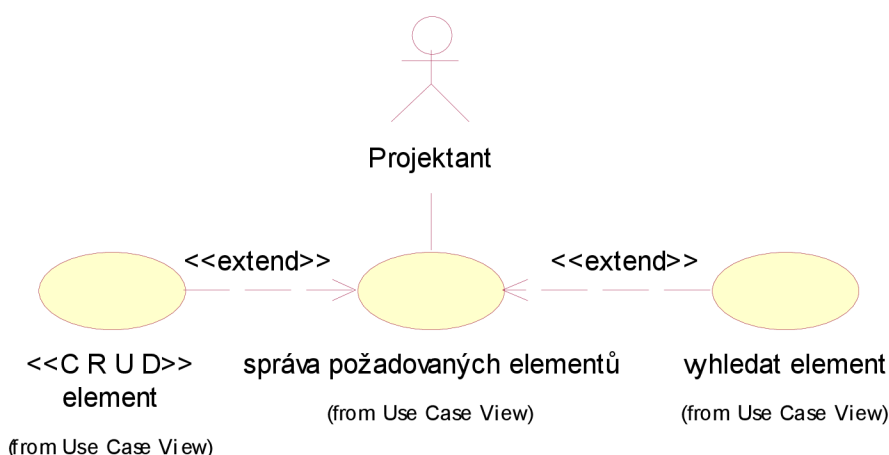
5. Správa požadovaných elementů projektu

funkčnost správy vyžadovaných elementů je popsána v kapitole 4.9

Rozšíření

- Vytvoření tiskové sestavy
- Správa souborů

4.9 Správa požadovaných elementů



obrázek 4.9.1: Správa požadovaných elementů

Popis

Případ užití umožňuje spravovat *projektantovi* údaje o potřebných elementech k projektu. Tj. umožňuje elementy k projektům přidávat, mazat, upravovat, zobrazovat i vyhledávat.

Speciální požadavky

Budou specifikovány během další iterace.

Předpoklady

Přihlášení Aktéra do systému.

Aktér se nachází v případě použití „*Správa projektu*“

Ukončovací podmínky

Návrat do správy údajů o pohovoru.

Tok událostí

Zde je zahrnutý tok událostí pro správu požadovaných elementů k projektu. Úroveň je zobrazena po výběru správy požadovaných elementů projektu ve správě údajů o projektu.

Základní tok událostí

Přidání požadovaného elementu

1. Aktér zvolí přidání požadovaného elementu.
2. Zobrazí se formulář pro přidání elementu.
3. Aktér vybere ze seznamu elementů konkrétní element.
4. V položce stupeň znalosti vybere aktér stupeň znalosti potřebný pro daný projekt.
5. Po potvrzení přidání, se provede kontrola existence požadovaného elementu pro projekt v systému a uloží znalost s jeho stupněm do systému. Operaci lze také přerušit, v tom případě případ užití končí.

6. V případě přidání více požadovaných elementů do systému se opakují kroky 1-5.

zobrazený stupeň znalosti je dle objemu znalostí rozdělen:

- do tří vědomostních kategorií (junior, senior, expert) tedy hodnoty 1,2,3.

Alternativní toky

Zobrazení požadovaného elementu

1. Systém zobrazí seznam elementů se základními údaji.
2. Pro zobrazení podrobností o elementu Aktér vybere konkrétní element a potvrdí jeho výběr.
3. Systém načte a zobrazí podrobný popis elementu.

Základní údaje zobrazené v seznamu elementů:

- Element KH, stupeň znalosti (1-3)

Podrobný popis elementu obsahuje:

- Popis každého stupně, společně se zkratkou elementu.

1. Existence požadovaného elementu

V případě akce „Přidání požadovaného elementu“ kdy systém nalezne existující požadovaný element projektu, systém vypíše chybové hlášení, že je element k projektu již přiřazen. Navrátí aktéra do menu přidávání požadovaného elementu a zobrazí výběr dalšího požadovaného elementu k projektu znovu.

2. Editace požadovaného elementu

1. Systém zobrazí seznam požadovaných elementů projektu.
2. Aktér vybere řádek s požadovaným elementem, který bude chtít editovat potvrdí výběr.
3. Systém načte a zobrazí údaje o požadovaném elementu projektu ve formuláři pro možnou editaci.
4. Aktér upraví zobrazené údaje, které je třeba editovat.
5. Po potvrzení uložení požadovaného elementu, se provede kontrola existence požadovaného elementu pro projekt v systému. Pak systém aktualizuje element KH daného uchazeče.
6. V případě nalezení již existujícího elementu v systému, se aplikuje stejný krok jako při „přidávání požadovaného elementu“.
7. V případě editace změn více požadovaných elementů u projektu se opakují kroky 2-7.

seznam požadovaných elementů projektu obsahuje:

- Zkratku požadovaného elementu, název požadovaného elementu, stupeň požadované znalosti

Zobrazené údaje, které je třeba editovat jsou:

- Element KH, stupeň znalosti (1-3)

3. Smazání elementu KH

1. Systém zobrazí seznam požadovaných elementů projektu.
2. Aktér vybere řádek s požadovaným elementem, který bude chtít smazat potvrdí tento výběr.
3. Zobrazí se systémem generované potvrzení smazání požadovaného elementu.
4. Aktér potvrdí smazání požadovaného elementu.
5. Systém smaže požadovaný element.
6. V případě mazání více požadovaných elementů ze systému se opakují kroky 2-5.

seznam požadovaných elementů projektu obsahuje:

- Zkratku požadovaného elementu, název požadovaného elementu, stupeň požadované znalosti

4. Vyhledání elementu

1. Aktér zvolí vyhledat projekt.
2. Zobrazí se formulář pro vyhledání projektu.
3. Aktér zadá informace o projektu podle kterých chce vyhledávat.
4. Po potvrzení vyhledávání, systém provede vyhledání projektů v systému podle zadaných kritérií. Pro vyhledání je potřebný alespoň jeden zadaný údaj. Vyhledané projekty se vypíší na obrazovku společně s formulářem pro vyhledávání a počtem nalezených údajů. Po té je možné vybrat konkrétní projekt, nebo upřesnit vyhledávání a nechat vyhledat znovu. V případě výpisu většího množství projektů je výpis výběr rozdělen na více stránek.
5. V případě vyhledání více projektů v systému se opakují kroky 1-4.

k vyhledávání projektu lze použít:

- ID projektu, zkratka projektu, název, popis, datum zadání, seznam přidělených zaměstnanců, seznam vyžadovaných elementů, stav projektu.

Rozšíření

- Vytvoření tiskové sestavy

5 DOPLŇKOVÁ SPECIFIKACE

5.1 Předmluva

Tato kapitola definuje požadavky systému, které nejsou zahrnuty v případech použití use-case modelu. Tyto Doplnkové požadavky, neboli Supplementary Specifications dle metodiky RUP s use-case model tvoří dohromady úplnou množinu požadavků na systém.

Rozsah

Tato Doplnková specifikace definuje další požadavky pro úplnost. Umožňuje evidovat:

- Zaměstnance a jejich požadavky na přezkoušení
- Znalosti zaměstnanců a úroveň těchto znalostí.
- Výsledky přezkoušení zaměstnanců, jejich zápis do databáze
- Pohovory, které zaměstnanec absolvuje

Reference

1. Vize
2. Jednotlivé případy použití z kapitoly 4

5.2 Použitelnost

Tato sekce obsahuje seznamy funkčních požadavků zobecněně. Neváže se jen na jeden Use-case. Požadavky se týkají použitelnosti systému.

Logování systémových chyb

Všechny systémové chyby by měly být logovány. Fatální chyby by měly vést k bezpečnému ukončení systému.

Systémové hlášení chyby by mělo obsahovat text s popisem chyby, provozní chybové číslo (error code), modul detekující stav ve kterém se vyskytla systémová chyba, data na kterých se chyba projevila a časové razítko výskytu chyby. Všechny systémové chyby by měly zůstat v databázi sloužící pro logování chyb systému na serveru.

Vzdálený přístup

Veškerá funkcionalita by měla být dostupná vzdáleně přes Internet. To si vyžádá běžící aplikaci nebo službu na vzdáleném serveru.

Design jednoduchého použití

Uživatelské prostředí systému pro evidenci znalostí zaměstnanců by mělo být designováno pro lehké použití počítačově gramotného člověka, bez nutnosti absolvování dodatečného zaškolení

On-line nápověda

Každá ucelená část systému evidence znalostí zaměstnanců by měla mít On-line nápovědu obsahující jednotlivé kroky v instrukcích pro použití v systému. Tato nápověda by měla obsahovat definice termínů a zkratk.

5.3 Spolehlivost

V této části je uveden seznam požadavků na spolehlivost.

Dostupnost

Systém by měl být dostupný 24 hodin denně, 7 dní v týdnu.

Průměrná doba mezi pády systému (MTBF)

Průměrná doba mezi pády systému by měla přesahovat 400 hodin.

Průměrná doba zotavení systému (MTTR)

Průměrná doba zotavení systému by měla být kratší než 20 min.

Přihlášení

Veškeré operace by měly být proveditelné pouze po úspěšném přihlášení uživatele do systému.

5.4 Výkonnost

Výkonnostní charakteristiky systému:

Současně pracující uživatelé

Systém by měl podporovat až 100 současně pracujících uživatelů.

Současně prováděné transakce

Systému, by měl pojmout až 10 současně prováděných transakcí.

Doba odezvy přístupu do databáze

Doba odezvy systému by neměla být delší než 5 sekund.

Nouzový režim

Akceptovatelný režim vytižení informačního systému (nouzový režim) je takový, kdy je systém schopen obsluhovat připojené uživatele a provádět transakce se zpožděním maximálně 4x delším, než je běžný stav.

Objem dat

Systém bude spravovat okolo 20.000 záznamů v databázi.

Prostředky využití paměti, disku

K využití paměti serveru na systém by mělo být uvolněno až 60% volné paměti.

Disková kapacita by měla dosahovat 1,5 násobek volného místa stávající velikosti databáze na disku.

5.5 Omezení Návrhu

Tato sekce uvádí seznam omezení architektury systému.

Požadavky platformy

Klientská část systému by měla být spustitelná na jakémkoliv počítači i s architekturou odlišnou od IBM. Tyto počítače však musí podporovat grafické rozhraní.

Serverová část aplikace by měla být provozována na stabilním systému serverového modelu, tedy UNIX, Linux, případně Windows.

Internetové prohlížeče

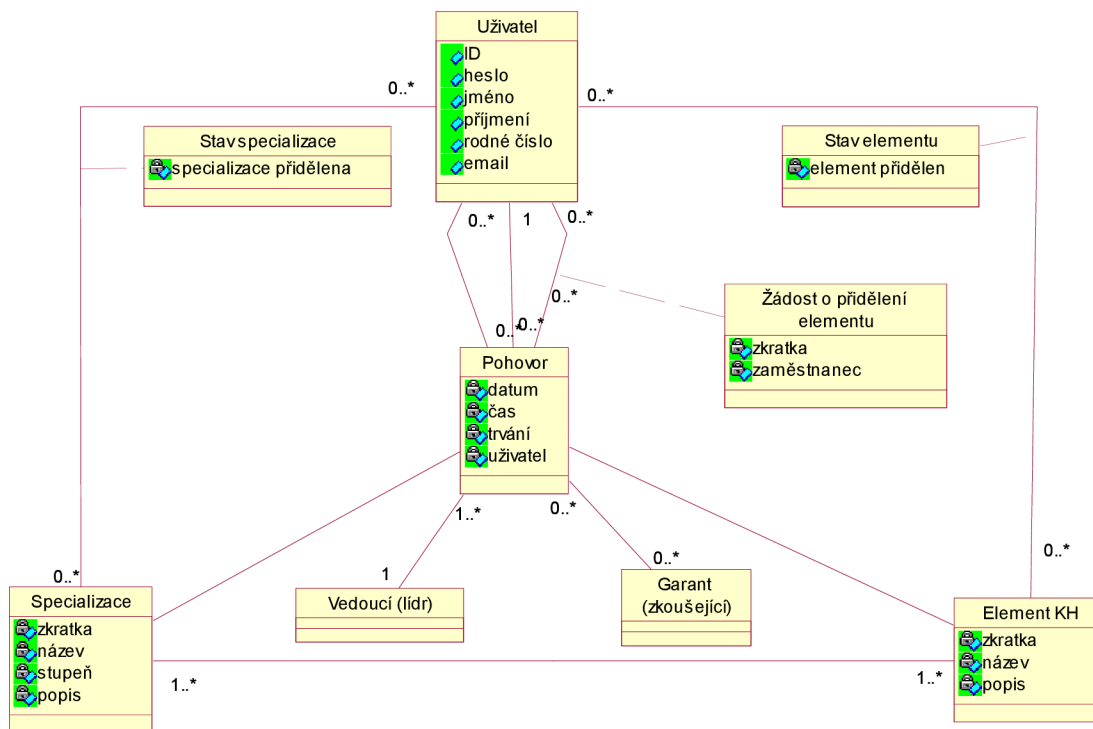
Klientské počítače musí být vybaveni Internetovým prohlížečem splňujícím validitu normy W3C. Případně nejrozšířenějšího prohlížeče Internet Explorer.

Kompatibilita jazyka

Serverová část systému bude nezávislá na systému klienta. Předávání informací bude probíhat pomocí značkovacího jazyka HTML.

6 MODEL PROBLÉMOVÉ DOMÉNY

Zde je znázorněn model problémové domény. Jedná se o množinu vzájemně závislých konceptů, ukazuje základní entity a vazby systému. Model reprezentuje obchodní pohled, modeluje data a aktivity, obsahuje business objekty.



obrázek 5.5.1: Model problémové domény

7 PLÁN REALIZACE

Z důvodu rozsáhlosti informačního systému proběhne realizace celého informačního systému ve třech etapových částech.

7.1 Funkčnosti realizované v jednotlivých etapách

První etapa

Pro první etapu vývoje je plánována následující funkčnost:

- Přihlášení.
- Přístupová práva.
- Prohlížení znalostí zaměstnance.
- Zadávání, aktualizace a prohlížení znalostí o zaměstnancích.
- Evidence elementů znalostí.

Druhá etapa

Ve druhé etapě vývoje se počítá s touto funkčností:

- Žádost o přidělení znalosti.
- Možnost uživatele měnit přístupové heslo.

Třetí etapa

Třetí etapa bude zahrnovat:

- Vytváření tiskových sestav.
- Výpis požadovaných elementů

7.2 Výstupy realizace a její časový harmonogram

Aplikace

Aplikace bude výstupem každé etapy jako celistvá část splňující specifikace z oddílu 7.1. Výstup bude předán v počátku zahájení uživatelských akceptačních testů (UAT) dané etapy.

Dokumentace

Dokumentace je sekundárním výstupem informačního systému ihned po aplikaci. Dokumentace zahrnuje On-line nápovědu, instalační a provozní dokumentaci. Dokumentace bude předána po předání aplikace, v průběhu UAT testů pro každou etapu vývoje.

Časový harmonogram

Realizace zhotovení projektu se předpokládá do prosince roku 2007

8 SOFTWAREOVÁ ARCHITEKTURA

8.1 Předmluva

Tato kapitola poskytuje obsáhlý přehled systému, z pohledu rozdílné architektury pohledu ke znázornění systému. Je plánované vystihnout a vyjádřit popis architektury rozhodnutí, podle kterých má být systém zhotoven.

8.1.1 Rozsah

Kapitola popisuje architektonické přehledy systému pro databázi znalostí zaměstnanců. Systém je určen k databázi znalostí firmy.

Kapitola je vygenerována za pomoci Analysis & Design Model implementovaného v Rose. Většina z částí byla vyjmuta z tohoto modelu a ze šablony Architecture Document, ta je součástí uvedené literatury [7]. Informace obsahující Use-case realizace jsou čerpány z [6].

8.1.2 Definice, akronymy a zkratky

Viz slovníček pojmů 11.3

8.1.3 Reference

použité reference:

1. Vize
2. Jednotlivé případy použití uvedené v Use Case pohledu
3. Doplnková specifikace
4. Slovníček pojmů

8.2 Reprezentace architektury

Tato architektura popisuje softwarovou architekturu jako tok systému, jak je reprezentován řadou pohledů: Use-Case pohledem, Logickým pohledem, vývojovým pohledem, implementačním pohledem a pohledem nasazení. Pro zobrazení modelů je využito jazyka UML (Unified Modeling Language) v s použitím Rose modelů.

8.3 Cíle a omezení architektury

Tato část popisuje softwarové požadavky a objektivnost významného dopadu na architekturu. Například bezpečnost, zabezpečení soukromí, přenosnost systému, distribuci a znovu použitelnost. Vyskytuje se zde speciální omezení, kterým může být design implementační strategie, vývojových nástrojů, týmové struktury, plánu.

Mezi klíčové požadavky a systémová omezení patří:

1. Všechny existující odkazy do systému, musí být přístupné ze sítě Internet.
2. Systém musí zajistit ochranu proti vniknutí neautorizovaným přístupem. Pro všechny vzdálené přístupy je po uživateli požadována identifikace a heslo.

3. Systém bude implementován jako systém Client-Server. Klientská část bude spuštěna z uživatelského PC, serverová bude provozována na serveru firmy poskytující systém.
4. Před samotným návrhem architektury musí být zváženy výkonnostní požadavky, jak je uvedeno v kapitolách 3.9 a 5.4

8.4 Use-case pohled

Use-case pohled představuje důležitý vstup k výběru množiny případů užití, které je zaměřeno na opakování. Popisuje množinu případů užití, které reprezentují významnou centrální funkcionalitu. Také popisuje množinu případů užití, které mají značný dopad na pokrytí architektury (používají mnoho architektonických prvků) nebo které zdůrazňují či demonstrují specificky citlivé místa architektury.

Hlavním případy jsou:

- Správa zaměstnanců
- Správa pohovoru
- Správa elementů KH zaměstnance
- Žádost o přidělení znalosti
- Správa elementů KH
- Správa údajů o projektu
- Správa požadovaných elementů KH

Všechny tyto případy užití jsou spouštěny některými u aktérů: Zaměstnanec, Garant KH, Školitel.

Z toho odvozené, architektonicky významné případy užití jsou:

- Seznam
- Vyhledávání
- Přidělení znalosti
- Správa pohovoru
- Tisková sestava

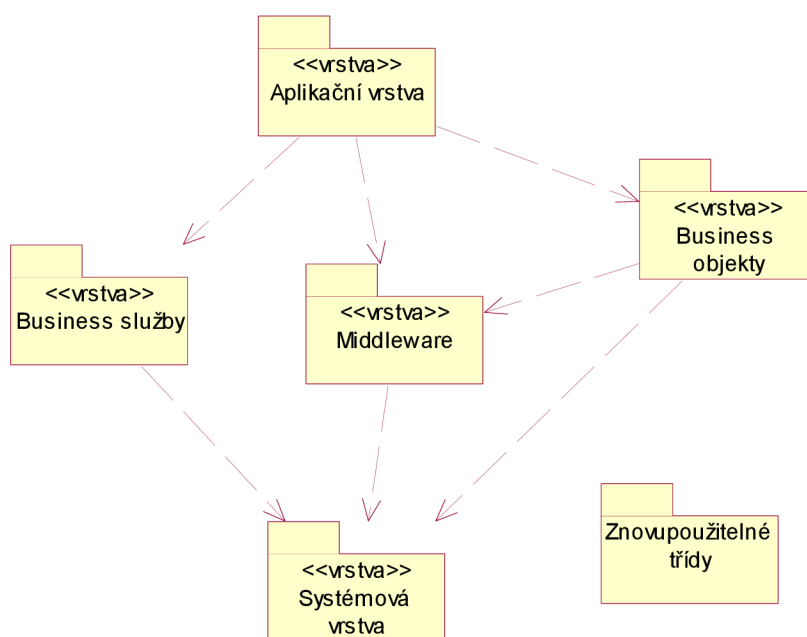
8.5 Logický pohled

Přehled

Logický pohled popisuje nejdůležitější třídy, jejich organizaci v balíčky a subsystémy, a organizaci těchto subsystémů do vrstev. Také popisuje nejdůležitější use-case realizace, například dynamická stránka architektury. S diagramy tříd mohou být demonstrovány vztahy tříd mezi architektonicky významnými třídami, subsystémy, balíčky a vrstvami.

Logický pohled Systému evidence znalostí zaměstnanců je složen z pěti hlavních balíčků: **Aplikační vrstva, Business služby, Business objekty, Middleware, Systémová vrstva.**

8.5.1 Popis architektury, návrhy významných balíčku



obrázek 8.5.1: Popis architektury – rozvrstvení základních vrstev

Aplikační vrstva – vrstva uživatelského rozhraní

Aplikační vrstva zahrnuje všechny boundary třídy (třídy rozhraní), které jsou reprezentovány prostředím, které uživatel vidí a se kterým pracuje, tedy uživatelské prostředí. Balíček Aplikační vrstvy obsahuje třídy pro jednotlivé formuláře, pomocí kterých aktéři komunikují se systémem. Třídy s rozhraním jsou určeny pro podporu: přihlášení, udržování údajů o zaměstnancích, správu elementů KH zaměstnanec, správu pohovoru zaměstnanec, správu požadovaných elementů KH, správu elementů KH a správu požadavků na přiznání elementu.

Business služby

Vrstva Business služeb představuje procesní vrstvu. Zahrnuje kontrolní třídy, které reprezentují use case, řídí chování aplikace. Tato vrstva reprezentuje hranici mezi uživatelem a střední vrstvou. Balíček Business služeb obsahuje třídy, které řídí chování a komunikují s IS na základě požadavků přezkoušení znalostí na pohovoru.

Business objekty

Vrstva Business objektů obsahuje všechny entity tříd, které reprezentují věcné vlastnosti problémové domény. Tyto business objekty jsou transformovány. V první fázi na konceptuální objekty, ty jsou postupným upřesňováním v druhé fázi transformovány na softwarové objekty.

Systémová vrstva

Systémová vrstva reprezentuje to, co nám dává k dispozici cílový operační systém a nebo zvolené implementační prostředí. Obsahuje třídy určené k zabezpečení přístupu do IS. Objektově orientovaný systém je takto podporován nízko-úrovňovým systémovým přístupem.

Middleware

Jedná se o proces integrace, který umožňuje spolupráci aplikací bez výrazných zásahů do těchto aplikací. Komunikace probíhá přímo na vyšší abstraktní úrovni. Middleware tímto tvoří střední vrstvu, tedy prostředníka mezi aplikacemi a jejich výstupy na straně klientů.

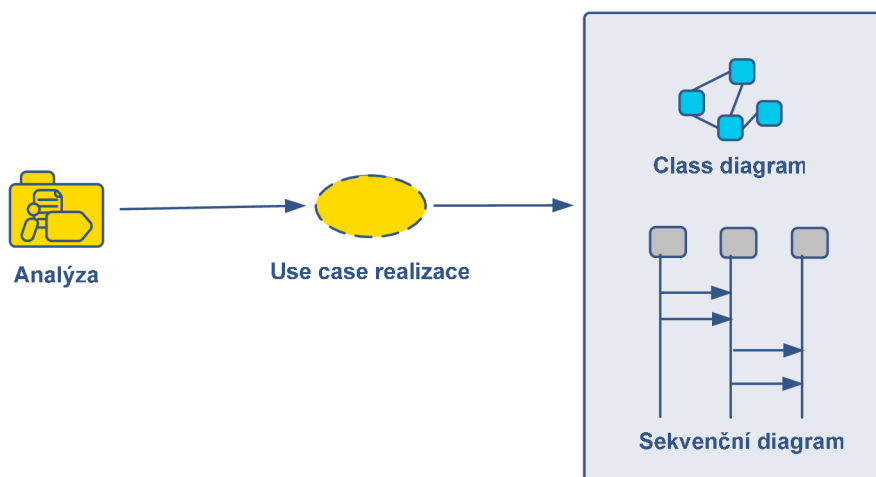
Znovupoužitelné třídy

Balíček Znovupoužitelných tříd zahrnuje třídy, které podporují a obsahují seznam funkcí a vzorů.

8.5.2 Use-Case realizace

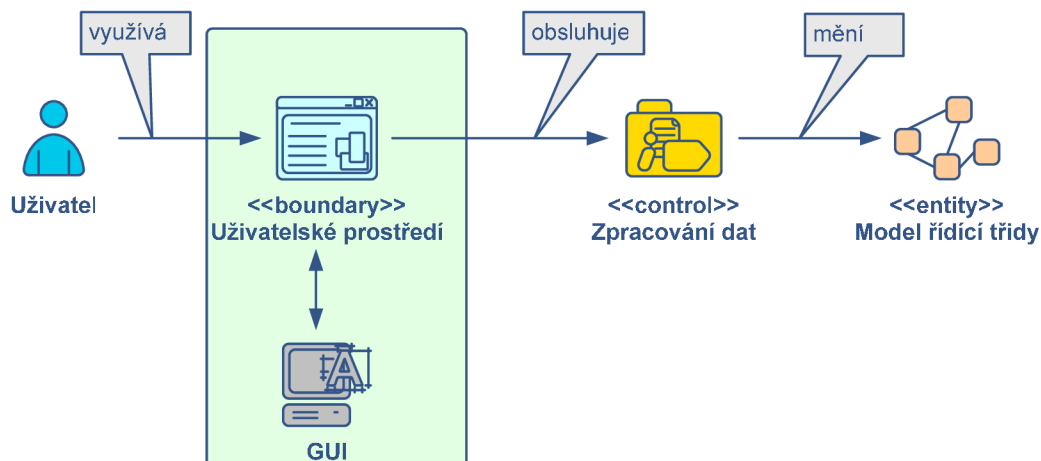
Tato část ilustruje jak systém ve skutečnosti pracuje. Pomocí několika vybraných use case a vysvětluje jak různý design model elementů přispívá k jejich funkčnosti.

Realizace případů užití je popisem spolupráce tříd pro dosažení požadovaného chování systému. Tato Use Case realizace případu užití je zobrazena na následujícím obrázku 8.5.2



obrázek 8.5.2: Use-Case realizace

Využitím tříd stereotypů typu **entity**, **boundary**, **control** (MVC – Model, View, Controller) získám následující rozdělení, jak je zobrazené na následujícím obrázku - obrázek 8.5.3. Tyto stereotypy reprezentují objektový model založený na frameworku model-view-controller vhodném pro další reprezentaci.

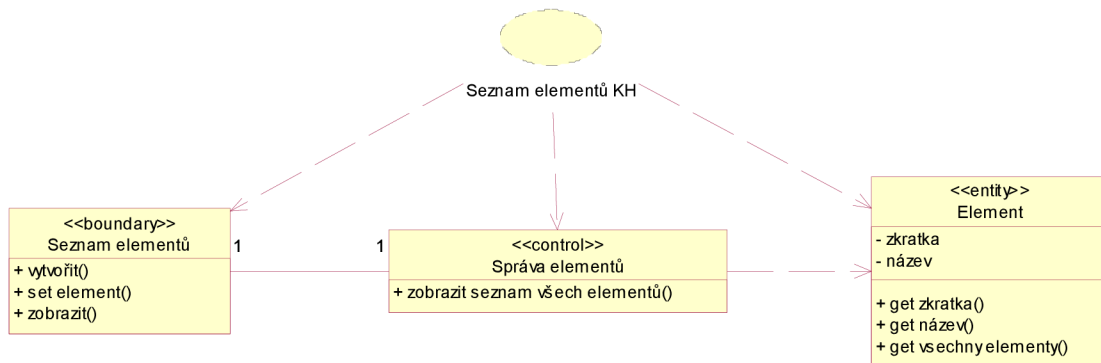


obrázek 8.5.3: Stereotypy tříd boundary, control, entity

8.5.3 Seznam

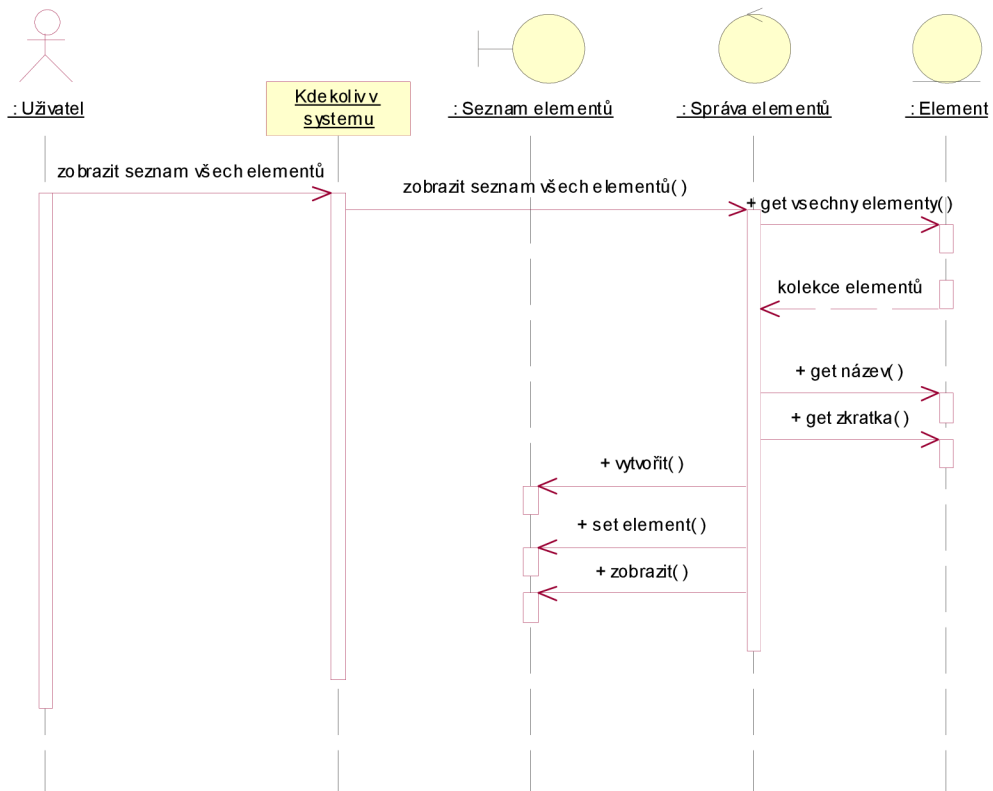
Tento případ použití je v diagramech znázorněn jako *Seznam elementů KH*.

Class diagram



obrázek 8.5.4: Class diagram – Seznam elementů

Sekvenční diagram

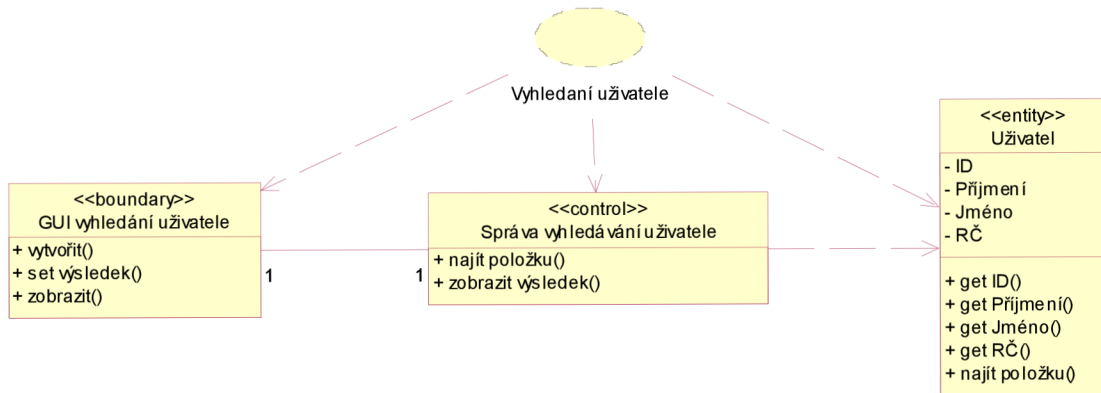


obrázek 8.5.5: Sekvenční diagram – Seznam elementů

8.5.4 Vyhledávání

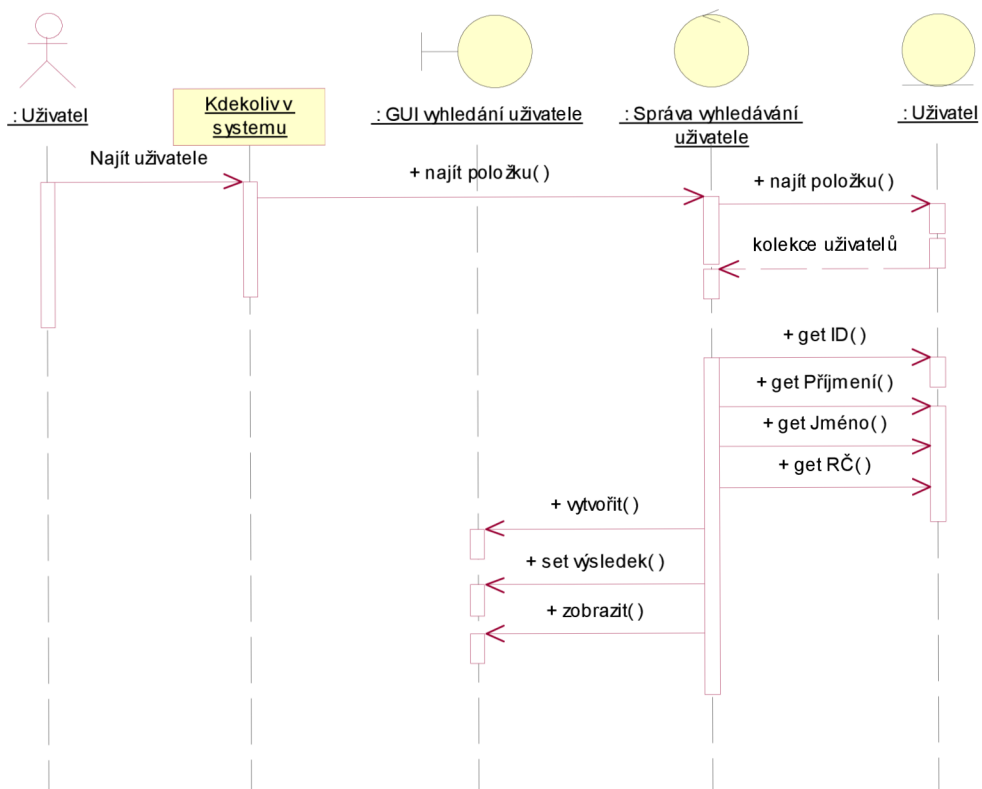
Tento případ použití je v diagramech znázorněn jako *Vyhledání uživatele*.

Class diagram



obrázek 8.5.6: Class diagram – Vyhledání uživatele

Sekvenční diagram

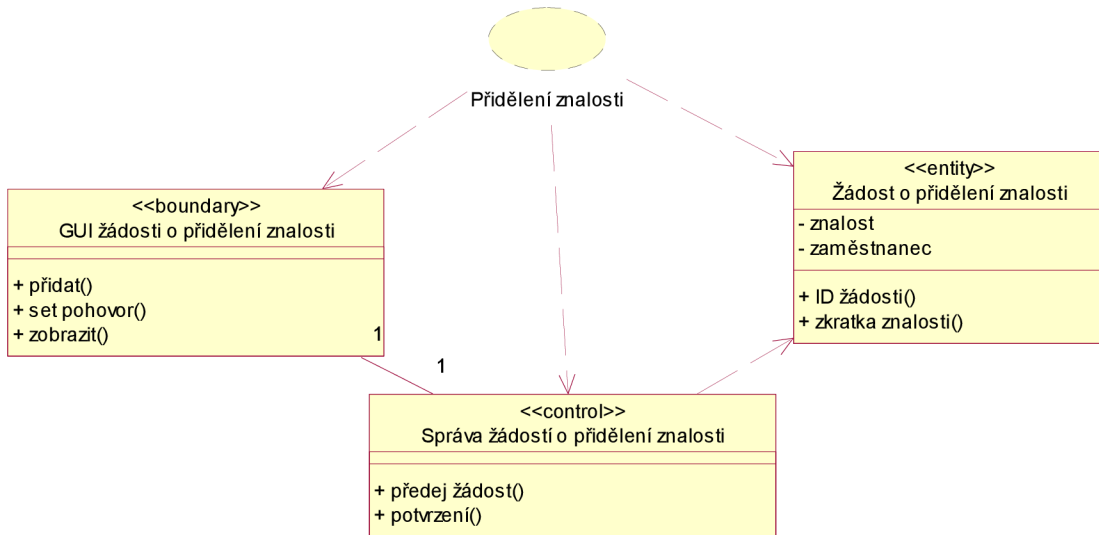


obrázek 8.5.7: Sekvenční diagram – Vyhledání uživatele

8.5.5 Přidělení znalostí

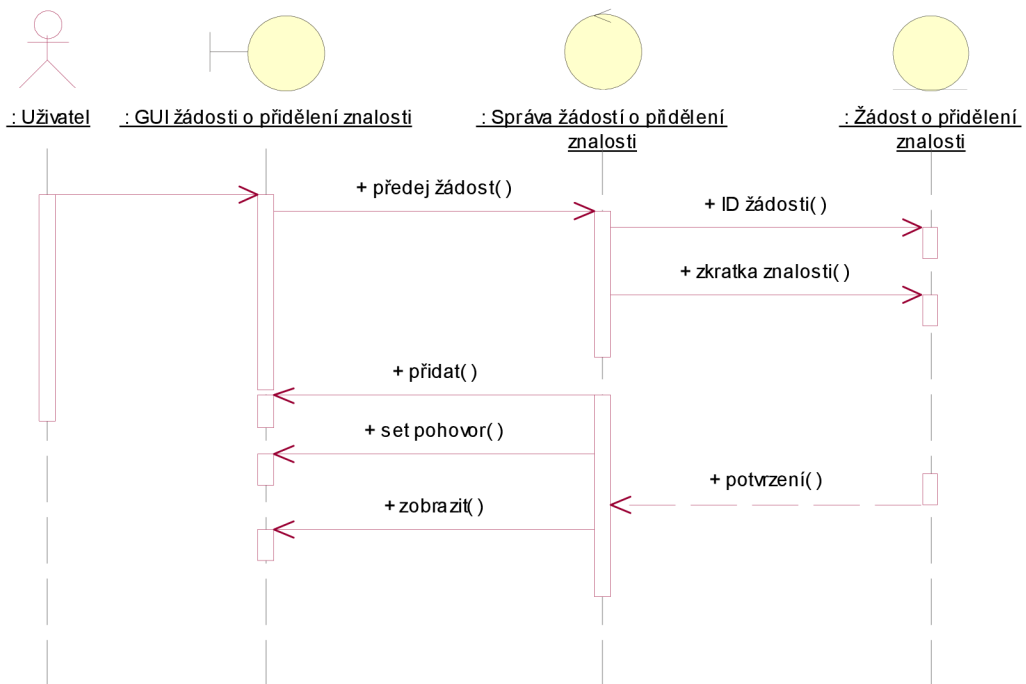
Tento případ použití je v diagramech znázorněn jako *Přidělení znalosti uživatele*.

Class diagram



obrázek 8.5.8: Class diagram – Přidělení znalostí

Sekvenční diagram

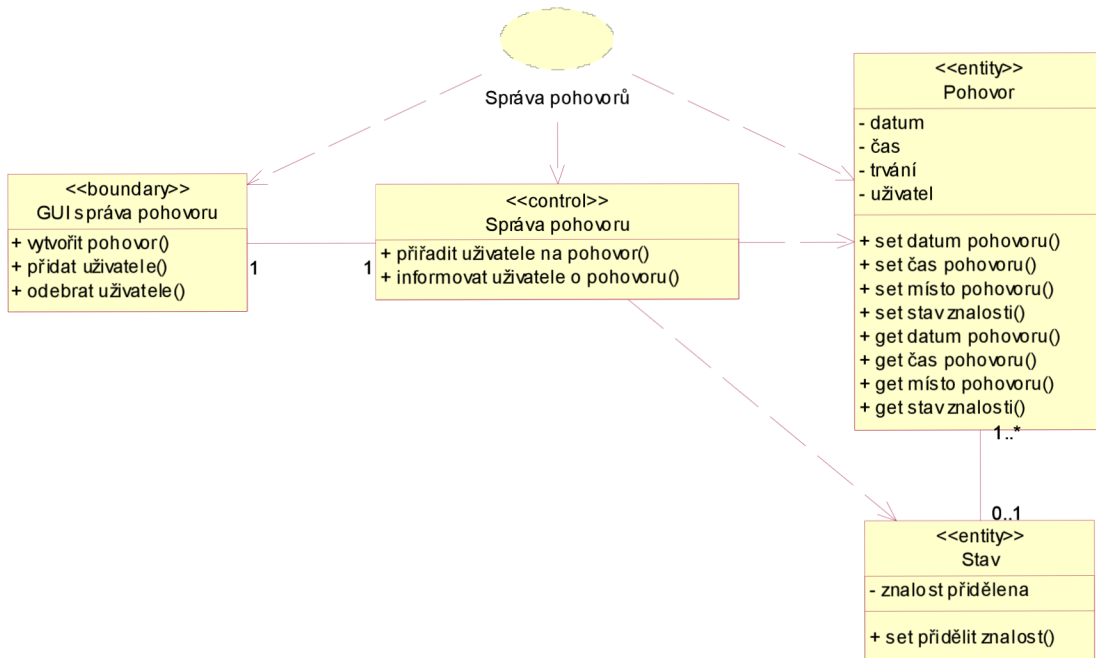


obrázek 8.5.9: Sekvenční diagram – Přidělení znalostí

8.5.6 Správa pohovoru

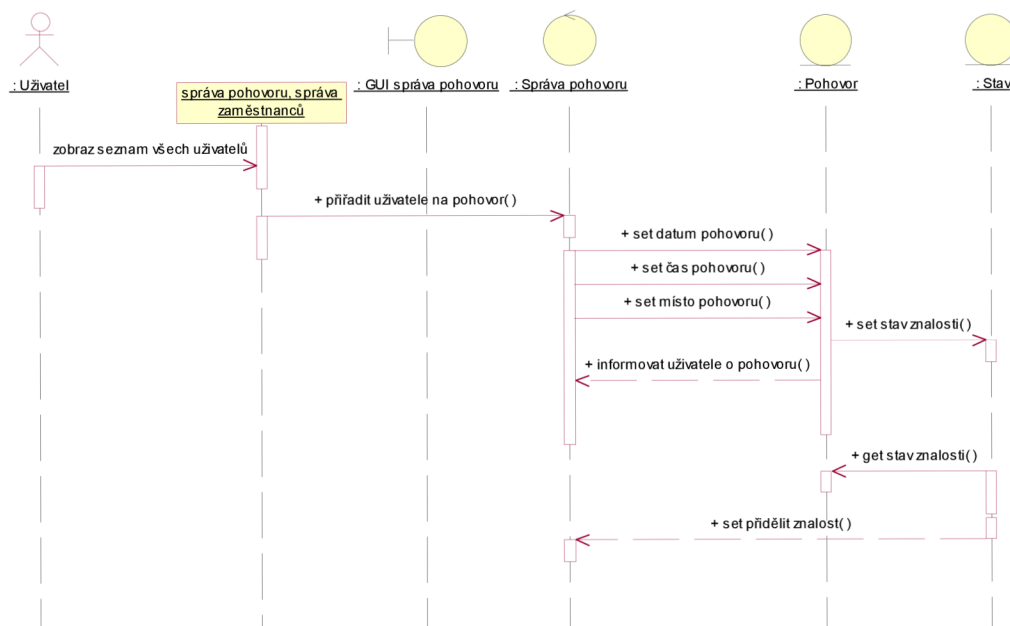
Tento případ použití je v diagramech znázorněn jako Správa pohovoru.

Class diagram



obrázek 8.5.10: Class diagram – Správa pohovoru

Sekvenční diagram

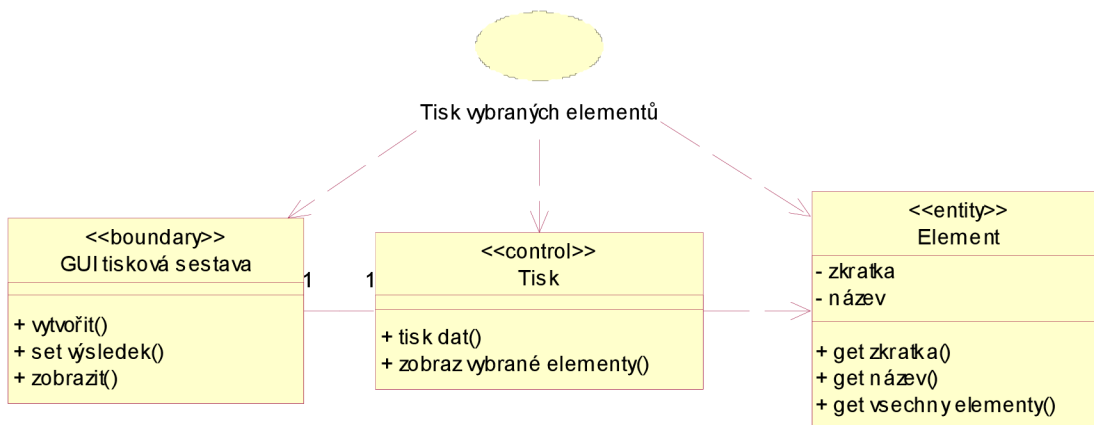


obrázek 8.5.11: Sekvenční diagram – Správa pohovoru

8.5.7 Tisková sestava

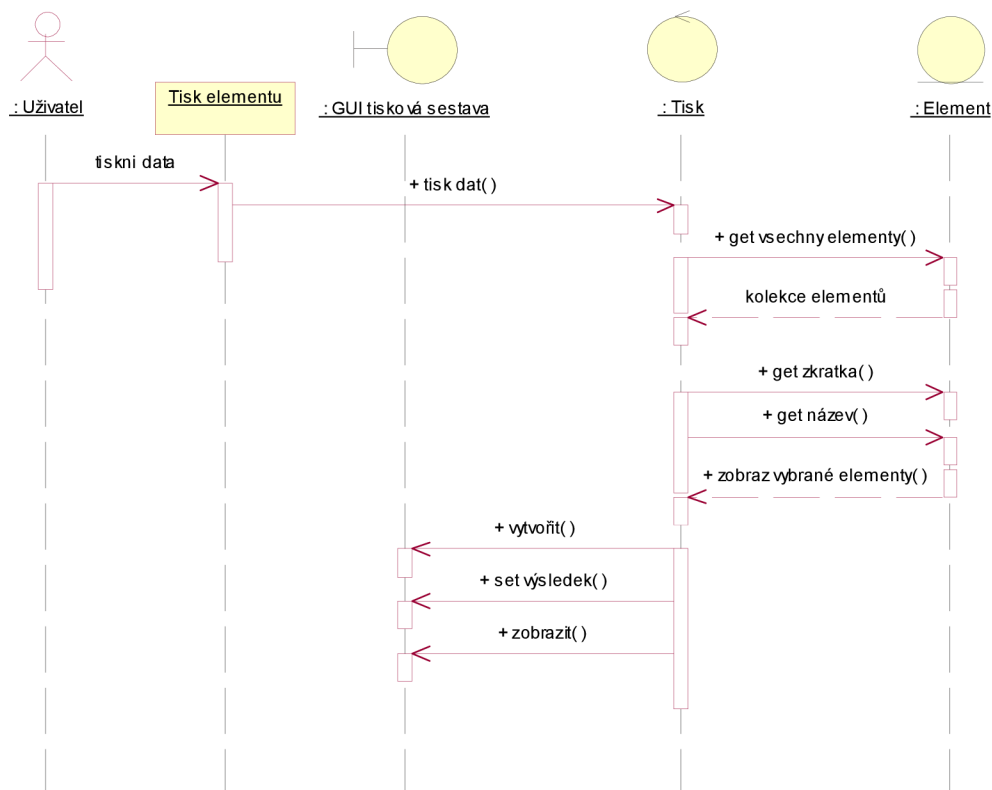
Tento případ použití je v diagramech znázorněn jako Tisk vybraných elementů.

Class diagram



obrázek 8.5.12: Class diagram – Tisk elementů

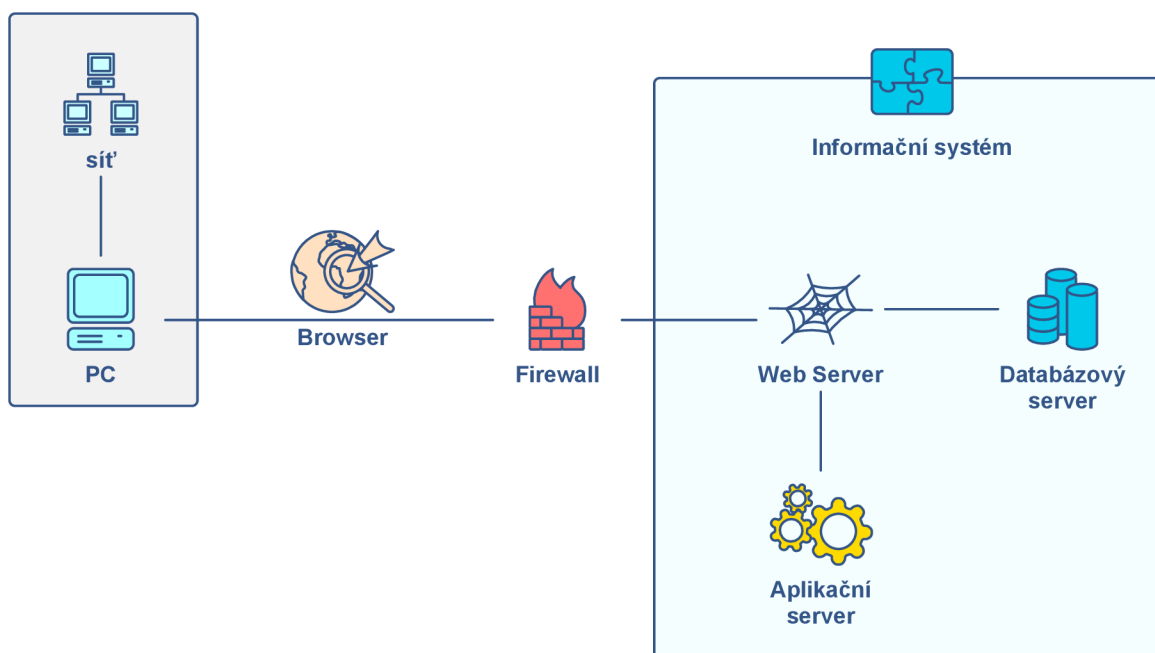
Sekvenční diagram



obrázek 8.5.13: Sekvenční diagram – Tisk elementů

8.6 Pohled nasazení

Informační systém se nachází na serveru, jehož součástí je databázový server. V případě této práce reprezentovaný *MySQL*. Dále webový server, ve výsledném prototypu zastoupen *Apache* serverem. Dynamická komunikace s web serverem probíhá pomocí Ruby (interpretovaný skriptovací programovací jazyk). Uživatel obsluhující PC má pouze taková práva pod kterými se přihlásí do systému. Pro komunikaci a zobrazení informací z IS je zapotřebí webový prohlížeč. Výstup z IS je uživateli zobrazen v HTML kódu. Firewall filtruje nežádoucí přístupy.



Obrázek 8.6.1: Pohled nasazení

8.7 Výkonnost

Softwarová architektura má následující výkonnostní charakteristiky systému:

1. Systém by měl podporovat až 100 současně pracujících uživatelů.
2. Systému, by měl pojmout až 10 současně prováděných transakcí.
3. Doba odezvy systému by neměla být delší než 5 sekund.
4. Akceptovatelný režim vytížení informačního systému (nouzový režim) je takový, kdy je systém schopen obsluhovat připojené uživatele a provádět transakce se zpožděním maximálně 4x delším, než je běžný stav.
5. Systém bude spravovat okolo 20.000 záznamů v databázi.
6. K využití paměti serveru na systém by mělo být uvolněno až 60% volné paměti.
7. Disková kapacita by měla dosahovat 1,5 násobek volného místa stávající velikosti databáze na disku.

8.8 Kvalita

Softwarová architektura má následující kvalitativní charakteristiky spolehlivosti systému:

1. Systém by měl být dostupný 24 hodin denně, 7 dní v týdnu.
2. Průměrná doba mezi pády systému by měla přesahovat 400 hodin.
3. Průměrná doba zotavení systému by měla být kratší než 20 min.
4. Veškeré operace by měly být proveditelné pouze po úspěšném přihlášení uživatele do systému.

9 UŽIVATELSKÉ ROZHRAVNÍ

V této kapitole je zobrazeno uživatelské rozhraní pro vybrané případy užití.

- Přihlášení
- Správa zaměstnanců
- Správa elementů KH
- Správa elementů KH zaměstnance
- Podrobnosti elementu

Prototyp webové aplikace

Aplikace je napsány podle zásad architektury Model-View-Controller, která důsledně odděluje funkční část aplikace (Model) od způsobu prezentace uživateli (View). Ke spojení funkční části s prezentací dohromady (Controller) bylo použito programovacího jazyka Ruby. Pro informace vedoucí k sestavení prototypu byla použita literatura uvedená v [8], [9], [10]

9.1 Přihlášení

Prototyp ukazuje uživatelské rozhraní pro přihlášení uživatele do systému. K tomuto prototypu se váže přihlášení Garanta KH.



The screenshot shows a web interface titled "Systém databáze znalostí zaměstnanců". It features a login form with the following elements:

- A header with the title "Systém databáze znalostí zaměstnanců" and a link "Odhlásit" in the top right corner.
- A message: "Pro přihlášení je nutné zadat uživatelské jméno a heslo."
- Input fields for "Jméno:" and "Heslo:".
- A "Přihlásit" button.
- A footer with a link "Zpět na hlavní stránku" and a copyright notice "(c) Všechna práva vyhrazena".

obrázek 9.1.1: GUI - Přihlášení uživatele do systému

9.2 Správa zaměstnanců

Prototyp zobrazuje výpis seznamu zaměstnanců, s možnostmi pro přidání nového zaměstnance do systému. Editačním tlačítkem na příslušném řádku lze zaměstnance editovat, obdobným způsobem lze zaměstnance smazat. Zobrazením podrobností je popsáno v podkapitole 9.4 Správa elementů KH zaměstnance.

Systém databáze znalostí zaměstnanců

[Odhlásit](#)

Seznam zaměstnanců:

ID	Jméno	Příjmení	Rodné číslo	email	město				
1	Václav	Novák	7312135568	v.novak@unicorn.cz	Praha	Podrobnosti	Editovat	Smazat	Přifadit znalosti
3	Petr	Navrátil	7512128512	p.navratil@unicorn.cz	Brno	Podrobnosti	Editovat	Smazat	Přifadit znalosti
2	Pavel	Zámečník	7509145211	p.zamecnik@unicorn.cz	Domažlice	Podrobnosti	Editovat	Smazat	Přifadit znalosti
4	Jiří	Barták	8004023561	j.bartak@unicorn.cz	Praha	Podrobnosti	Editovat	Smazat	Přifadit znalosti
5	Martin	Horáček	7701265024	m.horacek@unicorn.cz	Kolin	Podrobnosti	Editovat	Smazat	Přifadit znalosti
6	David	Tomášek	7202196497	d.tomasek@unicorn.cz	Praha	Podrobnosti	Editovat	Smazat	Přifadit znalosti
17	Milan	Kropáček	8003158846	m.kropacek@unicorn.cz	Kolin	Podrobnosti	Editovat	Smazat	Přifadit znalosti
18	Ondra	Strouhal	7201190565	o.strouhal@unicorn.cz	Výškov	Podrobnosti	Editovat	Smazat	Přifadit znalosti

[Přidat zaměstnance](#)
[Zpet](#)

[Zpět na hlavní stránku](#)

(c) Všechna práva vyhrazena

obrázek 9.2.1: GUI - Správa zaměstnanců

9.3 Správa elementů KH

Prototyp zobrazuje výpis seznamu elementů, možnosti přihlášeného uživatele s těmito právy jsou: přidání elementu, editace elementu, smazání elementu. Zobrazení podrobností je popsáno v podkapitole 9.5 Zobrazení podrobností o elementu.

Systém databáze znalostí zaměstnanců

[Odhlásit](#)

Seznam elementů:

ID	Zkratka	Název			
1	ENG1	Angličtina junior	Podrobnosti	Editovat	Smazat
2	ENG2	Angličtina senior	Podrobnosti	Editovat	Smazat
3	ENG3	Angličtina expert	Podrobnosti	Editovat	Smazat
16	NJ1	Němčina junior	Podrobnosti	Editovat	Smazat
17	NJ2	Němčina senior	Podrobnosti	Editovat	Smazat
18	NJ3	Němčina expert	Podrobnosti	Editovat	Smazat
22	CPP1	C++ junior	Podrobnosti	Editovat	Smazat
19	JAVA1	Java junior	Podrobnosti	Editovat	Smazat
20	JAVA2	Java senior	Podrobnosti	Editovat	Smazat
21	JAVA3	Java expert	Podrobnosti	Editovat	Smazat

[Přidat element](#)
[Zpet](#)

[Zpět na hlavní stránku](#)

(c) Všechna práva vyhrazena

obrázek 9.3.1: GUI - Správa elementů

9.4 Správa elementů KH zaměstnance

Uživatelském rozhraní tohoto prototypu umožňuje přihlášenému uživateli přiřazovat zaměstnancům jednotlivé dostupné elementy. Součástí zobrazení jsou také podrobnosti o zaměstnanci a seznam již přidělených elementů tomuto zaměstnanci.

System databáze znalostí zaměstnanců

[Odhlásit](#)

Přiřazení elementu zaměstnanci

ID:	2
Jméno:	Pavel Zámečník
RČ:	7509145211
email:	p.zamecnik@unicom.cz
Město:	Domažlice

Seznam přidělených elementů:
CPP1
JAVA2
ENG1

Celkem má zaměstnanec přiznaných elementů: **3**

Přidat element:

[Zpět na seznam zaměstnanců](#)

[Zpět na hlavní stránku](#)

(c) Všechna práva vyhrazena

obrázek 9.4.1: GUI - Správa elementů KH zaměstnanců

9.5 Zobrazení podrobností o elementu

Podrobnosti o elementu se zobrazují v následujícím GUI. Element je zde zobrazen i s popisem. Následuje výpis jednotlivých zaměstnanců, kteří tento element mají přiznaný. Toto zobrazení je dostupné všem zaměstnancům, není zapotřebí administrátorského přístupu.

System databáze znalostí zaměstnanců

Podrobnosti elementu:

ID:	22
Zkratka:	CPP1
Název:	C++ junior
Popis:	základy jazyka C++

Seznam uživatelů s tímto elementem:
Pavel Zámečník

Celkem má tento element uživatelů: 1

[Zpět na seznam elementu](#)

[Zpět na hlavní stránku](#) (c)2007 Všechna práva vyhrazena

obrázek 9.5.1: GUI – Zobrazení podrobností o elementu

10 ZÁVĚR

Cílem mé diplomové práce bylo provést analýzu a návrh informačního systému a sestavit k němu programový prototyp. Základem je tedy dokumentační část vypracovaná podle metodiky RUP. Jednotlivé kapitoly se v reálném použití metodiky RUP zpracovávají a předávají jako samostatné dokumenty. Tyto dokumenty shrnují důležité informace pro další skupinu vývojové části, která je podstatná pro další zpracování dokumentace k IS. Z této metodiky plyne, že i zde probíhá vzájemné předávání důležitých informací v rámci kapitol. Práce mi dala možnost zabývat se problematikou týmového vývoje produktu softwarové firmy. Při tvorbě dokumentu, jsem se řídil radami odborníků ze společnosti Unicorn a.s., kde tuto metodiku již úspěšně využívají.

Nashromážděné informace v dokumentu pokrývají 2 počáteční části metodického vývoje metodiky RUP. Na základě druhé elaborační fáze, kde se zabývám architekturou systému, je sestavený prototyp informačního systému. Prototyp běží jako součást webového serveru s využitím databáze. Prototyp je dle doporučení ze strany Unicornu, kde byla veškerá práce konzultována, naprogramován v jazyce Ruby za pomoci prostředí RadRails. Návaznost metodiky RUP je v tomto případě zachována, protože Ruby podporuje architekturu MVC (Model, Viewer, Controller). MVC zachovává rozdělení dat a akcí využitím tříd stereotypů typu entity, boundary, control. Ruby je moderní programovací jazyk s jehož pomocí lze vyvíjet rozsáhlé informační systémy právě podle metodiky RUP. Jedná se o objektový jazyk, který nabízí ve svém prostředí práci s třídami (modely). Díky této vlastnosti je zachována logika i funkcionalita. Tento jazyk je využíván zejména ve společnostech přizpůsobujících se novým trendům ve vývoji software. Umožňuje flexibilní se přizpůsobení změnám v návrhu tříd, jelikož se třídami v podobě modelů sám pracuje.

Přínosem bylo seznámení se z moderním způsobem vývoje softwarového produktu od analýzy, přes návrh, až po vytvoření programového prototypu poskytujícího stejné možnosti jako zmiňovaná metodika. Zejména programová část (prototyp) ukazuje jaké možnosti se skrývají za moderním vývojem software. Problém na který jsem narazil, zejména při samotném programování prototypu, byl nedostatek dokumentace, zejména pro práci s databázemi. Dalším milníkem dokumentační části, by bylo zpracování zbývajících dvou fází metodiky, tedy konstrukční a fáze zavedení. Tento dokument bude sloužit jako vstupní dokument pro fázi implementace. Další vývoj prototypu by mohl směřovat ke kompletnímu informačnímu systému zpravující veškeré stanovené funkce. Ideálním programovacím jazykem pro produkci výstupů z dokumentace je právě jazyk Ruby. Jedná se o jazyk, který si určitě zaslouží zvýšenou pozornost, právě pro jeho budoucí využití v praxi. Předpokládá se, že se v budoucnu naskytne příležitost k rozšíření základní funkčnosti systému. Pro další rozvoj systému lze započítat multi-jazyčné prostředí, podpora pro přístup z mobilních zařízení např. PDA a v neposlední řadě integrace systému do stávajícího informačního systému a tím jejich vzájemné propojení.

Věřím, že tato práce ukázala moderní možnosti jakými lze vést vývoj informačního systému. Pomocí metodiky RUP a následným zpracováním v prostředí Ruby on Rails software nepřestane být moderním a přizpůsobivě reagovat na nové požadavky a změny.

11 LITERATURA

11.1 Seznam použité literatury a materiálů

- [1] Šork Radim, Vitouš Otto. Rational Unified Process, stručný průvodce. Unicorn Multimedia, 1. vydání, 2000. ISBN: 80-238-6358-4.
- [2] Adopting The Rational Unified Process - PDF (Petr Janda, Michal Kadák, Petr Kobiersky, Michaela Kolouchová, Tomáš komenda, Jozef Ličko, Michal Minárik)
- [3] Informace o Use Case <http://web.sks.cz/users/ku/PRI/usecase.htm> (květen 2007)
- [4] článek o znalostech a jejich principech, návaznost na konzultaci <http://www.unicorn.cz/cz/press/clanek.php?id=7057> (květen 2007)
- [5] Specifikace UML, diagramy - <http://vendulka.zcu.cz/Download/Free/MetodySA-OO.doc> (květen 2007)
- [6] Jim Arlow, Ila Neustand. UML a unifikovaný proces vývoje aplikací. Computer press Addison-wesley. 1. vydání, 2003. ISBN 80-7226-947-X.
- [7] Elektronická dokumentace - Rational Unified Process 2003.06.00.65, Rational Software Corporation
- [8] <http://rails.jinak.cz/>
- [9] <http://www.root.cz/clanky/ruby/>
- [10] <http://pragmaticprogrammer.com/titles/rails>
- [11] Pojmy ze slovníčku pojmů čerpány z <http://wikipedia.com>

11.2 Použité nástroje

Rational Rose 2002, Microsoft WORD, Visio, Adobe Acrobat, Ruby on Rails, Rad Rails

11.3 Slovníček pojmů

Zkratka	Název
<<boundary>>	Stereotyp třídy, označení třídy rozhraní. (View)
<<control>>	Stereotyp třídy, označení třídy reprezentující zpracovávání dat. (Controller)
<<entity>>	Stereotyp třídy, označení řídicí třídy. (Model)
Apache	Softwarový webový server s otevřeným kódem umožňující umožňuje prohlížení stránek na základě HTTP protokolu.

CRUD	Zastupuje funkce Create Read Update Delete
Firewall	Síťové zařízení, sloužící k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a nebo zabezpečení.
Framework	Softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů.
GUI	Grafické uživatelské rozhraní
HTTP	Hyper Text Transfer Protocol, internetový protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML
HTML	Hypertext Markup Language, formát souboru
ID	Uživatelské číslo
IS	Informační systém
KH	Know-how, znalosti
Klient	Fyzická, nebo právnická osoba, která si nechává vytvořit IS firmou využívající databázi znalostí.
MTBF	Mean Time Between Failures - Průměrná doba mezi pády systému
MTTR	Mean Time To Repair - Průměrná doba zotavení systému
PDF	Portable Document Format, formát souboru
PSC	Poštovní směrovací číslo
RČ	Rodné číslo
RUP	Rational Unified Process
Ruby	Interpretovaný skriptovací programovací jazyk. Je plně objektově orientovaný.
SOAP	Simple Object Access Protocol – protokol pro komunikaci založený na XML. Slouží k výměně informací v decentralizovaném prostředí.
UAT	Uživatelské akceptační testy – testy použitelnosti software
UML	Unified Modeling Language, jazyk pro popis modelovacího procesu
XHTML	Extensible Hypertext Markup Language, rozšířený značkovací jazyk, formát souboru
Typové řešení	Informační systém, který byl „ušit“ na míru konkrétní problematice konkrétního tržního segmentu.

11.4 Seznam obrázků

obrázek 2.2.1: Osoba, aktivity a artefakty.....	12
obrázek 2.2.2: Lidé a osoby.....	13
obrázek 2.2.3: Příklad pracovního postupu	15
obrázek 2.3.1: Sekvenční proces	16
obrázek 2.4.1: Model architektury 4+1	22
obrázek 3.4.1: Systém evidence znalostí zaměstnanců – logický pohled	31
obrázek 3.4.2: Systém evidence znalostí zaměstnanců – přehled.....	32
obrázek 4.2.1: Případy užití – hlavní	40
obrázek 4.3.1: Příklad užití – Správa údajů o zaměstnancích	41
obrázek 4.4.1: Správa pohovoru.....	45
obrázek 4.5.1: Správa elementů KH zaměstnance	48
obrázek 4.6.1: Žádost o přidělení znalosti.....	51
obrázek 4.7.1: Správa elementů KH.....	52
obrázek 4.7.2: Závislosti elementů a specializací.....	55
obrázek 4.8.1: Správa údajů o projektu.....	56
obrázek 4.9.1: Správa požadovaných elementů.....	60
obrázek 5.5.1: Model problémové domény.....	66
obrázek 8.5.1: Popis architektury – rozvrstvení základních vrstev	70
obrázek 8.5.2: Use-Case realizace.....	71
obrázek 8.5.3: Stereotypy tříd boundary, control, entity	71
obrázek 8.5.4: Class diagram – Seznam elementů.....	72
obrázek 8.5.5: Sekvenční diagram – Seznam elementů	72
obrázek 8.5.6: Class diagram – Vyhledání uživatele.....	73
obrázek 8.5.7: Sekvenční diagram – Vyhledání uživatele	73
obrázek 8.5.8: Class diagram – Přidělení znalostí	74
obrázek 8.5.9: Sekvenční diagram – Přidělení znalostí.....	74
obrázek 8.5.10: Class diagram – Správa pohovoru.....	75
obrázek 8.5.11: Sekvenční diagram – Správa pohovoru.....	75
obrázek 8.5.12: Class diagram – Tisk elementů.....	76
obrázek 8.5.13: Sekvenční diagram – Tisk elementů.....	76
Obrázek 8.6.1: Pohled nasazení	77
obrázek 9.1.1: GUI - Přihlášení uživatele do systému.....	79
obrázek 9.2.1: GUI - Správa zaměstnanců.....	80

obrázek 9.3.1: GUI - Správa elementů	80
obrázek 9.4.1: GUI - Správa elementů KH zaměstnanců.....	81
obrázek 9.5.1: GUI – Zobrazení podrobností o elementu.....	82

12 PŘÍLOHY

Příloha 1. CD s prototypem programu s manuálem na obsluhu.